Master´s Thesis

Do news move the world?
Analysis of trend predictability in public traded financial instruments via automated semantic analysis.

Victor-Johannes Holl

Master´s Thesis

at Frankfurt School of Finance & Management

Do news move the world? Analysis of trend predictability in public traded financial instruments via automated semantic analysis

Supervised by

Prof. Dr. Florian Ellsäßer

Levente Szabados

Submitted by:

Victor-Johannes Holl

Master in Applied Data Science 1821

8401056

Sandweg 64, 60316 Frankfurt am Main

+491724385596

victor.johannes.holl@gmail.com

Frankfurt am Main, September 20

Table of Content

Table of Figures

List of Tables

1 Introduction

Do news move the world? This question concerns the humanity for years. Can someone use the information of the news to his or her advantage? In the world of high frequency trading it is important to know everything of what it is going in the world but the question is how to use it to predict what the trend of a security is or going to be. In the past before the upcoming of the internet and fast computers every person had the same chance to receive and analyse the news at the same time to see what kind of information it can tell about how a security will move in the future. The analysts draw their own conclusion what the news tells about the stock market. The risks of the human analysis were and still are today that the factor of subjectivity cannot be eliminated. Each individual human interprets news in different ways. There is a massive amount of data available in the fast-growing digital world, therefore the human needs the help of computers and algorithms to gather and compensate the data which are important for the analysis of stock price movement. The question is though can these algorithms help the humans to make better predictions. Is objectivity of the algorithms an advantage for the world or is the elimination of the subjectivity become a problem. Can algorithms have the same experience as investors?

Due to its complexity stock price trend prediction are a complex task. Financial news is not the only key factor which influence the movement of the price. Algorithms do technical analysis, but can they also do fundamental analysis. They can see pattern how a company operated in the past, but it cannot predict how a CEO will run his or her business in the future. What happens if there is a change in leadership, not only in companies but also when new governments have been elected.

The following paper tries to get to the bottom of these questions if machine learning algorithm can really make trend predictions with the use of financial news or are they going to be only tools the investors uses as a help for their security picks.

2 Literature Review

In todays world machine learning becomes a more and more important aspect. Especially in the business world. For the financial industry it is an urgent matter to stay on top of their competitors therefore, it is of paramount necessity to predict the development of prices of securities. Today it is obvious that the smallest news or social media post can have an impact on the stock markets. According to the paper "Improving Stock Market Prediction by Integrating Both Market News and Stock Prices" by Xiaodong Li, ChaoWang, JiaweiDong, FengWang, Xiaotie Deng and Shanfeng Zhu the Efficient Market Hypothesis, also called EMH, theoretically explains that stock prices already include and reveal all the information which are known for every investor in the market. This approach was refuted by researches of behavioral finance theory. They argue that EMH does not include the irrational behavior of the human who are influenced by various kind of market information as well by the individual interpretation of the information. Although there are differences between the theories both do not ignore the effect of market information. As mentioned by Li et al. news articles are known as one of the most important part of market information research and are widely used and analyzed by investors. With Bloomberg and Refinitiv, formerly known as Thomson Reuters, which are specialized on providing news the amount of it becomes overwhelming. Therefore, the financial institutions rely on high processing computing power to process all the information provided and analyze the content to predict the relevance regarding the impact on the stock market. The paper mentions that computer science researches studied the problem of how to model and analyze the market information so investors can gain an advantage of predicting the movement of security prices. According to their study their algorithm could give a directional prediction like "up, hold, down" based on the newly released news articles. But text classification which is the case in the analysis of financial news can only consider the articles impact but does not consider the information which are hidden in the security price shortly before the news was released, like other market news which could have had and impact on the stock price trend. Li et al. also mention if taking the news articles and the short time history price into consideration they believe that positive news may not always lead the security price of increasing immediately. It might just stop the price trend of falling further, because it is the first positive news for a long time, and it can indicate a change of the trend and it might lead that the stock price will increase again. Negative news does not necessarily mean the trend of the movement changes from going up to down. It can cause the price curve to be flatter. The main

statement which is proposed by Li et al. is that the traditional view of good news means up and bad news means down is obsolete due to the different impact news can have as described above.

The collection of the financial news can be a big challenge. An investor must pays thousands of Euro per month to Bloomberg or Refintiv (former Thomson Reuters) to use their services, which only big player like Hedge Funds or banks can do, but private or small investor do not have the financial power do so. These investors can use the power of machine learning algorithms to gather information they need for the analyze. According to the article on the Forbes website "AI Making Waves In News And Journalism" the technology is used more and more in everyday business. Machine Learning algorithms or as called in the article Artificial Intelligence (AI) are used to gather information from multiple sources in a short amount of time to help for the research. Machine learning algorithms can be programmed to search for articles with certain keywords depending on what the user is looking for.

After the collection of the market information, the collection of the information must be analyzed of their statement. Is the news positive, negative or neutral? The article "Big Data: Deep Learning for financial sentiment analysis" on the "link.springer" website is trying to tell the reader that with the help of deep learning so called data mining algorithms huge amounts of market information can be interpreted and analyzed where a single human would need months to do so. This time saving methods are crucial in the financial world because as the saying goes "Time is Money". Kalyani Joshi, Prof. Bharathi H., Prof. Jyothi Rao try to explain in their paper "STOCK TREND PREDICTION USING NEWS SENTIMENT ANALYSIS" when using sentiment analysis it is important to understand when predicting stock price movements that news sentiments are quantified and then presented as a time series. Therefore, it is important to use the right algorithm which can handle the issue of time series.

In this paper the attempt is made to see how far news can have an impact on the stock price movement. Are there stocks were news can have bigger impact than on others, which means are the market information already included in the security price when the news was published. Can news actually increase the price or drop the price, or do they just convert the current trend the security has into the other direction? Another question is what kind of influence the news have on the volatility of the security. The last question is which algorithm has the best ability to handle time series and predict the stock price movement?

3 Semantic Analysis

If the findings of the semantic analysis of financial news can have an impact on the movement of public trade instruments the term semantic analysis must be defined. Firstly, the definition of Natural Language Processing (NLP) needs to be discussed. NLP handles and analyses the natural language and is part of machine learning. NLP assists computers to understand, interpret and manipulate the human language and should close the gap between human communication and the language processing skills.

The importance of achieving a working knowledge of NLP is knowing that NLP is a supervised learning algorithm. In supervised learning the developer of the algorithm knows the input (x) and the output (y) variable and uses "the algorithm to learn the mapping function (y = f(x)) from the input to the output"[1]. The objective of a supervised learning algorithm is to approximate the mapping function so well that with new input data (x) the algorithm predicts the output (y) of the data. For a clearer understanding, imagine a "teacher is supervising the algorithm"[2]. The correct answer is known by the teacher, iteratively the algorithm is making predictions based on the training data and will be corrected by the teacher until the algorithm has an acceptable outcome.

3.1 Definition of Semantic Analysis

In the field of "Natural Language Processing" the term semantic analysis is often mentioned, but to go further into detail of NLP and its impact in "Big Data", we must take the term "sentiment analysis" into consideration. The processes of semantic analysis and sentiment analysis are often applied at the same time but in order to gain a thorough understanding of these terms. The differences between the two need to be clarified.

3.1.1 Semantic Analysis

The term "semantic" in general means the "the study of meaning"[3]. It is at the centre of having a deeper understanding of the "nature of language and human language abilities"[4]. Semantic analysis assists in providing transparency to the study of linguistics and to avoid ambiguity. Throughout the analysis not every word or sentence is processed

[1] https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/
[2] https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/
[3] https://books.google.de/books?hl=de&lr=&id=XW4WL3mKjjkC&oi=fnd&pg=PP2&dq=semantic+analysis+definition&ots=X3mGz-LWsL&sig=QpAVw3dJEkO72avw__ejcCVlkQY#v=onepage&q&f=false
[4] https://books.google.de/books?hl=de&lr=&id=XW4WL3mKjjkC&oi=fnd&pg=PP2&dq=semantic+analysis+definition&ots=X3mGz-LWsL&sig=QpAVw3dJEkO72avw__ejcCVlkQY#v=onepage&q&f=false

and analyzed individually, instead the entire text is analyzed to avoid, as already mentioned, the ambiguity of single words or single sentences.

Antonyms for the word ambiguity can be for example “mine”[5], “apple”[6] or much more. “Mine” can either mean that a person is in the possession of an object or it can be understood as the place where raw materials like gold, coal, etc. are extracted from. The word “apple” can also have more than just one meaning if you are unaware of the context. It can either express the fruit “apple” or the technology company. Ambiguity does not just occur within single words; it can also turn up in a sentence. For example, the sentence “The professor said on Monday he would give an exam,”[7] clearly shows that it has a double meaning. It can be interpreted the professor has informed the class on Monday that there will be an exam, or that the exam will be on the next Monday[8].

3.1.2 Sentiment Analysis

Sentiment analysis or opinion mining on the other hand focuses on the analysis of “people's opinions, sentiments, evaluations, attitudes, and emotions from written language”[9]. The results of sentiment analysis are expressed in three different categories, positive, neutral, and negative. The usage of sentiment analysis is applied not just in computer science, but also in “management science and social science due to its importance to business and society”[10] decisions.

3.2 Semantic Analysis Today

Semantic analysis is used in data science especially in the field of Natural Language Processing and Text Mining. Semantic and sentiment analysis are playing a bigger role in many businesses today. With the help of semantic/sentiment analysis businesses can quickly gather deeper insight into larger amounts of data, “especially customer data”[11]. Companies can monitor social media posts about themselves and their products to evaluate and readjust in order to go ahead of their competitors.

Semantic Analysis is also applied when a company is facing a PR crisis. As a result of this analysis the company can quickly react to bad news and take the right actions to maneuver themselves out of the crisis if applied in the correct way. Furthermore, if a

[5] https://www.digital-mr.com/blog/view/sentiment-and-semantic-analysis
[6] https://www.digital-mr.com/blog/view/sentiment-and-semantic-analysis
[7] https://www.thoughtco.com/syntactic-ambiguity-grammar-1692179
[8] https://www.thoughtco.com/syntactic-ambiguity-grammar-1692179
[9] https://www.morganclaypool.com/doi/abs/10.2200/s00416ed1v01y201204hlt016
[10] https://www.morganclaypool.com/doi/abs/10.2200/s00416ed1v01y201204hlt016
[11] https://www.analyticsinsight.net/sentiment-analysis-influencing-todays-market/

human being is facilitating the semantic analysis it may create mistakes as a result of the subjectivity that individuals possesses. Humans build their own opinion about the message of a sentence. For example it could be perceived as positive, neutral, or negative depending on who was asked. Humans have their own subjectivity, own beliefs, thoughts, and perceptions of the world and how these things are interpreted depends on the person who analyses the information. Semantic Machine Learning algorithms on the other hand, eliminate the subjectivity in human thinking, so that companies can be more efficient and accurate in their analysis of tweets, news, or other social media posts. This elimination is used to try to show the opinion of the majority of the people.

As demonstrated in the above chapter semantic analysis is used by companies to improve their relations with their customers. Companies analyze feedback from their customers and attempt to improve their businesses accordingly.

3.3 Semantic Analysis in Financial World

Semantic and sentiment analysis are used in the financial world to "extract insight of news, social media, financial reports"[12] and other financial related data. The gathered data can be used for the following cases:

- Analyze news articles or social media posts and trade a security shortly after the analysis
- Analyze large amount of financial reports and gather important insights from it
- "Gather insights"[13] from social media posts, "web forums, news and analysts' reports"[14]

Using the collected data in an effective way it can help investors, "stock market participants"[15] and others make quicker decisions before the market can react to it. As said sentiment analysis in finance is mostly used in "predicting the behaviour and possible trend"[16] stock market movements. Sentiment analysis receives a lot of attention as it can be integrated into the Fundamental Analysis, which is the analysis of stocks with the help of macroeconomic, industry-sector-specific and company specific data which are

[12] https://algotrading101.com/learn/sentiment-analysis-python-guide/
[13] https://algotrading101.com/learn/sentiment-analysis-python-guide/
[14] https://algotrading101.com/learn/sentiment-analysis-python-guide/
[15] https://www.analyticsinsight.net/sentiment-analysis-influencing-todays-market/
[16] https://www.twinword.com/blog/sentiment-analysis-for-the-financial-sector/

available to the public[17] and Technical Analysis in which the user chooses his or her investment strategies by "analysing statistical trends"[18] where the information is "gathered from the trading activity, such as price movements and volume"[19]. In the past financial analysts and traders based their trading decisions on news they had collected themselves. This approach that the traders used, was very subjective, subsequently with the help of computational semantic analysis and machine learning techniques the task of analysis became efficient.

Not only can Banks or financial institutions use semantic/sentiment analysis for their investment decisions, they may also use it in the same way as other companies outside of the financial sector. These institutions can use semantic/sentiment analysis for analysing customers opinions to gather information about what opinion a customer has of his or her financial institute. The gathered information can be used to improve the service of the bank or financial institute. In order to prevent loose any customers, the banks have to use the gained information to improve the business with the gathered information. At the end, with the improved business the banks can win new customers.

[17] http://www.daswirtschaftslexikon.com/d/fundamentalanalyse/fundamentalanalyse.htm
[18] https://www.investopedia.com/terms/t/technicalanalysis.asp
[19] https://www.investopedia.com/terms/t/technicalanalysis.asp

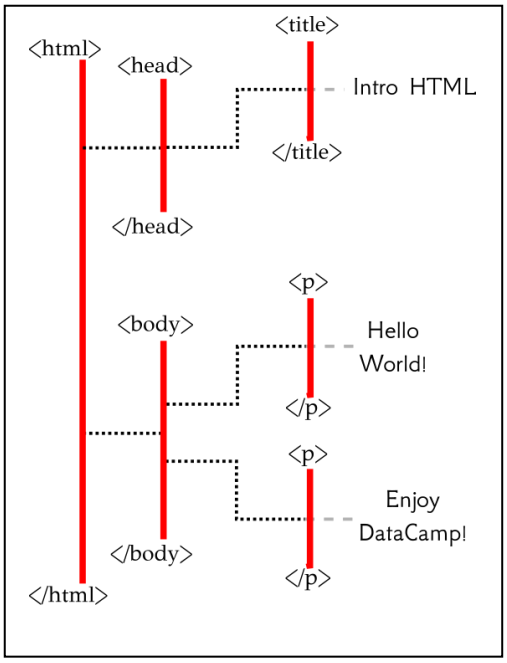
4 Data Collection/Preparation

In order to see whether financial news has an influence on the price movement of public traded financial instruments, the collection of data plays an important role. For the data collection the two following tools will be explained in the next chapters. Gathering financial news is done by a web scraping tool called “Scrapy”. The functionality of Scrapy will be explained in detail later in the written report. The collection of the security price data was accomplished with the help of the tool “Thomson Reuters Eikon” of the company “Refinitv”.

4.1 Web Scraping

Web scraping is a method to extract useful information from the world wide web. Since the analysis is mainly bases on HTML (Hyper Text Markup Language) analysis, HTML structures should be explained. It is used by web browsers to display, manipulate imagines, texts, and other web site content in the usual format every web browser user knows. A website, as seen in **Figure 1**, is structured into the following parts:

- <Head
    - title
- Body
    - P



The head contains metadata, which means it shows all the information of the HTML page. The title in the HTML code is the title of a webpage a user sees either on the browsers title bar or on the pages tab. The body tag shows the website user all the visible content like headings, paragraphs, images, hyperlinks, tables, lists, etc. The p element in an HTML code defines a certain paragraph. In this paragraph you can find for example the beginning of an article which a website user can visit then via a hyperlink. The notation of this tags are as follows. It starts <tag> and it ends with </tag>.

Figure 1: HTML Structure [20]

For the web scraping part of this analysis the tool scrapy was used to extract the data in the necessary form. Scrapy is a "fast high-level web scraping/web crawling tool"[21]. It can be used in a wide range, "from data mining to monitoring and automated testing"[22]. Originally scrapy was developed for web scraping purposes but it is used to collect data via APIs (Application Programming Interface; API is used so multiple applications can interact with each other).

The algorithm, as seen in **Figure 2**, starts to make requests to the website given by the user. Thereby it crawls the website for the certain attributes given by the user. In this paper the algorithm crawls the certain Reuters webpage of a specific stock and parses certain hyperlinks (hyperlinks are links to other HTML-documents, for example other websites, a text document or graphic) to a next function where the algorithm scrapes for certain attributes defined by the developer. The hyperlinks which are important, are the once which connects to news articles about the company.

For the semantic analysis of the news the last functions of the algorithm scrape predefined attributes. For this paper the important attributes are the URL link of the article's website, article headline, the date when the article has been published and of course

[20] https://campus.datacamp.com/courses/web-scraping-with-python/introduction-to-html?ex=4
[21] https://docs.scrapy.org/en/latest/
[22] https://docs.scrapy.org/en/latest/

the content of the article itself. At the end all the necessary scraped information of the website is getting saved to a comma separate value file or csv file. Saving data to a csv file is a helpful method to work with the data in the future. The advantage of a csv file is that it is structured as data frame, which means it is structured in a tabular format. This means it is easier to read for an algorithm.

```
class FinanceNewsScraperSpider(scrapy.Spider):
    name = "scrapername"

    def start_requests(self):
        start_urls = ['https://www.reuters.com/companies/[security identifier (RIC)]/news',
                      ]

        urls = start_urls

        for url in urls:
            yield scrapy.Request(url=url, callback=self.parse_newspage)

    def parse_newspage(self, response):
        links = response.xpath('//a[contains(@href,"/article/")]/@href').extract() #extract hyperlink
        for url in links:
            yield response.follow(url = url,callback = self.parse_article)

    def parse_article(self, response):

        item = VolvoItem()
        item['article_link'] = response.url
        item['article_headline'] = response.xpath('//*[contains(@class,"ArticleHeader_headline")]/text()').extract()
        item['article_date'] = response.xpath('//*[contains(@class,"ArticleHeader_date")]/text()').extract()
        item['article_text'] = response.xpath('//div[@class="StandardArticleBody_body"]//p/text()').extract()

        print(item)

        #saving data to file.
        path = 'news/'
        file = '[security name]news_' + str(datetime.now().strftime("%Y%m%d-%H%M")) + '.csv'
        file_name = open(path + file, 'a')

        fieldnames = ['article_link', 'article_headline','article_date','article_text'] #adding header to file

        writer = csv.writer(file_name, lineterminator='\n')
        writer.writerow([item[key] for key in item.keys()])
```

Figure 2: Scraper For Reuters Website

A disadvantage of Scrapy is that it has problems with parsing dynamic websites. Dynamic websites can be explained by the following example of the Reuters website. When a user loads the website in the web browser just the latest articles will be shown. If the user starts scrolling downwards on the website the page starts to automatically load new content which represents article from further in the past. This content will be loaded by an API when a certain event is triggered, in this case scrolling downwards on the page. The developer can try to retrieve the data from this API but he or she will run into complications because the important part of the API link is generated randomly by Reuters to make it more secure.

4.2 Security Price Collection

There are many different options to gather security price data. Data can be retrieved from open source databases like Yahoo Finance which is the most common or it can be gathered by accessing the API from famous financial data providers like Refinitiv (former Reuters) or Bloomberg where data is more accurate and maintained.

Therefore, the data in this paper is collected from the Thomson Reuters Eikon API which is more reliable. The API from Thomson Reuter offers the user to collect historic price data for every minute, for every hour, and just daily price of the trading day.

The algorithm to retrieve the stock price data is easy and fast to write if you want to retrieve the data for one day at a time. The developer needs an API key to get connected. The next steps are to know the Reuter Identifier Code (RIC) to tell the algorithm which security it should be looking for.

```
# needed libraries
import eikon as ek
from datetime import datetime, timedelta

#Reuters API Key
ek.set_app_key('API Key')

#date modfication
days = int(input('Enter number of days you want to go back: '))

current_date = datetime.today().strftime('%Y-%m-%d')
current_date_for_modification = datetime.today()
date_minus_certain_days = current_date_for_modification + timedelta(days=-days)
modified_date = date_minus_certain_days.strftime("%Y-%m-%d")
print(current_date)
print(modified_date)

#RIC, fields, time to receive stock prices
rics = ['security RIC']
fields = ['OPEN','HIGH','LOW','CLOSE','VOLUME']
df = ek.get_timeseries(rics=rics,
                       fields=fields,
                       start_date=str(modified_date) + 'T07:00:00',#'2020-01-02T09:00:00',
                       end_date=str(modified_date) + 'T22:01:00',#'2020-01-02T17:30:00',
                       interval='minute')

print(df)

#safe to csv
df.to_csv(r'C:\Users\victo\Master_Thesis\stockprice_data\audi\daily_stock_prices\audi_prices_' + str(modified_date) + '.csv')
```

Figure 3: Code To Retrieve Security Prices From Reuters Eikon API

The format of the data comes in the usual OHLC (OPEN, HIGH, LOW, CLOSE), plus Volume form. The user must be aware of choosing the right time from which it should collect the data, because the time which is shown in the data after collecting is always UTC (Coordinated Universal Time) therefore if the user is in Germany he or she has to convert it to German time. It is necessary that the user of the algorithm converts the time to the time zone he or she is living in (example figure 4).

```
df_daily_stock_prices['Date'] = pd.DatetimeIndex(pd.to_datetime(df_daily_stock_prices['Date'])
                                                 ).tz_localize('UTC').tz_convert('Europe/Berlin')
df_daily_stock_prices['Date'] = pd.to_datetime(df_daily_stock_prices['Date'].dt.strftime('%Y-%m-%d %H:%M:%S'))
```

Figure 4: Convert UTC Time To German Time

Since the Thomson Reuters Eikon API will stop collecting data if there is no data available for a certain day, the algorithm needs to be programmed to skip certain days or to simply ignore fault messages. Collecting the data of each day to a csv file will run into problems. The Thomson Reuters Eikon API will stop collecting data if there is not data available on a certain day. This can be solved by ignoring the message and tell the algorithm to skip it (**Figure 6** in appendix).

A small disadvantage of the Thomson Reuters Eikon API is that data collection is only possible for the past year if the user tries to collect minutely and hourly data, so for this paper there is a time period of one year, starting at the 22/07/2019.

In this paper three different stock price data will be used, daily (one OHLC price per day, hourly, and minutely.

4.3 Data Preparation

Data needs to be properly formatted so it can be used for any kind of algorithm. The used file format for this paper is saved in a csv file. As already mentioned, saving the data in csv file is a helpful method to work with the data in the future.

The collected data of the stock prices are structured in the following table format. The first table shows the date and the time from when a certain price is from. The data is from every minute of a day during the trading period of the certain day. The second column of the price data file shows the opening price of the stock price, the third column represents the high price, the fourth column shows the low price to that certain time, the fifth column is the close price and the last column represents the volume of stocks which were traded to that certain time.

For doing the correlation analysis a new file format has been created, where the first five columns are the same as in the stock price file. The first new column is representing the differences between the closing price and the opening price to find out how the return

of the stock was to a specific time during the day. In the next new column, the user can find the one-hot encoded stock return. This means the return is represented in binary terms. A “0” represents to a certain time the return has been negative and if the return of the stock was positive it is shown as a “1”. The next two columns show the same for the volume of the traded stock.

The representation in files of the semantic analysis of the financial news, which will be explained in more detail in the following chapter, is formatted as well in csv files. The first columns are the same as how the web scraped data is saved. The subsequent columns show the output of each of the semantic analysis.

In the last file which is used for the prediction of the stock price movement all the feature of every file will be merged to one file. Each of the semantic analysis of the financial news will be merged to the trading time of the stock at the time the article was published to have an accurate prediction. If this would not have been done the algorithm can not accurately predict whether the content of the published article has an influence on the stock price movement or not.

5 Analysis

The research was manly focused on European Car manufactures of Audi AG, BMW AG (Bayerische Motoren Werke AG), Daimler AG, Ferrari NV, Fiat Chrysler Automobiles NV, Peugeot SA, Porsche Automobile Holding SE, Renault SA and Volkswagen AG and Volvo AB. The Audi AG stock will be eliminated from the analysis to the lack of stock price data of the Audi AG, which comes from the small free float ("0.36 %")[23] of the stock. In comparison the Daimler stock has a free float of "61.79 %"[24]. Free float is the term of how many stocks in total are available for the public to trade with.

In this research paper, the analysis will not just focus on the headlines of the collected financial news, as many research papers do, it will also investigate sentiment of the content of each article. In each analysis it will be shown that there can be differences between the sentiment of the headline and the content of the article.

5.1 Semantic Analysis of Financial News

There are a variety of models which can be used for analysing the semantics of financial news. For the analysis of the collected financial news the following models where used:

- Vader Sentiment
- Flair
- Textblob

5.1.1 Vader Sentiment

"Vader (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media" [25]. It "uses a combination of a sentiment lexicon and a list of lexical features (e.g., words) which are generally labelled according to their semantic orientation as either positive or negative"[26]. In other words, Vader uses the bag of words approach with simple heuristics, for example "increasing the intensity of sentiment if words like really, so or a

[23] https://www.onvista.de/aktien/unternehmensprofil/Audi-Aktie-DE0006757008
[24] https://www.onvista.de/aktien/unternehmensprofil/Daimler-Aktie-DE0007100000
[25] https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/
[26] https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f

bit are presented" [27]. (Bag of words is vector representation which turns normal text into "fixed-length vectors" while counting the appearance of each word).

An advantage of Vader is that it does not only tell if a text is positive or negative, it also calculates how strong the negativity or positivity is. Another advantage of this tool is that it recognizes if a sentence contains "negated positive words (e.g. not happy, not good) which will be classified as a negative sentence sentiment. There are simpler sentiment analysis tools which "will just take the average of the sentiments of the words and would miss subtle details like"[28] in the given example.

On the other side is Vader not able to handle Out of Vocab (OOV) words, which the tool has not seen before and therefore will not be classified according to their actual sentiments. Vader can be helped by the user by upgrading the lexicon where the user tells Vader the word and the weight of its sentiment as seen in **Figure 4**.

The output of the Vader sentiment analysis differs between minus one to one, where minus one means the text is highly negative, zero means the text is neutral and one means the text is extremely positive. An advantage of Vader is that the output can also be between zero and one or zero and minus one, this means the text has different weights which has an importance on the analysis if a certain article has a stronger or less stronger impact on the price movement of the stock.

```
# New words and values
new_words = {'crushes': 10,
             'beats': 5,
             'misses': -5,
             'trouble': -10,
             'falls': -100,
             }

print('Start!')
# Instantiate the sentiment intensity analyzer with the existing lexicon
vader = SentimentIntensityAnalyzer()
# Update the lexicon
vader.lexicon.update(new_words)

print('ok!')
```

Figure 5: Update Of Vader Sentiment Lexicon

As already mentioned in the introduction of this chapter there can be differences between the result of the analysis of the headline of the article and the content of the article. In the table below are just some examples where this issue occurs.

[27] https://medium.com/@b.terryjack/nlp-pre-trained-sentiment-analysis-1eb52a9d742c
[28] https://medium.com/@b.terryjack/nlp-pre-trained-sentiment-analysis-1eb52a9d742c

URL	compound_vader_header	compound_vader_articel_content
https://www.reuters.com/article/us-bmw-daimler-carsharing/daimler-bmws-free-now-service-to-restructure-integrate-french-app-kapten-i-dUSKCN21Y1JZ	0.616	-0.784
https://www.reuters.com/article/us-bmw-results-q1/bmw-cuts-outlook-sees-coronavirus-pain-lasting-all-year-idUSKBN22I0JE	-0.670	0.969
https://www.reuters.com/article/us-health-coronavirus-fca-loan/fiat-chryslers-loan-request-raises-doubts-about-5-5-euro-billion-dividend-i-dUSKBN22V0Q2	-0.29	0.962
https://www.reuters.com/article/us-fiat-chrysler-m-a-psa/fiat-chrysler-does-not-see-delay-in-psa-merger-fiom-union-says-idUSKBN21I3A6	0.241	-0.923
https://www.reuters.com/article/us-volvo-results/swedens-volvo-hit-by-cancelled-orders-as-pandemic-creates-new-normal-idUSKCN2250ID	0.025	-0.984
https://www.reuters.com/article/us-geely-volvo-recall/geelys-volvo-announces-its-biggest-ever-recall-over-seat-belt-cable-i-dUSKBN2424WT	0.0	-0.812

Table 1: Outcome Vader Where Sentiment Of Headline Is The Opposite Of Content

It can often be that the headline does not cover every information from the content. The headline is used to get a reader hooked to dive deeper into the article. The article content covers more information than the headline, many important information can not be covered by the headline. Therefore, the chances can be high that the sentiments of the headline differ to the sentiments of the article content.

5.1.2 Flair

Flair is an open-source library for python developed by Zalando Research and the Humboldt University of Berlin. Its "sentiment classifier is based on a character-level LSTM (Long Short-Term Memory) neural network which takes sequences of letters and words into account when predicting".[29] Flair allows the user to apply its model for named entity recognition (NER)[30], Part-of-Speech tagging, "sense disambiguation and classification"[31]. Named Entity Recognition can be explained by an example from the fashion

[29] https://medium.com/@b.terryjack/nlp-pre-trained-sentiment-analysis-1eb52a9d742c
[30] https://github.com/flairNLP/flair
[31] https://github.com/flairNLP/flair

world where it is tried to highlight and identify "fashion-related entities such as colors, looks, designs and brands in text"[32]. Part-of-Speech tagging (PoS)[33] is the process of converting sentences into forms like list of words. The tag in part of speech tagging means that a word is identified either as a "noun, adjective, verb"[34] or other.

The output of the Flair sentiment analysis comes with the following notation. Firstly, the sentiment is mentioned, whether it is positive or negative, secondly there is in brackets displayed how strong or not strong the sentiment is. The range reaches from zero to one. The main difference between Flair and Vader is that Flair does not flag texts with the sentiment neutral, so Flair just differ between positive and negative sentiments. This can have a huge impact of the analysis of the stock price movement. Financial news articles can also have a neutral sentiment which can lead the price of a stock moves sideward, although it can be assumed if the value inside the bracket is close to zero that article can be flagged as neutral.

URL	flair_sentiment_header	flair_sentiment_content
https://www.reuters.com/article/us-turkey-competition-germany-autos/turkish-competition-board-launches-probe-into-audi-porsche-vw-mercedes-benz-and-bmw-i-dUSKBN24267E	POSITIVE (0.914)	NEGATIVE (0.991)
https://www.reuters.com/article/us-hungary-audi-orban/hungary-ready-to-help-audi-run-local-plant-at-full-capacity-orban-idUSKBN23M1V7	POSITIVE (0.993)	NEGATIVE (0.995)
https://www.reuters.com/article/us-daimler-sales-premiumcrown/mercedes-benz-poised-to-clinch-premium-sales-crown-for-2019-idUSKBN1Z811M	POSITIVE (0.965)	NEGATIVE (0.791)
https://www.reuters.com/article/us-audi-management/audi-boss-to-head-carmakers-rd-division-sources-i-dUSKBN23P2N9	POSITIVE (0.984)	NEGATIVE (0.943)
https://www.reuters.com/article/volkswagen-management-diess/exclusive-volkswagen-supervisory-board-to-discuss-vw-brand-leadership-sources-idUSS8N2D9086	POSITIVE (0.814)	NEGATIVE (0.999)
https://www.reuters.com/article/volkswagen-ford/volkswagen-approves-further-projects-in-ford-alliance-i-dUSL8N2DA63Z	POSITIVE (0.971)	NEGATIVE (0.990)

Table 2: Outcome Flair Where Sentiment Of Headline Is The Opposite Of Content

[32] https://research.zalando.com/welcome/mission/research-projects/flair-nlp/
[33] https://github.com/flairNLP/flair
[34] https://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging/

As already explained in the chapter of the Vader sentiment analysis, there are also examples in the Flair analysis where the headline of the article is either positive or negative and the analysis of the article content is the exact opposite of it. This can be traced back to the explanation that the headline of a news article cannot cover all the information of the article content.

5.1.3 Textblob

Textblob is a python library which "provides a simple API (Application Programming Interface) to jump into common NLP task such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation"[35] and many more NLP related tasks. Textblob uses a bag of words classifier as well. The advantage of Textblob is that it analyses subjectivity of texts, it shows how factual and opiniated a certain text is.

On the other side it does not include heuristics, which means it does not "intensify or negate"[36] the sentiment of a sentence. The range of the semantics output of Textblob goes as well from minus one to one where one defines that a text is extreme positive where on the other side minus one shows the extreme negativity of a text, but Textblob also offers the analysis how subjective a text is. The scale for the subjectivity analysis in Textblob goes from zero to one where one means the text is highly subjective and zero shows the non subjectivity of the text.

[35] https://textblob.readthedocs.io/en/dev/
[36] https://medium.com/@b.terryjack/nlp-pre-trained-sentiment-analysis-1eb52a9d742c

For the prediction of the stock price movement it will be mainly focussed on the outcome of the polarity score of the Textblob which shows the sentiment analysis of a certain text.

URL	polarity_textblob_sentiment_header	polarity_textblob_sentiment_content
https://www.reuters.com/article/brief-psa-signs-social-solidarity-agreem/brief-psa-signs-social-solidarity-agreement-to-protect-employees-idUSFWN2BW0K2	0.033	-0.143
https://www.reuters.com/article/volkswagen-skoda-ceo/vws-skoda-boss-maier-to-step-down-at-end-of-july-idUSL1N2EG0RI	-0.155	0.288
https://www.reuters.com/article/us-volkswagen-results-2020/volkswagen-expects-very-bad-second-quarter-positive-2020-adjusted-operating-profit-idUSKBN23M1JB	-0.341	0.057
https://www.reuters.com/article/us-health-coronavirus-mexico-volkswagen/scores-of-volkswagens-mexico-staff-test-positive-for-coronavirus-idUSKBN23U05D	0.227	-0.024
https://www.reuters.com/article/us-daimler-outlook/daimler-to-deepen-cost-cuts-after-expected-quarterly-loss-idUSKBN2490YI	-0.1	0.014
https://www.reuters.com/article/us-renault-m-a-dongfeng/renault-quits-its-main-china-venture-after-weak-sales-idUSKCN21W0HB	-0.104	0.051

Table 3: Outcome Flair Where Sentiment Of Headline Is The Opposite Of Content

Textblob analyses as well as the Vader and Flair analyses also show that their can be differences between the outcome of the analysis for the header of financial news and the content of the article.

5.2 Security Price Prediction

Security Price Prediction movement can be done by different Machine Learning tools. In the upcoming chapter methods of security price trend predictions will be discussed since they must have the ability to handle time series. Time series plays a key role in security price prediction. Before going further into detail why time series is important in security price prediction. The term time series has to be clarified.

Time series is a set of data points/observations from specified time intervals which all have the same size. These time intervals are used to predict future values based on the previous observations which are as well classified in a same sized time interval [37]. This time intervals can be interpreted as windows. Usually the time intervals are represented as a hour, a day, a month, a week or a year.

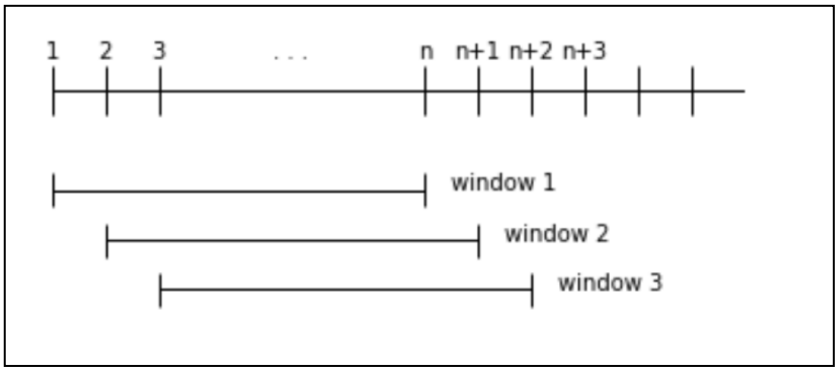


Figure 6: Times Series Windows [38]

Time series can be used for different kinds of prediction where time is an important part of it like "statistics, stock market forecasting, signal processing, pattern recognition, weather forecasting, earthquake prediction, astronomy, and largely in any domain of applied science and engineering"[39]. The main components of time series are trend, seasonality, irregularity and cyclic[40].

[37] https://medium.com/@josephabraham9996/time-series-analysis-on-stock-market-forecasting-arima-prophet-2b60cacf604

[38] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.splunk.com%2Fen_us%2Fblog%2Fit%2Fincremental-decremental.html&psig=AOvVaw3BJh88yQHJU56G6TM6iKcJ&ust=1596315126970000&source=images&cd=vfe&ved=0CAMQjB1qFwoTCJCTjvav-OoCFQAAAAAdAAAAABAD

[39] https://medium.com/@josephabraham9996/time-series-analysis-on-stock-market-forecasting-arima-prophet-2b60cacf604

[40] https://medium.com/@josephabraham9996/time-series-analysis-on-stock-market-forecasting-arima-prophet-2b60cacf604

5.2.1 Introduction of Different Machine Learning Algorithms in Stock Price Prediction

There are algorithms which can be used for time series analysis. In the following some of them will be explained and if they can be used for this kind of analysis or not.

5.2.1.1 Linear Regression

The prediction of the stocks can be done with a similar linear regression model. It can be used as base model to see how the prediction can be and where it can be improved. Linear regression comes from the area of statistics. In linear regression the algorithm is an approach to analyse the relation of a depended variable and one or more independent variables. It was adapted to be use in Machine Learning. The model representation of linear regression is simple. As mentioned earlier liner regression is an equation which combines a certain set x (independent variable) and y (dependent variable) [41], x is the solution to "the predicted output for the set of input variable (y)[42]. Both variables are in numeric notation. Linear Regression in Machine Learning is a supervised learning algorithm.

Shown in an example in the graph below were x (input variable) [43] can be the work experience of an employee and y (output variable) [44] is the salary of the employee. The regression line is the best fit to find out which salary is acceptable for a certain work experience.

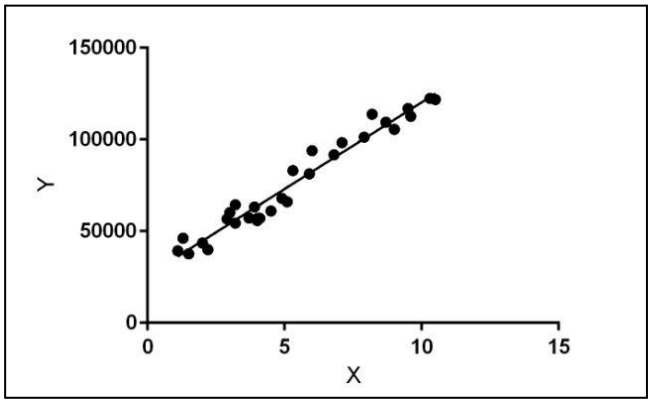


Figure 7: Example For Linear Regression [45]

[41] https://machinelearningmastery.com/linear-regression-for-machine-learning/
[42] https://machinelearningmastery.com/linear-regression-for-machine-learning/
[43] https://www.geeksforgeeks.org/ml-linear-regression/
[44] https://www.geeksforgeeks.org/ml-linear-regression/
[45] https://www.geeksforgeeks.org/ml-linear-regression/

The equation of linear regression can be represented as seen in figure 8 where x can be seen as “input training data (univariate – one input variable(parameter))” [46] and y can be seen as “labels to data (supervised learning)”[47]. The goal is to fit the regression in the best possible way by finding the optimal θ_1 and θ_2.

$$y = \theta_1 + \theta_2.x$$

Figure 8: Equation Of Linear Regression [48]

The disadvantage of linear regression in stock price prediction it does not “capture changes in direction (for example downtrend to uptrend and vice versa)”[49] in a fully accurate way.

[46] https://www.geeksforgeeks.org/ml-linear-regression/
[47] https://www.geeksforgeeks.org/ml-linear-regression/
[48] https://www.geeksforgeeks.org/ml-linear-regression/
[49] https://towardsdatascience.com/machine-learning-techniques-applied-to-stock-price-prediction-6c1994da8001

5.2.1.2 Random Forest

Random Forest is an algorithm which cannot be taken out of consideration for time series and security price prediction.

A Random Forest is used for a "variety of task"[50] like regression and classification, it is built up of a "large number of small decision trees, also called estimators"[51]. Each of the estimators produces its own predictions. The algorithm "combines the predictions of the estimators to produce a more accurate prediction"[52]. On the other side the standard decision tree classifier has the disadvantage that it tends to overfit to the training data set. The design of the Random Forest allows the algorithm to compensate the overfitting and "generalize well to unseen data"[53], this also includes data with missing values. Another advantage of the Random Forest is that it can easily handle large data sets "with high dimensionality and heterogeneous feature types"[54]. The advantage of Random Forest lies in classification problems then in regression. In comparison to linear regression a Random Forest regressor is "unable to make predictions outside the range of its training data"[55]. A disadvantage of Random Forest algorithms is that it is difficult to look inside "and understand the reasoning behind its decision"[56]. On the other side due to its robustness it is the perfect algorithm on heterogeneous data types. For many data scientists the Random Forest algorithm is the first algorithm to use "when developing new machine learning systems"[57] and to get a first overview what kind of accuracy can be achieved on a problem.

Gathering a better understanding of a Random forest it is important to understand how decision tree algorithms are built. Decision trees are a "simple way of classifying examples"[58]. Looking into the example of the Iris Dataset, "which is a set of measurements of 150 flowers belonging to three species"[59]. With the use of this commonly used data set a decision tree "can be built to identify which is the most likely species that a specimen belongs to"[60], depending on its petal length, petal width, and sepal length.

[50] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[51] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[52] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[53] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[54] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[55] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[56] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[57] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[58] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[59] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[60] https://deepai.org/machine-learning-glossary-and-terms/random-forest

There is a variety of decision tree algorithms where for all the basic process to build the tree for the data set example is as follows:

1. The attribute which can split the data set in the most efficient way has to be chosen. For the data set example splitting the petal length at 2.45 can be considered as a good choice "if most of the training examples"[61] which appear above 2.45 belong to one species and most of the other species are below this point.
2. The next step is to split the data set based on this attribute. "This corresponds to creating a new node in the decision tree"[62].
3. In the third step for every time the data set is split, the process gets repeated and splits the data set on the best attribute.
4. In the last step the process is stopped of creating new nodes in the tree if all samples in this node "already belong to the same class"[63]. Another point is the process gets stopped if none of the features provides any values "or if the tree has already reached its maximum allowed depth"[64].

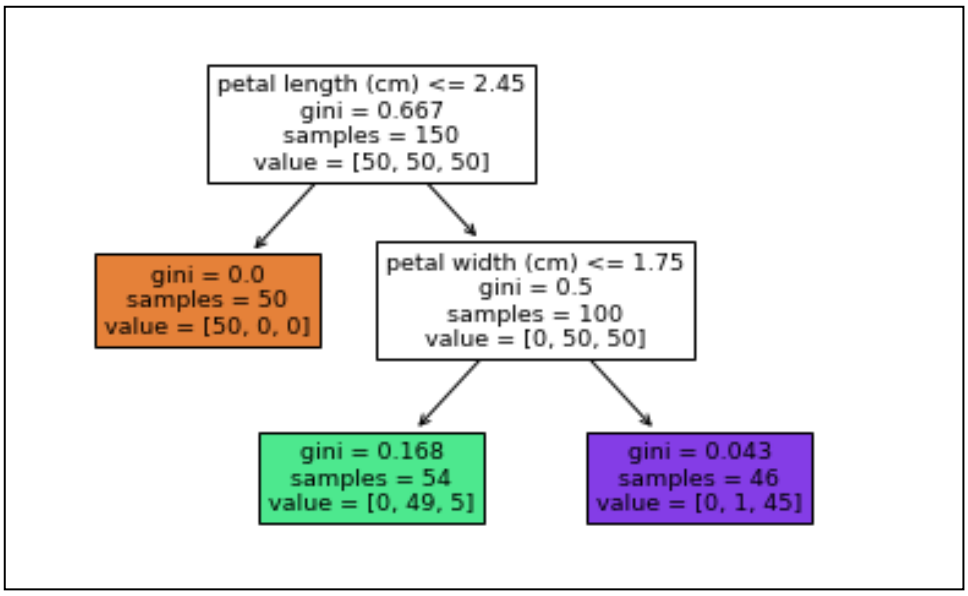


Figure 9: Example Decision Tree Of Depth Two For The Iris Data Set[65]

[61] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[62] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[63] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[64] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[65] https://deepai.org/machine-learning-glossary-and-terms/random-forest

Decision trees can be useful because they allow to instantly visualize the decision-making progress [66]. Decision trees in their basic form are limited to the size of the data set. This is the reason why the decision tree algorithm is barely used in the machine learning world due to the huge usage of data. Another reason for the rare usage of decision trees in machine learning is that they tend to overfit. Taking the data set of the example. There could be a single example of "iris setosa in the training data set with a given set of dimensions"[67]. This should indicate a different species but the decision tree "might create a node for that specimen"[68], that is were the Random Forest comes in handy. It is a modification to the basic decision tree and makes the algorithm more robust and corrects the problem of overfitting.

The most widely used Random Forest version based on "Leo Breiman's 2001 paper"[69] "RANDOM FORESTS". Assuming a training set has N training examples, each of it has N features. The Random Forest will have N_{tree} "decision trees, or estimators"[70].

A Random Forest consists out of different steps:

1. **Bagging[71]:**
   In this step the training set of N examples the algorithm repeatedly samples "subsets of the training data of the size n"[72], where n is smaller than N. The sampling happens random but with replacement. The subsampling of the training set "is called bootstrap aggregating"[73], or short bagging.

2. **Random subspace method[74]:**
   Assuming each training example has M features, the algorithm takes a subset of a size m < M to train each estimator [75]. With this method none of the estimators will see the full training set, each of the estimators will see only "m features of n training examples"[76].

[66] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[67] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[68] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[69] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[70] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[71] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[72] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[73] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[74] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[75] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[76] https://deepai.org/machine-learning-glossary-and-terms/random-forest

3. **Training Estimator[77]:**

   In this step the Random Forest creates N_{tree} decision trees (estimators) and will train each of the trees with a different set of "m features and n training examples"[78]. The trees in a Random forest "are not pruned, as they would be in the case of training a simple decision tree"[79].

4. **Perform inference by aggregating predictions of estimators[80]:**

   In this step it shows that the Random Forest can make prediction on new examples, the algorithm gets "passed the relevant of the example to each of the N_{tree} estimators"[81]. The Random Forest obtains the N_{tree} predictions, which needs to be combined to produce the overall prediction [82]. In the case of classification, the Random Forest will use majority voting to decide on the predicted class whereas in the case of regression it will take the mean value of the predictions of all the estimators[83].

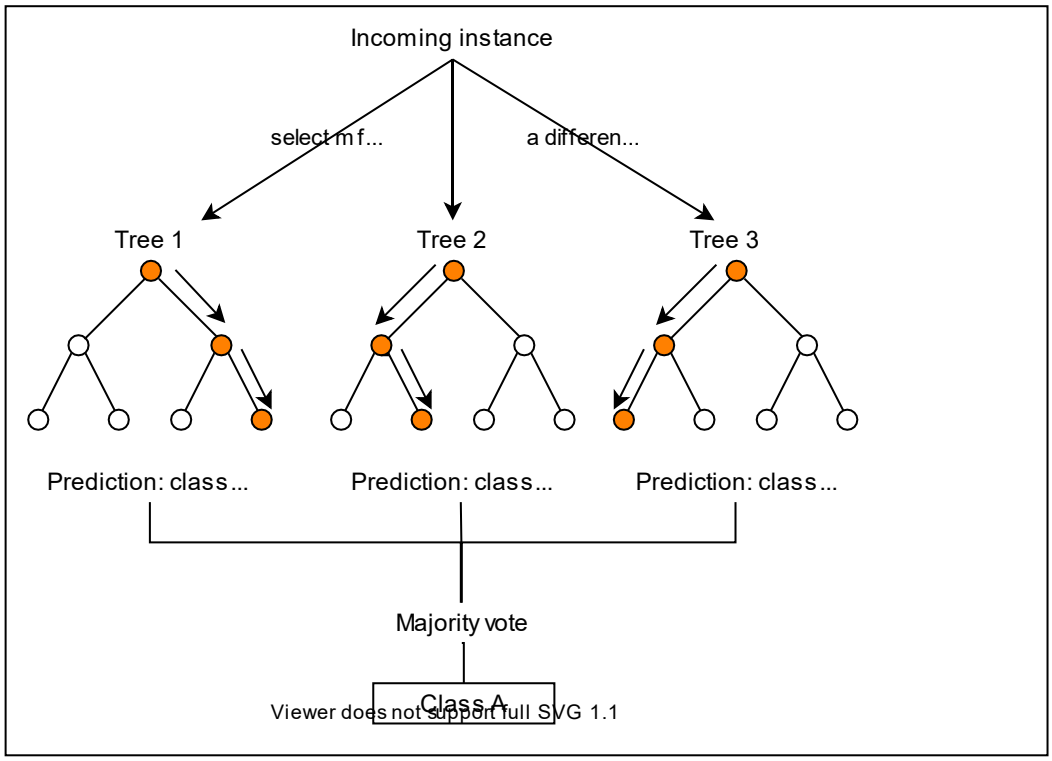


Figure 10: Random Forest Inference For A Simple Classification Example With Ntree = 3 [84]

[77] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[78] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[79] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[80] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[81] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[82] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[83] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[84] https://deepai.org/machine-learning-glossary-and-terms/random-forest

The use of many estimators is the reason why Random Forest is called an ensemble method[85], each one of the estimators is a weak learner, but when many weak trees are combined, they can produce a stronger learner. The ensemble method uses the strength in number approach, which means if the output of many smaller models is combined it produces a much more accurate and powerful prediction [86].

The question is now how a Random Forest handles time series. For Random Forest observations are independent and identically distributed, this is a problem violating the characteristics of times series data "which is characterized by serial dependences"[87]. Another problem is that the Random Forest is not able to predict trend which means "they do not extrapolate"[88]. The reason behind is that trees "operate by if -then rules that recursively split the input space. This leads to the fact that the Random Forest is unable to predict values which fall "outside the range of values of the target in the training set"[89], so in order to make the Random Forests to be able to predict stock market price the used data needs to be heavily preprocessed as can be seen in the code[90].

[85] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[86] https://deepai.org/machine-learning-glossary-and-terms/random-forest
[87] https://www.statworx.com/at/blog/time-series-forecasting-with-random-forest/
[88] https://www.statworx.com/at/blog/time-series-forecasting-with-random-forest/
[89] https://www.statworx.com/at/blog/time-series-forecasting-with-random-forest/
[90] https://github.com/JohannesHoll/Master_Thesis.git

5.2.1.3 Gradient Boosting

Another machine learning algorithm for stock price prediction is Gradient Boosting. XGBoost (eXtreme Gradient Boosting) is a gradient boosted decision tree algorithm which has been "designed for speed and performance"[91]. Gradient Boosting was initially developed for efficiency of saving computing time and memory resources. It can be easily explained with the AdaBoost[92] algorithm. AdaBoost evaluates the first tree of the algorithm where the weights of the observations, which are difficult to classify, are getting increased, where on the other side the weights of the observations, which are easy to classify, will be decreased. The second tree of the algorithm grows with the "weighted data"[93]. Mainly the algorithm tries to improve the prediction of the first tree with the constantly updated data of the second tree. AdaBoost and Gradient differs in how they handle and "identify the shortcomings of weak learner (for example: decision trees). The algorithm of AdaBoost recognizes the "shortcomings by using high weight data points"[94], Gradient Boosting performs in the same way although it uses the loss function (y=ax+b+e , e = error term) instead of the high data points. A Loss function describes how the coefficients of a model are fitting to its underlying data.)

Gradient Boosting comes with different advantages. It can handle missing data values automatically, it has a "block structure"[95] which helps to support and parallelize the tree construction. Another advantage is XGBoost constantly continues with the training, therefore a user of the algorithm can "further boost an already fitted model on new data"[96]. Particularly important is the scaling of the features, so XGBoost is running properly and to predict the stock price movement. Without any scaling the model trains just in a certain range of for example the closing price[97], which means the model predicts its outputs withing this certain range. If values are out of the model's range, it has difficulties to be able to generalize well.

It is always important to scale/normalize the data independent of the model which is used for the prediction. Taking all into consideration XGBoost can be a helpful Machine Learning tool to predict the stock price movement.

[91] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
[92] https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab
[93] https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab
[94] https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab
[95] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
[96] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
[97] https://towardsdatascience.com/machine-learning-techniques-applied-to-stock-price-prediction-6c1994da8001

5.2.1.4 Long Short-Term Memory Neural Network (LSTM)

The Long Short-Term Memory Neural Network (or short LSTM) will be the main machine learning instrument for the analysis in the paper.

Long Short-Term Memory is a term from the area of artificial intelligence and neural network[98]. Neural Networks in Data Science are artificial algorithms which try to replicate the real neural network of the human brain. The reasons for the attempt to replicate the human neural net work within an artificial neural network are:

- The human brain is powerful to recognize patterns. The visual recognition of the brain can identify an object in a cluttered scenery in a split of milliseconds which it might not have seen before regardless from the object's location and size.
- The human brain has the ability to learn "difficult tasks through practice"[99]. Humans are "general-purpose learning masters"[100] to solve complex and specialized problems.
- A human does not get born with the understanding of logic and rules. The human brain learns this behavior by constantly training it.
- A human brain is built with billions of neurons which are constantly communicating which one another.

LSTM Neural networks are part of the recurrent neural network (RNN) family where LSTMs are the most powerful and well known. RNN and LSTMs are designed for pattern recognition in sequences of data sets. These data sets can go from "numerical times series data emanating from sensors, stock market data"[101], texts, handwriting and the spoken word. The differences of LSTMs/RNN to other neural networks are that LSTMs and RNN "take time and sequence into account"[102], which means they have a temporal dimension.

To understand the functionality of Recurrent Neural Networks or LSTMs, it is important to understand the basics of feedforward networks. In feedforward neural networks the input data is fed into the network without going through a node twice. A node

[98] https://www.bigdata-insider.de/was-ist-ein-long-short-term-memory-a-774848/
[99] https://medium.com/cracking-the-data-science-interview/neural-networks-101-ee21cd508499
[100] https://medium.com/cracking-the-data-science-interview/neural-networks-101-ee21cd508499
[101] https://pathmind.com/wiki/lstm
[102] https://pathmind.com/wiki/lstm

"also called neuron or Perceptron, is a computational unit that has one or more weighted input connections"[103]. A certain function transfers the information which connects the input nodes and the output nodes in a certain way. Nodes are later connected to layers to form a network. For example, in a single layer network, as seen in **Figure 7**, each node of the single layer is connected directly to an input node and contributes at the same time to an output node.

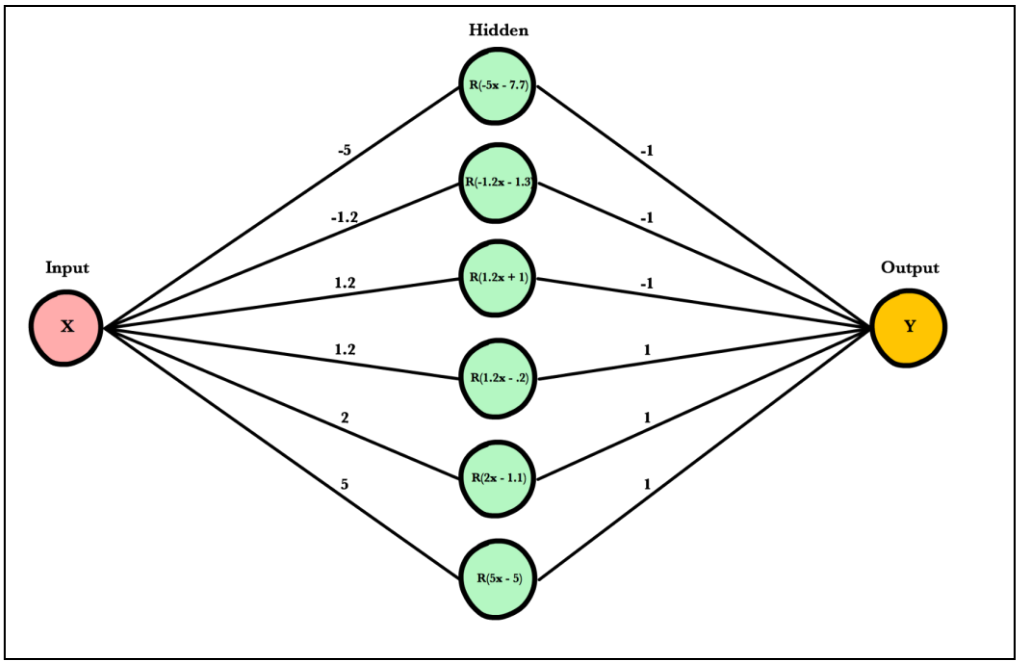


Figure 11: Single Layer Feed-Forward Neural Network [104]

In a feedforward neural network inputs are given to the network and later transformed into an output. Giving a supervised learning example to the network the output of the network will represent a certain label. Images are used as the input of the network and the output will label the images to a certain label like "cat", "boat", etc.

[103] https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/
[104] https://blog.goodaudience.com/neural-networks-part-1-a-simple-proof-of-the-universal-approximation-theorem-b7864964dbd3

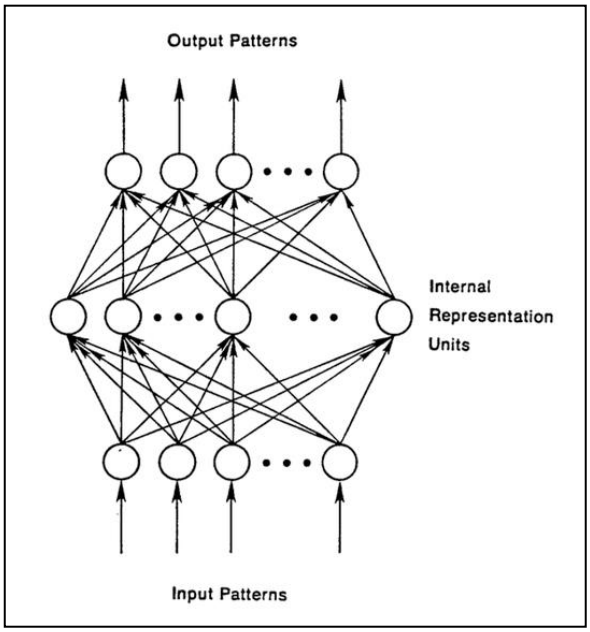


Figure 12: Structure Of A Feedforward Neural Network [105]

Feedforward neural network, which is trying to label images, trains itself to minimize the error it makes while predicting the label of a certain image. Based on the trained weighted parameters the neural network will go on and label the data it has not seen before. An already trained feedforward neural network can handle any random collection of images, which means that the first exposed image to the neural network will not mislead the network by classifying the next one.

Feedforward neural networks are not able to handle data with any kind of times series in it. They just consider the data they have been fed with; feedforward neural networks only remember "the formative moments of training"[106].

Recurrent Neural networks on the other side do not just consider the current input they have been exposed to but also the input they previously experienced. Explained in an example, seen **Figure 9**: BTSXVPE shows the input at the current state, where "CONTEXT UNITS represents the output of a previous moment"[107].

[105] https://pathmind.com/wiki/lstm
[106] https://pathmind.com/wiki/lstm
[107] https://pathmind.com/wiki/lstm

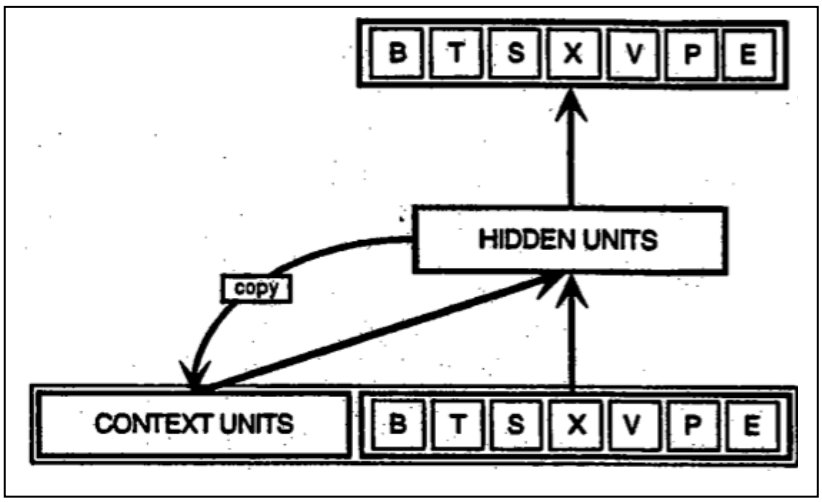


Figure 13: Visualization Of Recurrent Neural Network [108]

The CONTEXT UNITS, which represents "t-1" affects the decision the network will make at the moment "t" at the current state. Recurrent neural networks have therefore two input parameters, "t-1" (the past) and "t" (the present). Both parameters will determine how the network will react to new data. The main difference between feedforward neural networks and recurrent neural network is the "feedback loop"[109], which is the connection to the past. In other words, recurrent neural networks are algorithms with a memory. The purpose of having a memory and adding more and more memory information to the algorithm helps to find information in the sequence itself which feedforward neural network cannot do to perform the tasks. The sequential information is preserved in the hidden states of the recurrent neural network. Through this ability the network "manages to span many time steps as it cascades forward to affect the preprocessing of each new example"[110]. The recurrent neural network can find "correlations between"[111] different events which are separated by a certain number of steps. This part of the network is "called long-term dependencies"[112]. The term describes an event downwards in time depending on one or more events which came before. Recurrent neural networks can be described as networks which share weights over a certain period.

In comparison to the human neural network where memory is affecting the decision making without revealing the full intent why a human behaved in a certain way; the

[108] https://pathmind.com/wiki/lstm
[109] https://pathmind.com/wiki/lstm
[110] https://pathmind.com/wiki/lstm
[111] https://pathmind.com/wiki/lstm
[112] https://pathmind.com/wiki/lstm

hidden state of a recurrent neural network behaves in the same way. In mathematically terms the memory collection of the network can be described with the formula below:

$$\mathbf{h}_t = \phi\left(W\mathbf{x}_t + U\mathbf{h}_{t-1}\right),$$

Figure 14: Memory Collection Of Recurrent Neural Network

This function shows the input layer of the recurrent neural network, where "h_t" is the hidden state at time "t", "x_t" represents the new input which is modified by the weights "W" each input has. This product of "x_t" and their weight is added by the hidden state of the previous time step "h_{t-1}" which is then "multiplied by its own hidden-state-to-hidden-state matrix "U"[113], also know as transition matrix which is similar to a Markov chain. The weight matrices determine how important both inputs, the past and the current input, are. The errors which will then be generated are returned via backpropagation help to adjust the weights of the input layer until the error will be minimized to a certain level until the network cannot be more improved. The sum of the function will be fed to the function ϕ, which can be either a logistic "sigmoid function"[114] or a "tanh"[115]. This is a standard way for packing large and small values into a logistic space or "making gradients workable for backpropagation"[116].

The feedback, which brings the past steps into the input again, happens every time step. This means that every hidden state contains not only the memory of the previous hidden state but also the memory of those hidden states which occurred before "h_{t-1}" for as long as the network persist the memory.

Another important aspect, which must be mentioned in recurrent neural networks, is the term backpropagation. In feedforward neural network with the help of backpropagation the gradient can be efficiently evaluated by the "means of the error

[113] https://pathmind.com/wiki/lstm
[114] https://pathmind.com/wiki/lstm
[115] https://pathmind.com/wiki/lstm
[116] https://pathmind.com/wiki/lstm

backpropagation"[117]. The main idea behind backpropagation is to "propagate errors from the output layer back to the input layer by a chain rule"[118].

$$\frac{\partial E}{\partial \mathbf{W}^{(l)}} = \frac{\partial E}{\partial \mathbf{a}^{(L)}} \frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{a}^{(L-1)}} \cdots \frac{\partial \mathbf{a}^{(l+2)}}{\partial \mathbf{a}^{(l+1)}} \frac{\partial \mathbf{a}^{(l+1)}}{\partial \mathbf{a}^{(l)}} \frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} \frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{W}^{(l)}} \tag{1.6}$$

Figure 15: Chain Rule Of Backpropagation [119]

In backpropagation it is necessary to propagate the error from the cost function backwards to each layer and updates its weights according to the error message[120]. Achieving the goal of backpropagation successfully two things has to be taken care of:

1. Computation of the error message to the previous layer
2. Computation error of gradient to weight of this layer

Recurrent neural networks have an extension of the backpropagation which is called "backpropagation through time". Recurrent neural networks use backpropagation to "learn from sequential training data"[121]. In recurrent neural networks backpropagation comes with more challenges. The reason therefore is "the recursive nature of the weights and their effect on the loss which spans over time"[122]. The time part of the backpropagation extends the series functions which are calculates derivatives by the chain rule.

Before diving into the explanation of LSTM neural networks; the term "vanishing gradient"[123] has to be mentioned. In recurrent neural networks and in other deep neural networks the vanishing gradient challenges the model to learn long-term dependencies. Looking into the example below:

The brown and black dog, which was playing with the cat, was a german shepherd.
(x_2) (x_4) (x_5) (x_{14}) (x_{15})

Figure 16: Example For Vanishing Gradient [124]

[117] https://medium.com/machine-learning-for-li/explain-feedforward-and-backpropagation-b8cdd25dcc2f
[118] https://medium.com/machine-learning-for-li/explain-feedforward-and-backpropagation-b8cdd25dcc2f
[119] https://medium.com/machine-learning-for-li/explain-feedforward-and-backpropagation-b8cdd25dcc2f
[120] https://medium.com/machine-learning-for-li/explain-feedforward-and-backpropagation-b8cdd25dcc2f
[121] https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f
[122] https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f
[123] https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f
[124] https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f

The recurrent neural network has to predict the last two words. To have an accurate prediction the network needs to take the words “brown”, “black” and “dog” into consideration. These words describe the characteristics of the German shepherd. The word “brown” is quite far from the word “shepherd”. Breaking down “the backpropagation error of the word “shepherd” back to “brown”[125] a lot of chain rules are necessary. Through this amount of chain rules the model can run into trouble, if the loss of one of the chain rules approaches zero on the way from “shepherd” to “brown” it thereby vanishes. This makes it difficult to take words into account that stands at beginning of a sentence. In forward propagation the word “brown” might not influence the prediction of the word “shepherd” “because the weights were not updated due to the vanishing gradient”[126]. The vanishing gradient is the major flaw in recurrent neural networks.

There have been major developments in recurrent neural networks such as “gated recurrent units (GRUs)”[127] and LSTMs which are able to handle the vanishing gradient.

LSTMs were introduced in the mid-90s by the German scientist Sepp Hochreichter and Jürgen Schmidhuber to solve the problem of the vanishing gradient. LSTMs are especially designed to “avoid the long-term dependency problem”[128]. Their default behavior is to remember information for long periods of time. As already mentioned, “recurrent neural networks have the form of a chain of repeating modules of networks”[129].

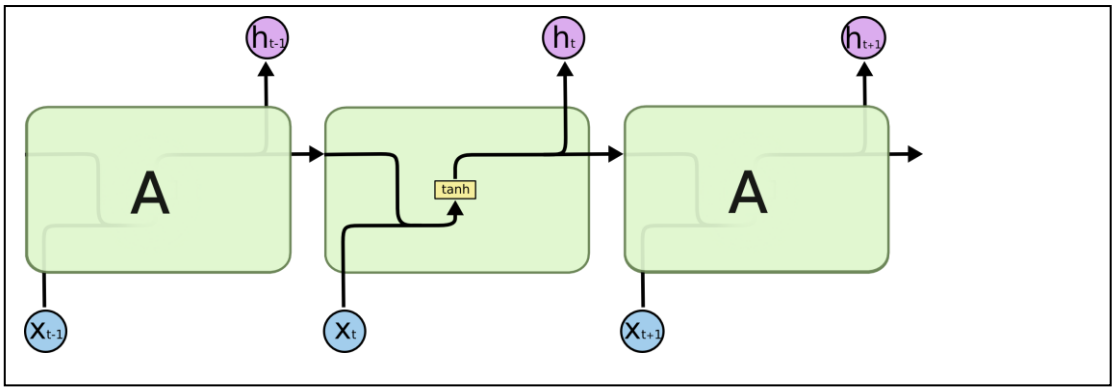


Figure 17: Structure Of Standard Recurrent Neural Network [130]

[125] https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f
[126] https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f
[127] https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f
[128] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[129] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[130] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

The repeating modules in this structure have a single tanh layer. “Tanh” is a so-called activation function. Activation functions tells the neural network if a neuron is active or not. This means if the information of this certain neuron will be used later or not. The range of the “tanh” function goes from minus one to one. LSTMs have the same structure.

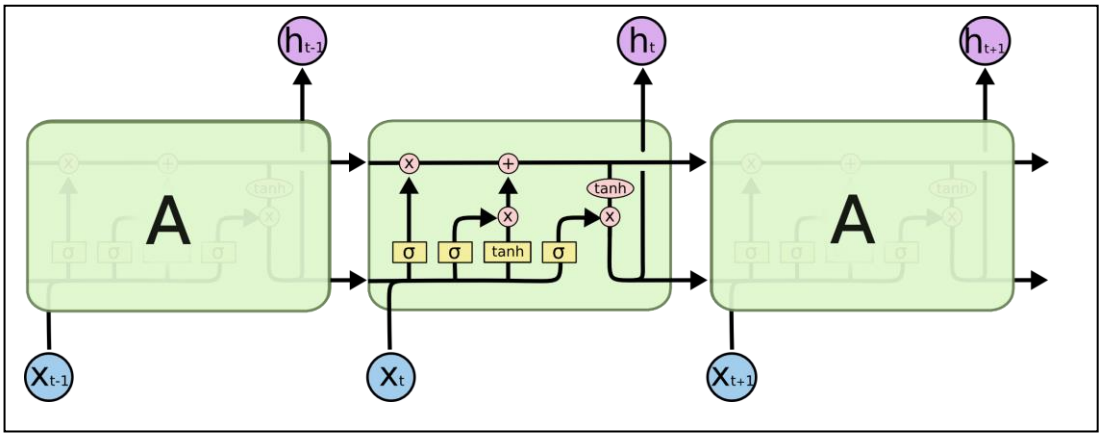


Figure 18: Structure Of A Long Short-Term Memory Neural Network [131]

The difference of recurrent neural networks to LSTM neural networks is that the repeating module has a different structure. Long Short-Term Memory neural networks have four layers instead of one, which interact in a special way to one another. To be more familiar with the meanings of the functions in the repeating module it is easier to understand the notation of it.

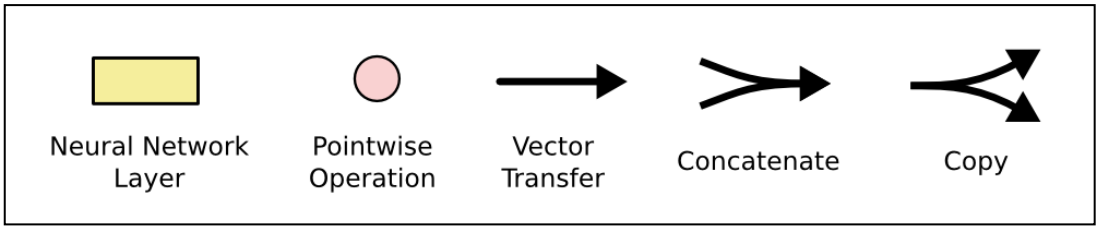


Figure 19: Meaning Of Symbols In The Repeating Module Of The LSTM [132]

The key to LSTM neural networks is the cell state[133] , which is represented by the line which runs horizontal trough the diagram at the top. The cell state goes straight through the entire chain, with some minor interactions in a module. Information from previous modules runs with the cell state. The LSTM can remove from or add to the cell

[131] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[132] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[133] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

state. This happens carefully by "structures called gates"[134]. These gates regulate which information go to the cell state and which not. This regulation is handled by a sigmoid neural net layer.

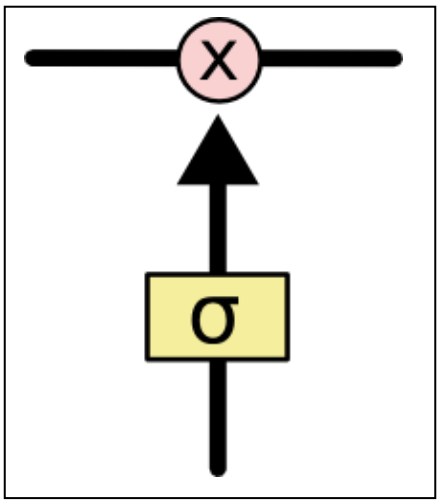


Figure 20: Structured Called Gate [135]

The output of the sigmoid layer are numbers between zero and one. The one of the sigmoid function tells the network that the information goes to the cell state where on the other side zero regulates that no information goes through to the cell state. A LSTM uses three of these gates which controls which information is allowed to go to the cell state.

The first step in a LSTM neural network is to decide which information will be deleted from the cell state. This decision is made, as mentioned above "by a sigmoid layer called forget gate layer"[136].

This layer looks at the previous module (h_{t-1}) and "x_t" and puts out a number between zero and ome to decide which information can be deleted. The next step in the LSTM neural network is to decide which new information will go trough to the cell state. This action is divided into two steps where in the first step a sigmoid layer calls the input layer and decide which value will be updated. In the second step the tanh layer of the LSTM neural network "creates a vector of new candidate values, C_t"[137], which can be added to the cell state. After the two steps the outputs of the two layers will be combined to a product which will be added as an update to the cell state.

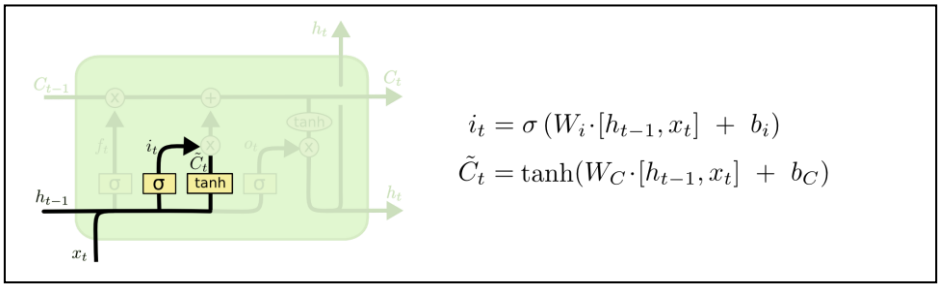


Figure 21: New Values To The Cell State [138]

[134] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[135] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[136] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[137] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[138] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

After the values are added to the cell state the next step is to update the old cell state (C_{t-1}) into the new one (C_t). This update will be handled in the same structure as the new values which come into the neural network. The old state gets multiplied by f_t which deletes unnecessary information then i_t * C_t is added which will create new candidate values and then added to the new cell state.

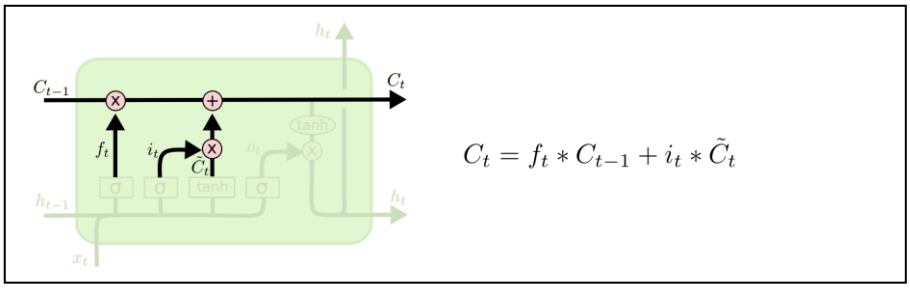


Figure 22: Adding Values Of Old State To The New State [139]

At the end it needs to be decided what the output of the LSTM neural network will be. Therefore, the output is based on the cell state. First the information will be fed to a sigmoid layer to decide which values will be shown in the output. Afterwards the cell state will be put through a tanh layer - this decides which values will have a negative weight and which one will have a positive weight - and multiply it with the output of the sigmoid gate (layer) so the output values will be the one decided by the developer.

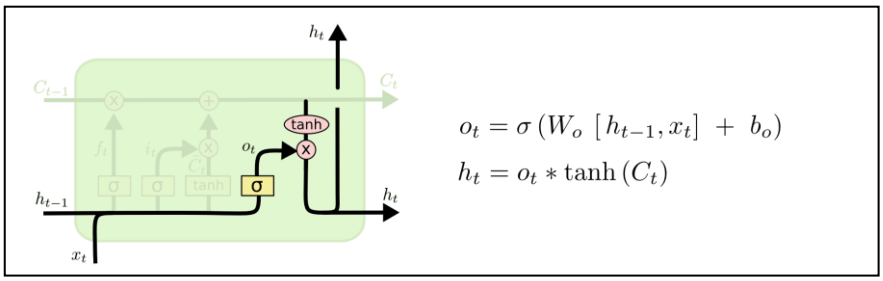


Figure 23: Output Of The LSTM [140]

Therefore, LSTMs are a perfect model for the prediction of stock price movement, because it can handle the time part which is a huge part of this prediction. Another important part is that it can remember important information which lie further in the past and can implement them into the output, since the vanish gradient problem has been eliminated.

[139] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[140] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

5.2.2 Correlation of Price Return and Semantics

An important aspect to understand if there is any impact of news articles to analysis if there is any correlation between the outcome of the sentiment analysis and the stock price return. You can draw first conclusion if there is an impact on the stock price movement trough the correlation analysis. Before starting with the analysis and the preparation of data it needs be explained what correlation is. Correlation is used in statistics to measure to which degree two variables move in relation to each other. The correlation is represented with the correlation coefficient which ranges between minus one and one. The correlation coefficient of one show positive correlation. If positive correlation occurs this shows that if one variable moves "up or down"[141] the other variable moves in the same direction. On the other side a correlation coefficient of minus one shows negative correlation. If two variables have negative correlation to each other means that if one variable increases the other decreases. A correlation coefficient of zero implies no linear relation between the variables.

Firstly, the data must be prepared in a way that the analysis has proper and correct results. For the preparation of the data it is important to calculate first the return of the stock. The term return can be simply explained in the money gained or lost on a stock or other security over a certain period. For the analysis later the money value return was calculated, which means depending on the stock price data set (minutely, hourly, daily), the difference between the current timestamp t (minute, hour, day) and the timestamp -t (-minute, -hour, -day) has been calculated. After the calculation of the return the method of One-Hot Encoding (representation of values in binary form to help algorithm to be more efficient) was used to represent the return in a binary form to calculate the correlation in a more efficient way. If the return was positive the new values becomes one, on the other hand if it is negative or not moving the values becomes zero.

The calculation of the correlation has been done on all three data sets of the stocks (minutely, hourly, daily). Looking on the calculation of the minutely stock price data for all the different stocks (**Table 6** to **Table 23**) there is almost no correlation between the different semantic analyses and the stock price return. The reason for the lack of correlation is due to the lack of news articles which are available for the analysis and the huge amount of stock price data which is available in the minutely stock price data set. This leads to a biased correlation outcome. The tendencies show that with more

[141] https://www.investopedia.com/terms/c/correlation.asp

articles there would be a clearer picture how the analysis of news (semantics) correlate with the return of stocks. Looking into the example of Fiat Chrysler in "**Table 12**" (Appendix) the analysis of Flair on the content and the header of the articles indicates that there can be a negative correlation between the stock return and the semantic analysis. On the other hand the correlation between the analysis of Vader on the header of the articles and the stock return indicates positively correlation but the correlation between the Vader analysis on the article contents and the stock return indicates a negative correlation. Textblob shows another different picture where it is clear that there can be a positive correlation on the header analysis/the content analysis and the stock return. Already with this picture it shows that the different semantic algorithms come to a different conclusion if news is positive, negative, or neutral.

The correlation analysis on the hourly stock price data does not show different results because all the numbers are close to zero as seen in the appendix. The difference between the minutely and the hourly data set is that for the calculation the semantic results where aggregated to fit it to the time gap when the news where published. The numbers of the correlation are slightly larger than the numbers in the minutely data set. Looking at the example of Fiat Chrysler (**Table 32** Appendix) again the numbers of the correlation increased by a 5 %. Another interesting outcome is that Flair indicates now a positive correlation of the analysis of the header of the article and the content of article to the return of the stock where with the minutely data set it was reversed. The Vader analysis of the article content has now a positive correlation where on the minutely it has a slight negative correlation. Textblob has it also reversed while with minutely data the correlation seems to be positive it appears to have a negative correlation on the hourly stock price data.

On the daily data set the outcome of the correlation with example Fiat Chrysler (**Table 52**, Appendix) is the same as on the minutely data set except that the numbers are 85% higher. All the numbers are still close to zero like the analysis of Textblob on the header where the correlation is at -0,0638. Staying at the example of Fiat Chrysler the analysis shows more significant results where the correlation goes up to 0,1. This result shows almost no correlation but the tendencies towards a positive correlation.

Having a clearer picture of the correlation between the securities return and the semantics of the news it is of great importance to collect a much larger number of articles to draw first conclusion what kind of impact news can have on the trend of a public tradable security.

5.2.3 Prediction of Stock Price Movement

The influence of news articles on the stock price movement was analysed with four different algorithms. These algorithms are LSTM, a RandomForest base model, a RandomForest feature model and a XGBoost. For the algorithm to interpret the given data it needs to be formatted properly. Mainly the data needs to be normalized, which was performed by the tool MinMax Scaler from the python library scikit-learn. Scikit-learn is a library for machine learning algorithm but it can also be used to preprocess data to bring it in a form that it can fit it different algorithms. The MinMax Scaler of scikit-learn transforms features "by scaling each feature" to a given range"[142]. This preprocessing algorithm "scales and translates"[143] each data point individually so it is in a certain given range for example between zero and one. Before the data can be normalized it has to be divided into training, validation, and test dataset. The training dataset is firstly used to train the algorithm. The validation dataset is a sample of data which is used to estimate the skills of a model while tuning the hyperparameters of the algorithm. The test dataset is the sample of the data which is used to give an "unbiased estimate of the skill of the final tuned model"[144]. The results of the test dataset are used to compare models and select which gives the best prediction.

The next step is to create the target value (y) and the time windows (X) for modeling. The term windows in time series analysis, as explained at the beginning of this chapter, is used to bring the data into a form to predict the next time step with the use of the prior time steps. For the different data sets (minutely, hourly, daily) the sizes of the time windows and the forecast distance are defined differently. For the minutely data set the window size is set to 60 minutes for the prediction, the forecast distance is set to 30 minutes. The reason for the smaller windows size is the dependency of a short time prediction if minutely or live data is used in a real-world setting. The time window for the hourly data set is set 45 hours for the prediction and the forecast distance is set to 9 hours. The 9 hours are used to verify how the stock price will be the next day, since the stock exchange is open for 9 hours. The reason to choose 45 hours as the time window values is to predict the stock price movement 5 days advanced. For the last data set, the daily data set, the values for the time window and the forecast distance are 30 and 5 days. 30 days is chosen to predict the movement of the stock price one month ahead and the

[142] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[143] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[144] https://machinelearningmastery.com/difference-test-validation-datasets/

forecast distance will show how the stock price is in one week. The selection for these values comes in the perspective of a private investor. Most private investors invest their money for a larger time horizon.

5.2.3.1 LSTM prediction

The prediction of the stock price development is going to be calculate separately for every sentiment analysis and the analysis without sentiment analyses. Why this way is chosen, is to see which sentiment analysis of the three different sentiment algorithms have the biggest influence, and on which stock the algorithm can predict if news have a bigger influence on some stocks then on others from the list which were chosen for this analysis.

The LSTM structure can be different from programmer to programmer. In this paper the python library Keras was used to build the neural network. Keras was especially developed to build neural networks in an efficient way in python.

For the analysis of this paper the hyperparameters are chosen for the LSTM of the minutely data set as seen in **Table 4**. In this example it was chosen to use two LSTM layers with the size of 75 and 40. It is of course reasonable to use more layers or use a bigger size of each layer which tend the algorithm run longer because it has do more calculations. The dropout rate helps preventing the algorithm to overfit and in this example, it is chosen to be 0.1. A lower dropout rate tend to still overfit an algorithm as a higher one but the example should overfit a bit to see if there is an impact of the news on the price prediction because the amount of news is lower than the amount of stock price data. For the optimizer Adam is chosen as a valuable optimizer function and for the loss function the Mean-Absolute-Error provides the best outcome. Epochs means how many times the algorithm goes over the entire training dataset. Due to the huge amount of stock price data the value of epochs is chosen relatively small to have a faster result of the algorithm.

First LSTM Layer	Second LSTM Layer	Dropout Rate	Optimizer	Loss	Epochs	Batch_Size
75	40	0.1	Adam	Mean_Absolute_Error	10	32

Table 4: LSTM Parameters For Minutely Data Set

Going further to the evaluating and the outcome of the algorithm the example of Fiat Chrysler is chosen as in the correlation chapter. As shown clearly in **Figure 22** news can have an impact on the price prediction of securities. In the **Figure 22** it shows that the predictions are either above or below the prediction when not using any semantics.

If the prediction is accurate it is important to check the RMSE (Root Mean Squared Error). The RMSE is the "standard deviation of the residuals"[145]. The residuals show how far from the regression line the data points are[146]. It shows how concentrate the data is around the line of the best fit[147]. The smaller the RMSE is the better is the prediction of the algorithm. In the case of Fiat Chrysler on the minutely data set the RMSE is around 0.34 (**Table 5**) which already is a good result for the small amount of semantic data and the big amount of stock price data.

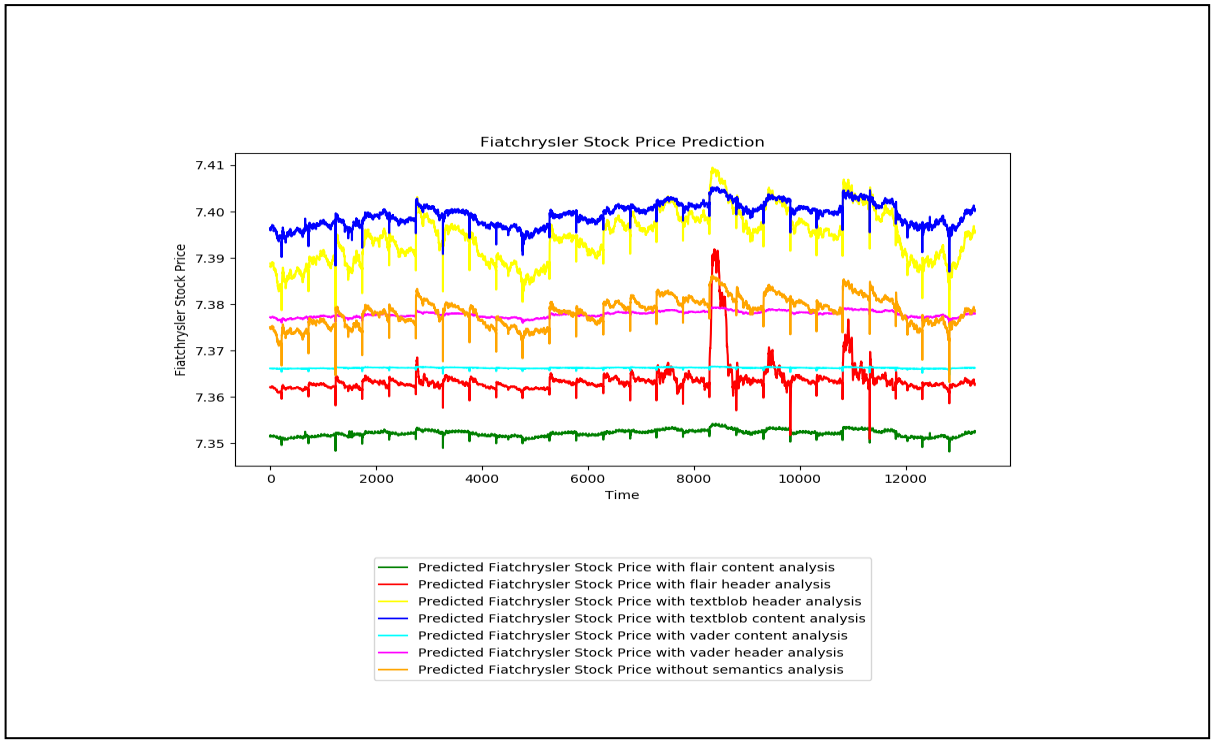


Figure 24: Stock Price Prediction Of Fiat Chrysler With Minutely Stock Price Data Using LSTM

[145] https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/
[146] https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/
[147] https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/

RMSE flair content	RMSE flair header	RMSE text-blob content	RMSE text-blob header	RMSE vader content	RMSE vader header	RMSE without-semantics
0.3418	0.3417	0.3372	0.3388	0.3411	0.337	0.3377

Table 5: RMSE Fiat Chrysler On Minutely Stock Price Data

The LSTM structure for the daily data set does not vary much to the structure of the minutely data set. The only difference is the epochs variable. It is set to 50 to see if the loss function declines.

First LSTM Layer	Second LSTM Layer	Dropout Rate	Optimizer	Loss	Epochs	Batch_Size
75	40	0.1	Adam	Mean_Absolute_Error	50	32

Table 6: LSTM Parameters For Daily Data Set

Looking at the loss as seen in **Figure 23** the loss declines over time which is the way it supposes to do in the best case. The training data shows a better course in comparison of the validation data. This is due to the few data points in this data set. But with different sets of the hyperparameters the course of the loss function of the validation data the up and down movements can be flattened.

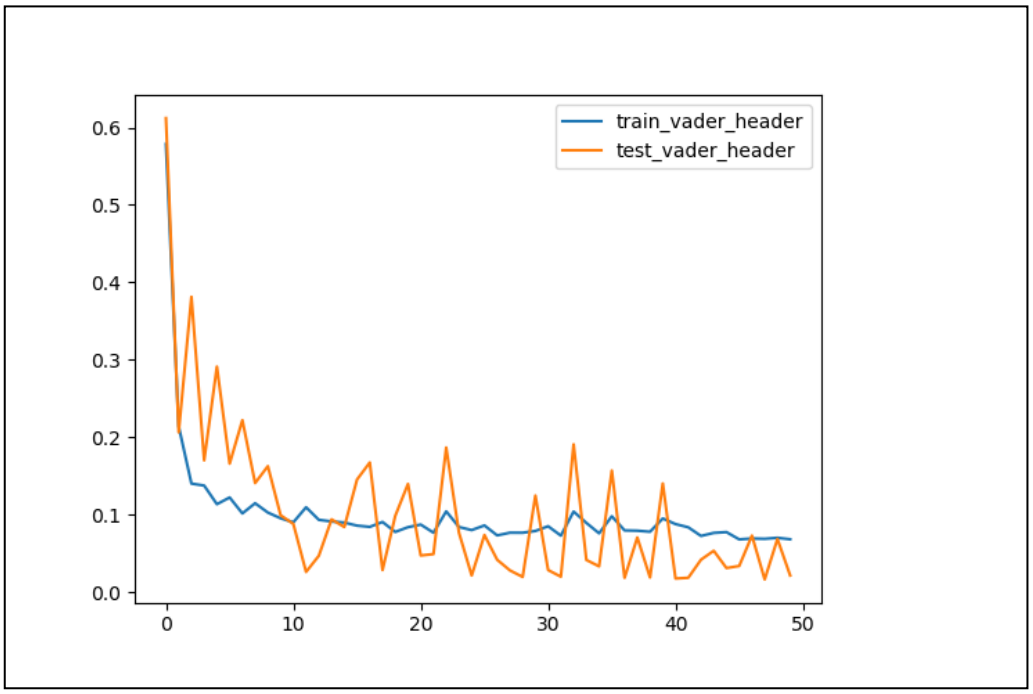


Figure 25: Loss Of Vader Prediction On Fiat Chrysler Daily Data Set

The small RMSE as seen in **Table 7** shows the prediction is close to regression line which means the data does not spread far from the line of the best fit. This indicates with daily stock price data the algorithm can make more precise predictions.

RMSE flair content	RMSE flair header	RMSE textblob content	RMSE textblob header	RMSE vader content	RMSE vader header	RMSE withoutsemantics
0.1163	0.1289	0.1338	0.1144	0.1204	0.1287	0.1159

Table 7: RMSE Fiat Chrysler On Daily Stock Price Data

The prediction in **Figure 79** shows as well that the LSTM easily predicts the difference when using semantics or not.

For the last data set the hourly data, the structure of the LSTM is the same as of the LSTM from the daily data set as seen in **Table 8**.

First LSTM Layer	Second LSTM Layer	Dropout Rate	Optimizer	Loss	Epochs	Batch_Size
75	40	0.1	Adam	Mean_Absolute_Error	50	32

Table 8: LSTM Parameters For Hourly Data Set

The loss function for the hourly data is declining as well as seen in **Figure 24**. The difference is that it does not decline as smooth as for the daily data set, especially the training data where with amending the hyperparameters the course of the loss function can be improved.

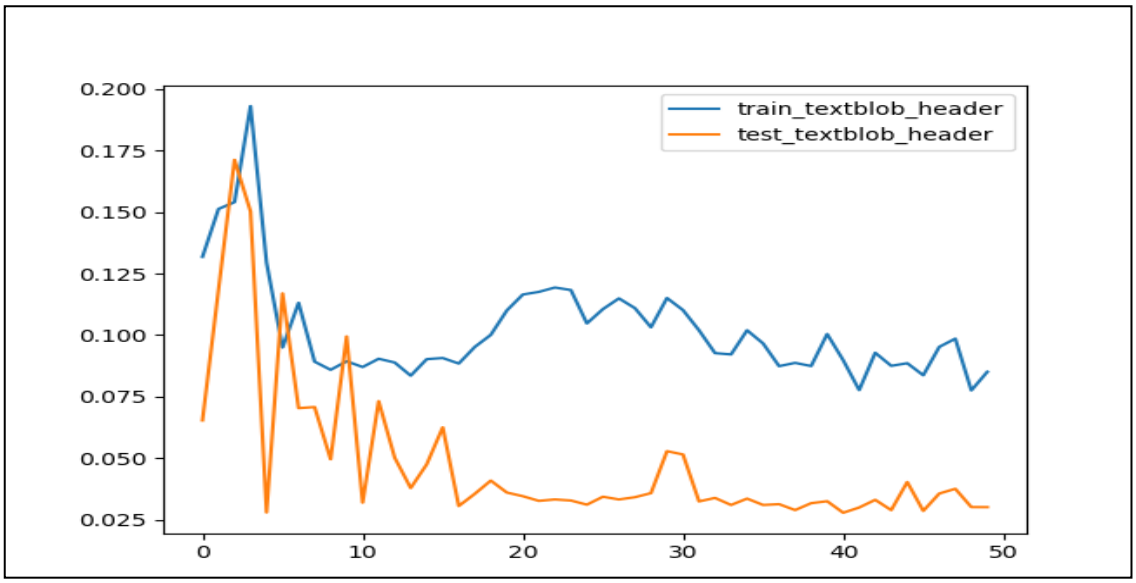


Figure 26: Loss Function Course Of Textblob Header Fiat Chrysler

RMSE flair content	RMSE flair header	RMSE text-blob con-tent	RMSE text-blob hea-der	RMSE vader con-tent	RMSE vader hea-der	RMSE wit-houtseman-tics
0.1679	0.1621	0.1639	0.1667	0.1642	0.1731	0.1627

Table 9: RMSE Fiat Chrysler On Hourly Stock Price Data

The RMSE of the hourly data shows that the data also does not spread far around the line of the best fit which is a good indication that the algorithm predicts the stock prices well.

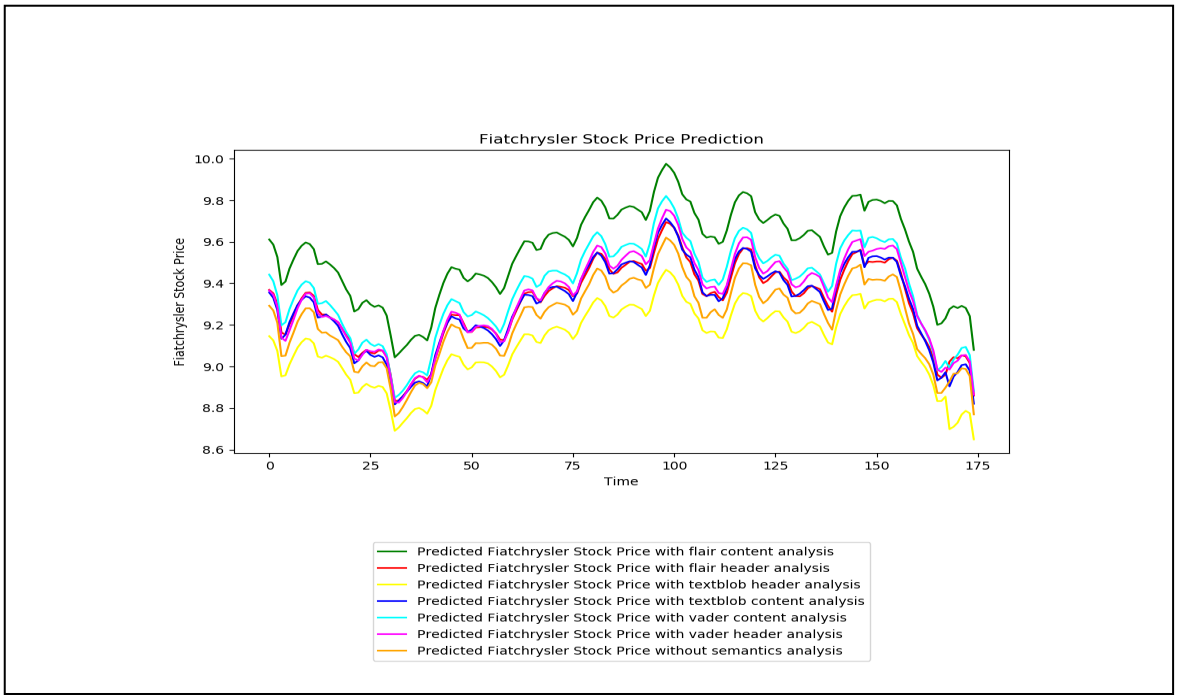


Figure 27: Stock Price Prediction Of Fiat Chrysler With Hourly Stock Price Data Using LSTM

As seen in the **Figure 25** the LSTM clearly shows the differences between the prediction with semantics and without.

5.2.3.2 Random Forest Prediction

On the outcome of the Random Forest it clearly shows that there are differences between the prediction with semantics and without for the daily data set and the hourly data set as seen in the figures in the appendix. Looking at the outcome on the minutely data set as seen in **Figure 28** the differences are barely recognisable.

Comparing the outcome of the RMSE it clearly shows that for the daily data set LSTM model predicts the better forecast of the stock price movement as comparing the numbers in **Table 7** and **Table 10** although the numbers are very close to each other.

RMSE flair content	RMSE flair header	RMSE textblob content	RMSE textblob header	RMSE vader content	RMSE vader header	RMSE withoutsemantics
0.1931	0.193	0.193	0.2121	0.193	0.1977	0.2061

Table 10: RMSE Random Forest Fiat Chrysler Daily Data Set

Comparing the outcome of the RMSE on the hourly data set and the minutely data set it shows a particularly good RMSE but at the same time it indicates that the model at this point is heavily overfitted. Therefore, the hyperparameters have to be adjusted.

RMSE flair content	RMSE flair header	RMSE textblob content	RMSE textblob header	RMSE vader content	RMSE vader header	RMSE withoutsemantics
0.0323	0.0322	0.0315	0.0311	0.0312	0.0316	0.0304

Table 11: RMSE Random Forest Fiat Chrysler Hourly Data Set

RMSE flair content	RMSE flair header	RMSE textblob content	RMSE textblob header	RMSE vader content	RMSE vader header	RMSE withoutsemantics
0.0231	0.0236	0.023	0.0234	0.0234	0.0233	0.0227

Table 12: RMSE Random Forest Fiat Chrysler Minutely Data Set

As mentioned already for the minutely data set and the hourly data set there is barely any difference recognizable.

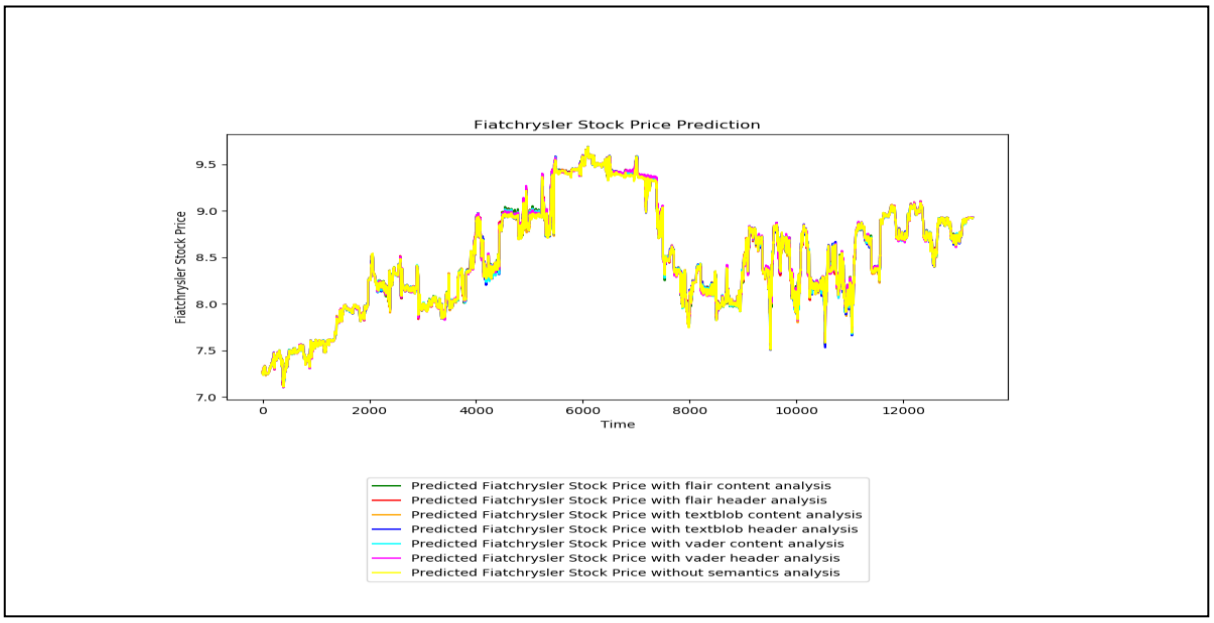


Figure 28: Stock Price Prediction of Fiat Chrysler With Minutely Stock Price Data Using Random-Forest Base Model

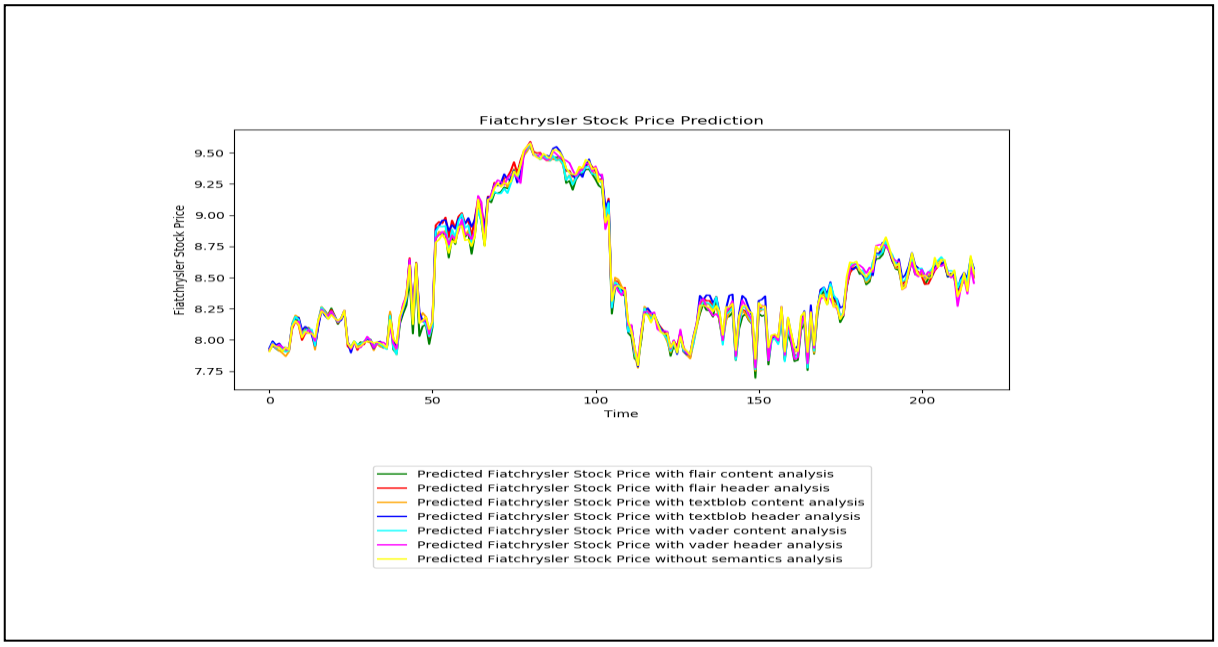


Figure 29: Stock Price Prediction of Fiat Chrysler With Hourly Stock Price Data Using Random-Forest Base Model

The only time difference between the analyses are recognizable is when the algorithm is trying to forecast based on the daily data set as seen in **Figure 30** below.

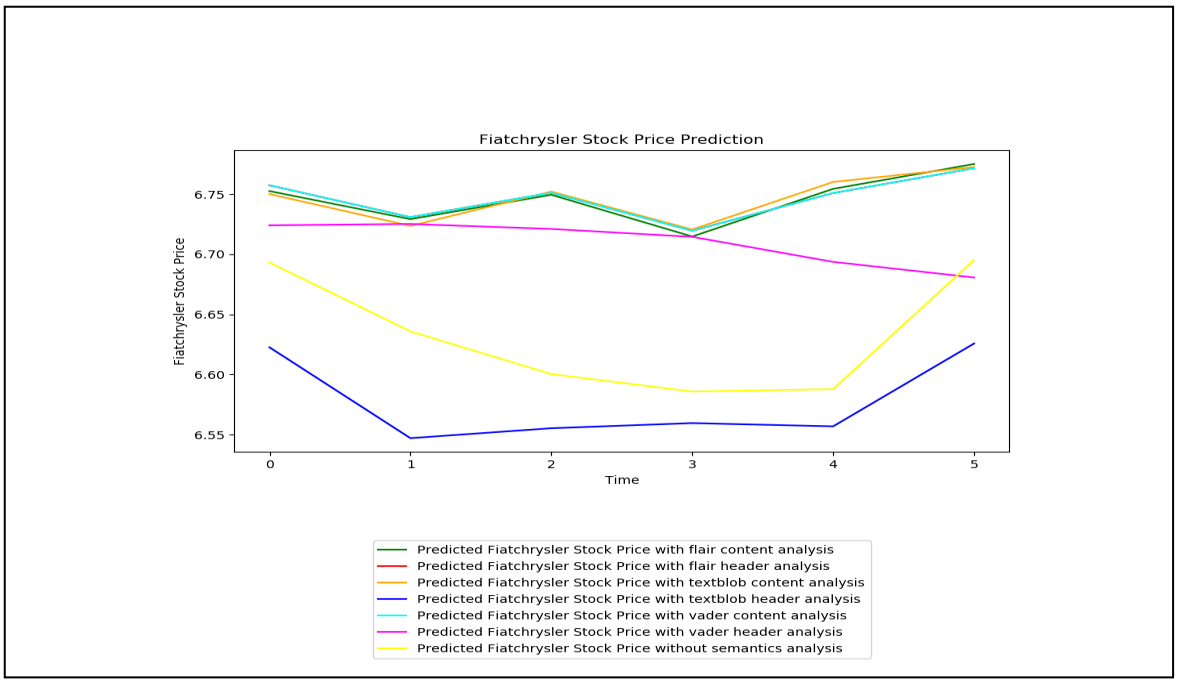


Figure 30: Stock Price Prediction of Fiat Chrysler With Daily Stock Price Data Using Random-Forest Base Model

Clearer differences are recognizable in the hourly data set is when the data set is fed to a Random Forest Feature model as seen in **Figure 81** in the appendix. This is explainable due to the additional features which help the algorithm to make better predictions at the end.

5.2.3.3 XGBoost Prediction

For the prediction of the stock price movement with the XGBoost algorithm shows that there is not much difference to the Random Forest. The XGBoost shows the differences between the prediction with and without semantics clearly as seen in the figures in the appendix. When predicting the movement for the minutely data there is as well barely any difference recognisable.

Looking at the outcome of the RMSE from the XGBoost for the daily data set it is recognisable that the algorithm does worse prediction than the LSTM and the Random Forest base model.

RMSE flair content	RMSE flair header	RMSE textblob content	RMSE textblob header	RMSE vader content	RMSE vader header	RMSE withoutsemantics
0.2129	0.2129	0.2129	0.2129	0.2136	0.2129	0.2129

Table 13: RMSE Random Forest Fiat Chrysler Daily Data Set

The graph of the prediction of the daily data set of Fiat Chrysler shows that all of the prediction goes the same direction as in the Random Forest prediction. On the other side the LSTM prediction shows a completely different outcome where the prediction are clearly distinguishable.

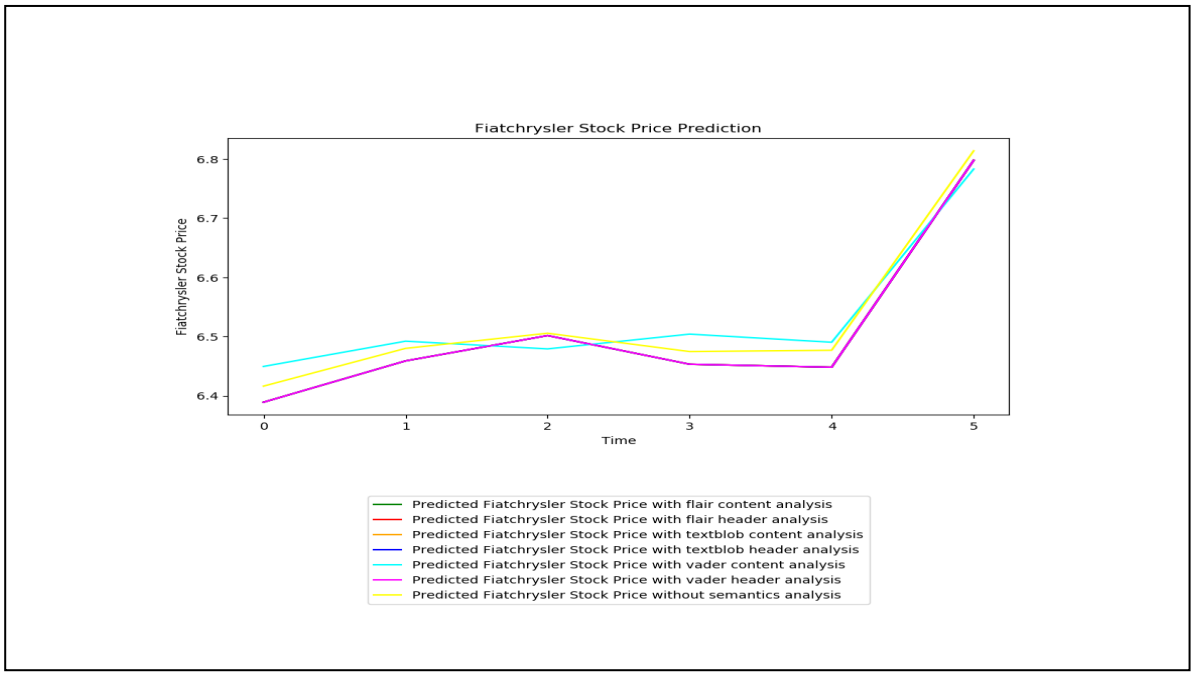


Figure 31: Stock Price Prediction Of Fiat Chrysler With Daily Stock Price Data Using XGBoost

RMSE flair content	RMSE flair header	RMSE text-blob content	RMSE text-blob header	RMSE vader content	RMSE vader header	RMSE without semantics
0.123	0.124	0.123	0.1274	0.1285	0.1274	0.1279

Table 14: RMSE Random Forest Fiat Chrysler Hourly Data Set

Comparing the outcome to the Random Forest base model and the LSTM the XGBoost RMSE is better then the LSTM but worse than the Random Forest. For the Random Forest prediction there has to come hyperparameters adaption to make better predictions.

RMSE flair content	RMSE flair header	RMSE text-blob content	RMSE text-blob header	RMSE vader content	RMSE vader header	RMSE without semantics
0.016	0.016	0.016	0.016	0.016	0.016	0.016

Table 15: RMSE Random Forest Fiat Chrysler Minutely Data Set

The result for the minutely data set shows the same results as for the Random Forest. This kind of results shows that the XGBoost and the Random Forest has its problems when there are, as in this case, too many times series data points. Also, the graph of the prediction shows that there are not many differences between the prediction with semantics and without semantics recognisable.

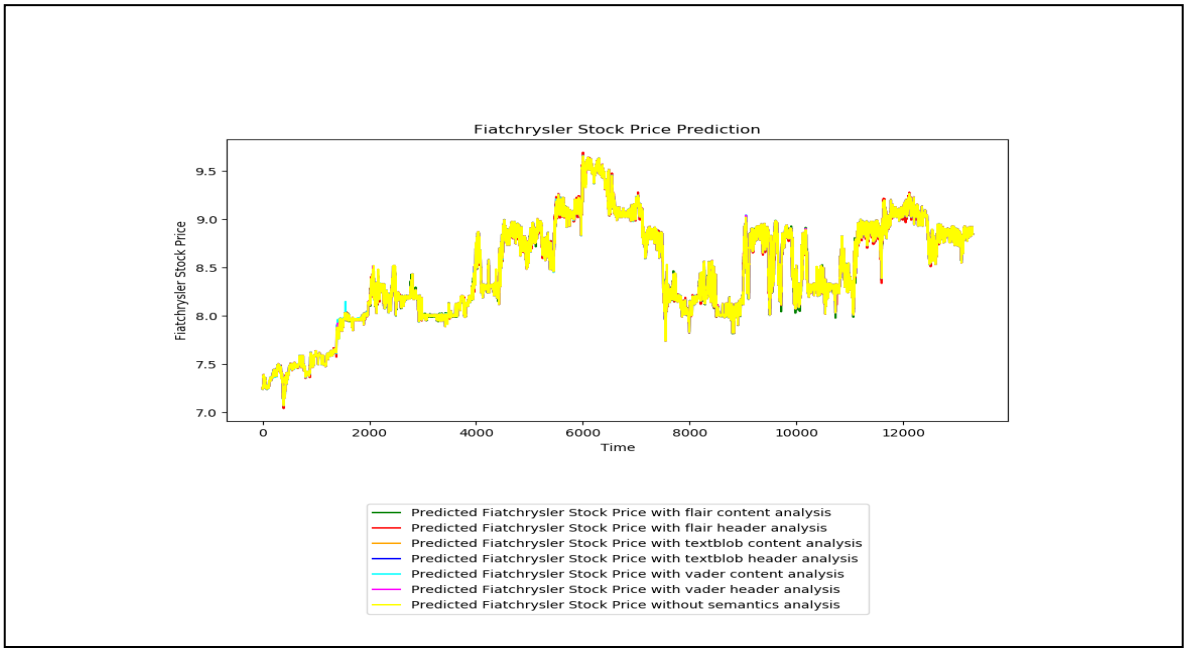


Figure 32: Stock Price Prediction Of Fiat Chrysler With Minutely Stock Price Data Using XGBoost

6 Conclusion

Stock price trend prediction with the use of semantic analysis can be done by different algorithms. The results show that it is from great importance to analyse the content of the whole article an not just take the header into account. As seen in the analysis of the paper the semantic algorithms of Vader, Flair, and Textblob show that the header can be negative or positive where the content is the opposite. Therefore, just using the header for the analysis can lead to a wrong prediction of the stock price trend. The semantic analysis has to be done with caution in this regard.

The use of different algorithms for the trend prediction clearly shows that the LSTM can handle the time series issue better than the Random Forest and XGBoost. Using an LSTM for trend prediction of public traded stocks shows the most accurate result. This paper already shows that there are trends of the stock price visible for the different stocks, but this has to be looked at with caution. The reason therefore is that the amount of stock price data predominates the amount of sentiment data. For a clearer and more accurate result there should be at least one sentiment data point for each time value of the stock price data. This can be difficult to achieve because it needs a lot of research and analysis. As seen in the appendix some stocks show a bigger impact of the semantics but to clearly tell if some stocks are impacted more by the news it is necessary to have the same amount of semantic data for each stock.

Furthermore, data from Thomson Reuters is reliable and can be used rather well for the analysis but having data from only one data source is very biased, data from more than one source is necessary to have an even more accurate prediction, since news in it self can be biased and the interpretation of an event can differ from article to article.

At the end can be said that the trend of a stock price can be predicated using different algorithms with help of the semantic analysis of financial market news and the most accurate results in the setting of this paper can be produced by LSTM.

7 Outlook

For further analysis of this issues and to have more reliable result, different factors has to be taken into consideration. First, as already mentioned, one data source of financial news is not enough to counteract a potential bias. Collecting the news from multiple sources can boost up the accuracy of the algorithm. Also just using the data from one financial news issuer may not cover every aspect. Today social media is an important tool for information. And even unproven opinions may influence the stock prices as they do with political opinions. Therefore, it is important to also analyse social media posts because of there influence.

Another aspect is to analyse if the news just moves the market up or down, or do they stop the current trend the market has. It is also important to see what happened before the news has been released and what kind of market information are already include in the market price.

For future analysis it is important to check if every semantic analysis has the same weight. If every semantic analysis which is used by the algorithm to predict stock market trends has the same weight, unimportant news might be overrepresented in their power towards the prediction. So, it must be analysed if every released news have the same impact.

It should also be look into the fact, that not only company specific news does have an impact on the specific stock price but also general stock market news. So, is must be taken into consideration how these general news influences the stock market price.

For a closer look into the code please contact me at victor.johannes.holl@gmail.com so I can grant you access to my repository.

Statement of Certification

I hereby confirm that this thesis constitutes my own work, produced without aid and support from persons and/or materials other than the ones listed. Quotation marks indicate direct language from another author. Appropriate credit is given where I have used ideas, expressions or text from another public or non-public source.
The paper in this or similar form has never been submitted as an assessed piece of work in or outside of Germany. It also has not yet been published.

2020-09-28

City, Date | Signature

List of Literature

Literature for Thesis information:

- A Beginner's Guide to LSTMs and Recurrent Neural Networks. (n.d.). Retrieved from https://wiki.pathmind.com/lstm

- Audi Aktie | Firmenprofil | Termine | 675700 | DE0006757008. (2020, September 25). Retrieved from https://www.onvista.de/aktien/unternehmensprofil/Audi-Aktie-DE0006757008

- Azhikannickel, J. (2019, December 4). Time Series Analysis on Stock Market Forecasting(ARIMA & Prophet). Retrieved from https://medium.com/@josephabraham9996/time-series-analysis-on-stock-market-forecasting-arima-prophet-2b60cacf604

- Brownlee, J. (2019, August 6). How to Configure the Number of Layers and Nodes in a Neural Network. Retrieved from https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/

- Brownlee, J. (2020a, April 21). A Gentle Introduction to XGBoost for Applied Machine Learning. Retrieved from https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

- Brownlee, J. (2020b, August 14). Linear Regression for Machine Learning. Retrieved from https://machinelearningmastery.com/linear-regression-for-machine-learning/

- Brownlee, J. (2020c, August 14). What is the Difference Between Test and Validation Datasets? Retrieved from https://machinelearningmastery.com/difference-test-validation-datasets/

- Brownlee, J. (2020d, August 20). Supervised and Unsupervised Machine Learning Algorithms. Retrieved from https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

- Daimler Aktie | Firmenprofil | Termine | 710000 | DE0007100000. (2020, September 25). Retrieved from https://www.onvista.de/aktien/unternehmensprofil/Daimler-Aktie-DE0007100000

- Flair: State-of-the-Art Natural Language Processing (NLP). (n.d.). Retrieved from https://research.zalando.com/welcome/mission/research-projects/flair-nlp/

- flairNLP/flair. (n.d.). Retrieved from https://github.com/flairNLP/flair

- Fundamentalanalyse - das Wirtschaftslexikon .com. (n.d.). Retrieved from http://www.daswirtschaftslexikon.com/d/fundamentalanalyse/fundamentalanalyse.htm

- Goddard, C. (2011). Semantic Analysis. Retrieved from https://books.google.de/books?hl=de&lr=&id=XW4WL3mKjjkC&oi=fnd&pg=PP2&dq=semantic+analysis+definition&ots=X3mGz-LWsL&sig=QpAVw3dJEkO72avw__ejcCVlkQY#v=onepage&q&f=false

- Gupta, M. (2018, September 13). ML | Linear Regression. Retrieved from https://www.geeksforgeeks.org/ml-linear-regression/

- Hayes, A. (2020a, March 16). Technical Analysis. Retrieved from https://www.investopedia.com/terms/t/technicalanalysis.asp

- Hayes, A. (2020b, May 24). Correlation. Retrieved from https://www.investopedia.com/terms/c/correlation.asp

- Holl, J. (n.d.). Master Thesis. Retrieved from https://github.com/JohannesHoll/Master_Thesis

- HTML tree wordy navigation | Python. (n.d.). Retrieved from https://campus.datacamp.com/courses/web-scraping-with-python/introduction-to-html?ex=4

- Joshi, K. (2016, July). *STOCK TREND PREDICTION USING NEWS SENTIMENT ANALYSIS*. Retrieved from https://arxiv.org/ftp/arxiv/papers/1607/1607.01958.pdf

- Khuong, B. (2020, September 18). The Basics of Recurrent Neural Networks (RNNs) - Towards AI — Multidisciplinary Science Journal. Retrieved from https://medium.com/towards-artificial-intelligence/whirlwind-tour-of-rnns-a11effb7808f

- Klein, J. (2018, June 3). Neural Networks Part 1: A Simple Proof of the Universal Approximation Theorem. Retrieved from https://blog.goodaudience.com/neural-networks-part-1-a-simple-proof-of-the-universal-approximation-theorem-b7864964dbd3

- Le, J. (2020, May 10). Neural Networks 101 - Cracking The Data Science Interview. Retrieved from https://medium.com/cracking-the-data-science-interview/neural-networks-101-ee21cd508499

- Li, X. (2011, August 29). Improving Stock Market Prediction by Integrating Both Market News and. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-23091-2_24

- Liew, L. (2020, July 7). Sentiment Analysis with Python - A Beginner's Guide - AlgoTrading101 Blog. Retrieved from https://algotrading101.com/learn/sentiment-analysis-python-guide/

- Litzel, N. (2019, March 19). Was ist ein Long Short-Term Memory? Retrieved from https://www.bigdata-insider.de/was-ist-ein-long-short-term-memory-a-774848/

- Liu, B. (2012, May). Sentiment Analysis and Opinion Mining. Retrieved from https://www.morganclaypool.com/doi/abs/10.2200/s00416ed1v01y201204hlt016

- Michael, M. (2014, August 14). Sentiment And Semantic Analysis. Retrieved from https://www.digital-mr.com/blog/view/sentiment-and-semantic-analysis

- Ng, Y. (2019, October 3). Machine Learning Techniques applied to Stock Price Prediction. Retrieved from https://towardsdatascience.com/machine-learning-techniques-applied-to-stock-price-prediction-6c1994da8001

- NLP | Part of Speech - Default Tagging. (2019, October 28). Retrieved from https://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging/

- Nordquist, R. (2019, September 18). What Is Syntactic Ambiguity? Retrieved from https://www.thoughtco.com/syntactic-ambiguity-grammar-1692179

- P. (2020, June 11). How Sentiment Analysis Is Influencing Today's Market? Retrieved from https://www.analyticsinsight.net/sentiment-analysis-influencing-todays-market/

- Pandey, P. (2020, September 11). Simplifying Sentiment Analysis using VADER in Python (on Social Media Text). Retrieved from https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f

- Python | Sentiment Analysis using VADER. (2019, January 23). Retrieved from https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/

- RMSE: Root Mean Square Error. (2020, July 6). Retrieved from https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/

- Schmelzer, R. (2019, August 27). AI Making Waves In News And Journalism. Retrieved from https://www.forbes.com/sites/cognitiveworld/2019/08/23/ai-making-waves-in-news-and-journalism/#5d417d677748

- Scrapy 2.3 documentation — Scrapy 2.3.0 documentation. (2020, March 3). Retrieved from https://docs.scrapy.org/en/latest/

- Shah, D. (n.d.). Predicting the Effects of News Sentiments on the Stock Market. Retrieved from https://arxiv.org/ftp/arxiv/papers/1812/1812.04199.pdf

- Singh, H. (2018, November 4). Understanding Gradient Boosting Machines - Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

- sklearn.preprocessing.MinMaxScaler. (2020, August 4). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

- Sohangir, S. (2018, January 25). Big Data: Deep Learning for financial sentiment analysis. Retrieved from https://link.springer.com/article/10.1186/s40537-017-0111-6?error=cookies_not_supported&code=f8e36468-584a-45d9-bf03-7718da7eb365

- Team, E. S. (2020, May 7). Natural Language Process semantic analysis: definition. Retrieved 3 March 2020, from https://expertsystem.com/natural-language-process-semantic-analysis-definition/

- Terry-Jack, M. (2019, May 2). NLP: Pre-trained Sentiment Analysis - Mohammed Terry-Jack. Retrieved from https://medium.com/@b.terryjack/nlp-pre-trained-sentiment-analysis-1eb52a9d742c

- TextBlob: Simplified Text Processing — TextBlob 0.16.0 documentation. (n.d.). Retrieved from https://textblob.readthedocs.io/en/dev/

- Tilgner, M. (2019, October 22). Time series forecasting with random forest. Retrieved from https://www.statworx.com/at/blog/time-series-forecasting-with-random-forest/

- Twinword, T. (2020, September 23). Sentiment Analysis For Finance. Retrieved from https://www.twinword.com/blog/sentiment-analysis-for-the-financial-sector/

- Understanding LSTM Networks. (2015, August 27). Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

- What is HTML. (n.d.). Retrieved from https://www.w3schools.com/whatis/whatis_html.asp

- Wood, T. (2020, September 10). Random Forests. Retrieved from https://deepai.org/machine-learning-glossary-and-terms/random-forest

- Yin, L. (2018, December 21). Explain FeedForward and BackPropagation - Machine Learning for Li. Retrieved from https://medium.com/machine-learning-for-li/explain-feedforward-and-backpropagation-b8cdd25dcc2f

Literature for Programming:

- Akbik, A. (n.d.). Flair: State-of-the-Art Natural Language Processing (NLP). Retrieved from https://research.zalando.com/welcome/mission/research-projects/flair-nlp/

- Applying Spacy Parser to Pandas DataFrame w/ Multiprocessing. (2017, June 6). Retrieved from https://stackoverflow.com/questions/44395656/applying-spacy-parser-to-pandas-dataframe-w-multiprocessing

- Better way to create json file from multiple lists? (2018, May 25). Retrieved from https://stackoverflow.com/questions/50519818/better-way-to-create-json-file-from-multiple-lists

- Export csv file from scrapy (not via command line). (2014, August 6). Retrieved from https://stackoverflow.com/questions/25163023/export-csv-file-from-scrapy-not-via-command-line

- flairNLP/flair. (n.d.). Retrieved from https://github.com/flairNLP/flair

- Hi. I want to fetch historical 1 minute intraday prices for a portfolio of European stocks via API. Which API is the most best for this task? Thanks - Forum | Refinitiv Developer Community. (2018, October 19). Retrieved from

https://community.developers.refinitiv.com/questions/33524/hi-i-want-to-fetch-historical-1-minute-intraday-pr.html

- How can I use the fields_to_export attribute in BaseItemExporter to order my Scrapy CSV data? (2013, December 24). Retrieved from https://stackoverflow.com/questions/20753358/how-can-i-use-the-fields-to-export-attribute-in-baseitemexporter-to-order-my-scr

- How do I construct a UTC `datetime` object in Python? (2015, February 13). Retrieved from https://stackoverflow.com/questions/28498163/how-do-i-construct-a-utc-datetime-object-in-python

- How to create a filename with the current date and time in python when query is ran. (2017, October 12). Retrieved from https://stackoverflow.com/questions/46705867/how-to-create-a-filename-with-the-current-date-and-time-in-python-when-query-is

- How to ignore days with no data while looping over a certain date range? - Forum | Refinitiv Developer Community. (2020, July 11). Retrieved from https://community.developers.refinitiv.com/questions/63014/how-to-ignore-days-with-no-data-while-looping-over.html

- Item Exporters — Scrapy 2.2.1 documentation. (2020, June 14). Retrieved from https://docs.scrapy.org/en/latest/topics/exporters.html

- Item Pipeline — Scrapy 0.9 documentation. (2010, April 2). Retrieved from https://docs.scrapy.org/en/0.9/topics/item-pipeline.html#activating-a-item-pipeline-component

- Jones, R. (2020, June 3). Sentiment Anaylsis with the flair NLP library - Riley Jones. Retrieved from https://medium.com/@rileymjones/sentiment-anaylsis-with-the-flair-nlp-library-cfe830bfd0f4

- „list“ object has no attribute „shape“. (2014, January 9). Retrieved from https://stackoverflow.com/questions/21015674/list-object-has-no-attribute-shape

- Li, S. (2019, January 10). Introducing TextBlob - Towards Data Science. Retrieved from https://towardsdatascience.com/having-fun-with-textblob-7e9eed783d3f

- Mwiti, D. (n.d.). Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices. Retrieved from https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html

- N. (n.d.). NGYB/Stocks. Retrieved from https://github.com/NGYB/Stocks/blob/master/StockPricePrediction/StockPricePrediction_v4a_lstm.ipynb

- Pattnaik, S. (2019, December 25). Step-by-Step Guide for building Sentiment Analysis model using Flask/Flair. Retrieved from https://beingdatum.com/sentiment-analysis-flask-flair/

- Python Scrapy: How to get CSVItemExporter to write columns in a specific order. (2011, August 4). Retrieved from https://stackoverflow.com/questions/6943778/python-scrapy-how-to-get-csvitemexporter-to-write-columns-in-a-specific-order

- Python Scrapy Print Item Keys as Header in CSV. (2016, August 11). Retrieved from https://stackoverflow.com/questions/38886499/python-scrapy-print-item-keys-as-header-in-csv

- Pythonically add header to a csv file. (2013, December 3). Retrieved from https://stackoverflow.com/questions/20347766/pythonically-add-header-to-a-csv-file

- Rao, P. (2019, September 9). Fine-grained Sentiment Analysis in Python (Part 1) - Towards Data Science. Retrieved from https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-1-2697bb111ed4

- RegEx for matching dates (Month Day, Year OR m/d/yy). (2019, May 9). Retrieved from https://stackoverflow.com/questions/56065683/regex-for-matching-dates-month-day-year-or-m-d-yy

- Regex for time format HH:MM AM/am/PM/pm in python. (2018, March 11). Retrieved from https://stackoverflow.com/questions/49217248/regex-for-time-format-hhmm-am-am-pm-pm-in-python

- Regex to match month name followed by year. (2010, April 16). Retrieved from https://stackoverflow.com/questions/2655476/regex-to-match-month-name-followed-by-year

- Regular expression for matching HH:MM time format. (2011, September 24). Retrieved from https://stackoverflow.com/questions/7536755/regular-expression-for-matching-hhmm-time-format/7536768

- Save a csv file in python with datetime as filename. (2018, March 5). Retrieved from https://stackoverflow.com/questions/49105693/save-a-csv-file-in-python-with-datetime-as-filename

- Saving the output of spider in a variable rather than in a file. (2018, February 1). Retrieved from https://stackoverflow.com/questions/48573298/saving-the-output-of-spider-in-a-variable-rather-than-in-a-file

- Scrapy - How to export a cvs file with item key in header. (2017, October 4). Retrieved from https://stackoverflow.com/questions/46565855/scrapy-how-to-export-a-cvs-file-with-item-key-in-header

- Scrapy - run at time interval. (2018, July 31). Retrieved from https://stackoverflow.com/questions/51610838/scrapy-run-at-time-interval

- scrapy csvpipeline to export csv according to spiders name or id. (2017, April 8). Retrieved from https://stackoverflow.com/questions/43289927/scrapy-csvpipeline-to-export-csv-according-to-spiders-name-or-id

- scrapy duplicate filter with csv file. (2013, August 1). Retrieved from https://stackoverflow.com/questions/17984822/scrapy-duplicate-filter-with-csv-file

- Scrapy: Extract links and text. (2015, January 3). Retrieved from https://stackoverflow.com/questions/27753232/scrapy-extract-links-and-text

- Scrapy: one row per item. (2013, January 14). Retrieved from https://stackoverflow.com/questions/14317530/scrapy-one-row-per-item

- Scrapy pipeline to export csv file in the right format. (2015, April 29). Retrieved from https://stackoverflow.com/questions/29943075/scrapy-pipeline-to-export-csv-file-in-the-right-format

- Scrapy save URLs titles in text file. (2015, April 2). Retrieved from https://stackoverflow.com/questions/29420892/scrapy-save-urls-titles-in-text-file

- Selectors — Scrapy 2.2.1 documentation. (2020, March 20). Retrieved from https://docs.scrapy.org/en/latest/topics/selectors.html#working-with-relative-xpaths

- Sentiment analysis on Dataframe. (2017, October 16). Retrieved from https://stackoverflow.com/questions/46764674/sentiment-analysis-on-data-frame

- Terry-Jack, M. T. J. (2019, May 1). NLP: Pre-trained Sentiment Analysis. Retrieved from https://medium.com/@b.terryjack/nlp-pre-trained-sentiment-analysis-1eb52a9d742c

- TypeError: descriptor „strftime“ requires a „datetime.date“ object but received a „Text“. (2015, May 7). Retrieved from https://stackoverflow.com/questions/30112357/typeerror-descriptor-strftime-requires-a-datetime-date-object-but-received

- „utf-8“ codec can’t decode byte 0x92 in position 18: invalid start byte. (2017, September 1). Retrieved from https://stackoverflow.com/questions/46000191/utf-8-codec-cant-decode-byte-0x92-in-position-18-invalid-start-byte

- write header rows to csv python. (2017, December 1). Retrieved from https://stackoverflow.com/questions/47589327/write-header-rows-to-csv-python
- Modeling air pollution. (n.d.). Retrieved from https://colab.research.google.com/drive/1KZkfWlFDK2aX6fiGkjO_c5tZi_wMAzZA

Appendix

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0129	0,0129		0,0129		-0,0129
flair_header	0,0129	1	1		1		-1
flair_content	0,0129	1	1		1		-1
vader_header							
vader_content	0,0129	1	1		1		-1
textblob_header							
textblob_content	-0,0129	-1	-1		-1		1

Table 16: Correlation Audi On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0319	-0,0319		-0,0319		0,0319
flair_header	-0,0319	1	1		1		-1
flair_content	-0,0319	1	1		1		-1
vader_header							
vader_content	-0,0319	1	1		1		-1
textblob_header							
textblob_content	0,0319	-1	-1		-1		1

Table 17: Correlation Audi On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,00269	0,00517	0,00443	-0,001973	0,003481	0,00189
flair_header	0,00269	1	0,46696	0,183694635	-0,86277	0,72132	0,97488
flair_content	0,00517	0,46696	1	0,76694004	-0,41862	0,73724	0,2980
vader_header	0,00443	0,18369	0,7669	1	0,07198	0,197114	0,063324
vader_ content	-0,00197	-0,86277	-0,41862	0,07198	1	-0,89093	-0,78402
textblob-header	0,00348	0,7213	0,7372	0,19711	-0,89093	1	0,57532
textblob_content	0,00189	0,97488	0,2980	0,06332	-0,784025	0,57536	1

Table 18: Correlation BMW On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,00294	-0,00033	0,00302	0,004669	-0,0039	-0,0027
flair_header	-0,00294	1	0,46696	0,1837	-0,86277	0,72132	0,97488
flair_content	-0,00033	0,46696	1	0,7669	-0,4186	0,7372	0,298001
vader_header	0,00302	0,18369	0,7669	1	0,07198	0,197114	0,06332
vader_content	0,00467	-0,86277	-0,4186	0,07198	1	-0,89093	-0,78403
textblob_header	-0,0039	0,7213	0,7372	0,1971	-0,89093	1	0,57532
textblob_content	-0,0027	0,9749	0,298	0,0633	-0,784025	0,57532	1

Table 19: Correlation BMW On Minutely Stock Price Data Volume vs. Semantics

	return	flair_ header	flair_ content	vader_header	vader_ content	textblob_ header	textblob_ content
return	1	-0,000663	-0,000480	-0,000238	0,003076	0,002963	0,003384
flair_ header	-0,000663	1	0,992331	-0,044565	0,06530	-0,576110	-0,569052
flair_ content	-0,000480	0,992331	1	0,041802	0,098117	-0,501297	-0,585779
vader_header	-0,000238	-0,044565	0,041802	1	0,604215	0,665655	0,103809
vader_ content	0,003076	0,06530	0,098117	0,604215	1	0,678773	0,653789
textblob_header	0,002963	-0,576110	-0,501297	0,665655	0,678773	1	0,685275
textblob_content	0,003384	-0,569052	-0,585779	0,103809	0,653789	0,685275	1

Table 20: Correlation Daimler On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_ content	textblob_ header	textblob_ content
volume	1	0,000104	-0,000083	0,00021	-0,003119	-0,002673	-0,003042
flair_header	0,000104	1	0,992331	-0,044565	0,06530	-0,576110	-0,569052
flair_content	-0,000083	0,992331	1	0,041802	0,098117	-0,501297	-0,585779
vader_header	0,00021	-0,04457	0,041802	1	0,604215	0,665655	0,103809
vader_content	-0,00311	0,06530	0,098117	0,604215	1	0,678773	0,653789
textblob_header	-0,00267	-0,57611	-0,501297	0,665655	0,678773	1	0,685275
textblob_content	-0,00304	-0,56905	-0,585779	0,103809	0,653789	0,685275	1

Table 21: Correlation Daimler On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,00064	-0,00268	0,00229	0,00517	0,00229	0,004
flair_header	0,00064	1	0,5245	0,6477	-0,04183	0,64767	-0,29976
flair_content	-0,00268	0,5245	1	0,5886	-0,74329	0,58859	-0,94669
vader_header	0,00229	0,6477	0,5886	1	0,09785	1	-0,307331
vader_content	0,00517	-0,0418	-0,74329	0,0978	1	0,09785	0,91696
textblob_header	0,00229	0,6477	0,5886	1	0,09785	1	-0,30733
textblob_content	0,004	-0,2998	-0,9467	-0,3073	0,9169	-0,30733	1

Table 22: Correlation Ferrari On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,00248	0,0048	0,0028	-0,0037	0,0028	-0,0046
flair_header	0,00248	1	0,52451	0,64766	-0,04182	0,64766	-0,29976
flair_content	0,0048	0,52451	1	0,58858	-0,74328	0,58858	-0,94669
vader_header	0,0028	0,64766	0,58858	1	0,09784	1	-0,30733
vader_content	-0,0037	-0,04182	-0,74328	0,09784	1	0,09784	0,91696
textblob_header	0,0028	0,64766	0,58858	1	0,09784	1	-0,30733
textblob_content	-0,0046	-0,29976	-0,94669	-0,30733	0,91696	-0,30733	1

Table 23: Correlation Ferrari On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,0027	-0,0051	0,0027	-0,0021	0,00075	0,003
flair_header	-0,0027	1	0,699047085	0,3576	0,31226	0,05104	-0,36263
flair_content	-0,0051	0,699	1	-0,4148	-0,0188	-0,55302	-0,75741
vader_header	0,0027	0,3576	-0,41485	1	0,46641	0,83313	0,54523
vader_content	-0,0021	0,31226	-0,0188	0,4664	1	0,60228	0,28426
textblob_header	0,00075	0,05105	-0,553	0,8331	0,60228	1	0,69005
textblob_content	0,003	-0,3626	-0,7574	0,545	0,28426	0,69005	1

Table 24: Correlation Fiat Chrysler On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,00225	-0,00018	-0,00242	0,00223	-0,00035	0,00106
flair_header	-0,0023	1	0,6991	0,3576	0,31226	0,05105	-0,36263
flair_content	-0,0002	0,69905	1	-0,4149	-0,018844	-0,553	-0,7574
vader_header	-0,0024	0,35763	-0,41485	1	0,46641	0,83313	0,54523
vader_content	0,00223	0,31226	-0,01884	0,4664	1	0,60229	0,28426
textblob_header	-0,0004	0,05105	-0,553	0,8331	0,60229	1	0,69005
textblob_content	0,00106	-0,3626	-0,7574	0,54523	0,284256	0,69005	1

Table 25: Correlation Fiat Chrysler On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,00203	-0,0046	0,00074	0,00567	0,0029	0,0044
flair_header	-0,002	1	0,741	0,445	0,026	0,294	-0,249
flair_content	-0,0046	0,741	1	-0,158	-0,41	-0,413	-0,413
vader_header	0,00074	0,448	-0,158	1	0,492	0,828	0,132
vader_content	0,00567	0,026	-0,41	0,492	1	0,585	0,817
textblob_header	0,0029	0,294	-0,413	0,828	0,587	1	0,21
textblob_content	0,0044	-0,249	-0,413	0,132	0,817	0,21	1

Table 26: Correlation Peugeot On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0053	0,0033	0,0034	0,0029	0,0023	0,0007
flair_header	0,0053	1	0,7407	0,4448	0,0263	0,2938	-0,249
flair_content	0,0033	0,7407	1	-0,158	-0,41	-0,413	-0,4134
vader_header	0,0034	0,4448	-0,158	1	0,492	0,828	0,1315
vader_content	0,0029	0,0263	-0,41	0,49213	1	0,5865	0,8174
textblob_header	0,0023	0,2938	-0,4128	0,828	0,5865	1	0,210
textblob_content	0,0007	-0,2492	-0,4134	0,1315	0,8174	0,21	1

Table 27: Correlation Peugeot On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0041	-0,0024	0,001	0,0048	0,0027	0,0021
flair_header	0,0041	1	0,3824	0,0101	0,1539	0,3726	0,1302
flair_content	-0,0024	0,3824	1	-0,3708	-0,0846	-0,219	-0,3063
vader_header	0,001	0,0101	-0,371	1	0,3304	0,7274	-0,4255
vader_content	0,0048	0,1539	-0,0846	0,3304	1	-0,0796	-0,3491
textblob_header	0,0027	0,3726	-0,2198	0,7274	-0,0796	1	-0,1204
textblob_content	0,0021	0,1302	-0,3063	-0,425	-0,3491	-0,1204	1

Table 28: Correlation Porsche On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0003	-0,0055	0,0011	0,0006	0,0022	0,0041
flair_header	0,0003	1	0,3823	0,0101	0,1539	0,3726	0,1303
flair_content	-0,0055	0,3824	1	-0,3708	-0,0846	-0,2198	-0,3063
vader_header	0,0011	0,0101	-0,3708	1	0,3304	0,7274	-0,4255
vader_content	0,0006	0,1539	-0,0846	0,3304	1	-0,0796	-0,3491
textblob_header	0,0022	0,3726	-0,2198	0,7274	-0,0796	1	-0,1204
textblob_content	0,0041	0,1303	-0,3063	-0,4256	-0,3491	-0,1204	1

Table 29: Correlation Porsche On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,0013	-0,0008	-0,002	0,0004	-0,0002	0,004
flair_header	-0,0013	1	0,553	0,539	0,253	0,465	-0,09
flair_content	-0,0008	0,553	1	0,266	0,591	0,421	-0,205
vader_header	-0,002	0,539	0,266	1	0,616	0,508	-0,289
vader_content	0,0004	0,253	0,591	0,616	1	0,513	-0,046
textblob_header	-0,0002	0,465	0,421	0,508	0,513	1	-0,042
textblob_content	0,004	-0,09	-0,205	-0,289	-0,046	-0,042	1

Table 30: Correlation Renault On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0022	-0,0034	-0,0024	-0,0018	-0,0001	0,0029
flair_header	-0,0022	1	0,5527	0,5393	0,2532	0,4653	-0,0904
flair_content	-0,0034	0,5527	1	0,2656	0,5914	0,4209	-0,2053
vader_header	-0,0024	0,5394	0,2656	1	0,6156	0,5076	-0,2889
vader_content	-0,0018	0,2532	0,5914	0,6156	1	0,513	-0,0459
textblob_header	-0,0001	0,4654	0,421	0,5076	0,513	1	-0,0422
textblob_content	0,0029	-0,0904	-0,2053	-0,2889	-0,0459	-0,0422	1

Table 31: Correlation Renault On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0032	0,0041	-0,0016	-0,0012	0,0006	-0,0041
flair_header	0,0036	1	0,3298	0,2441	-0,0261	0,2378	-0,2091
flair_content	0,0041	0,3298	1	-0,1146	-0,2043	-0,3273	-0,3292
vader_header	-0,0016	0,2441	-0,1146	1	0,5685	0,3761	0,1126
vader_content	-0,0012	-0,0262	-0,2043	0,5685	1	0,2897	0,3174
textblob_header	0,0006	0,2378	-0,3274	0,3761	0,2897	1	0,2341
textblob_content	-0,0041	-0,2091	-0,3292	0,1126	0,3174	0,2341	1

Table 32: Correlation Volkswagen On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0026	-0,0014	0,0031	0,0023	0,0038	-0,0015
flair_header	-0,0026	1	0,3298	0,2441	-0,0261	0,2378	-0,2091
flair_content	-0,0014	0,3298	1	-0,1146	-0,2043	-0,3274	-0,3292
vader_header	0,0031	0,2441	-0,1146	1	0,5685	0,3761	0,1126
vader_content	0,0023	-0,0261	-0,2043	0,5685	1	0,2897	0,3174
textblob_header	0,0038	0,2378	-0,3274	0,3761	0,2897	1	0,2341
textblob_content	-0,0015	-0,2091	-0,3292	0,1126	0,3174	0,2341	1

Table 33: Correlation Volkswagen On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,0014	-0,0015	0,0022	-0,0022	-0,0034	-0,0026
flair_header	-0,0014	1	0,9999	0,4103	0,5541	0,4009	0,3939
flair_content	-0,0015	0,9999	1	0,4083	0,5541	0,4082	0,3951
vader_header	0,0022	0,4103	0,4083	1	0,4763	-0,00001	0,3273
vader_content	-0,0022	0,5541	0,5541	0,4763	1	0,4279	0,315
textblob_header	-0,0034	0,4009	0,4082	-0,00001	0,4279	1	0,3042
textblob_content	-0,0026	0,3939	0,3951	0,3273	0,315	0,3042	1

Table 34: Correlation Volvo On Minutely Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0021	0,0022	-0,0028	0,0011	0,0027	-0,0036
flair_header	0,0021	1	0,9999	0,4103	0,5541	0,4009	0,3939
flair_content	0,0022	0,9999	1	0,4083	0,5541	0,4082	0,3951
vader_header	-0,0028	0,4103	0,4083	1	0,4763	-0,00001	0,3273
vader_content	0,0011	0,5541	0,5541	0,4763	1	0,4279	0,3151
textblob_header	0,0027	0,4009	0,4082	-0,00001	0,4279	1	0,3042
textblob_content	-0,0036	0,3939	0,3951	0,3273	0,3151	0,3042	1

Table 35: Correlation Volvo On Minutely Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0299	0,0741	-0,0327	-0,0123	-0,019	0,0069
flair_header	0,0299	1	0,3673	0,182	0,265	0,3016	0,266
flair_content	0,0741	0,3673	1	-0,443	-0,2325	-0,304	0,0067
vader_header	-0,0327	0,182	-0,4428	1	0,5853	0,714	-0,227
vader_content	-0,0123	0,2647	-0,2325	0,585	1	0,3742	-0,227
textblob_header	-0,019	0,3016	-0,3041	0,714	0,3742	1	0,1294
textblob_content	0,0069	0,266	0,0067	-0,226	-0,227	0,1294	1

Table 36: Correlation Audi On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0246	-0,0267	-0,041	-0,0046	-0,019	0,044
flair_header	-0,0246	1	0,3673	0,182	0,2647	0,3016	0,266
flair_content	-0,0267	0,3673	1	-0,443	-0,2325	-0,304	0,0067
vader_header	-0,041	0,182	-0,443	1	0,585	0,714	-0,227
vader_content	-0,0046	0,2647	-0,2325	0,585	1	0,374	-0,227
textblob_header	-0,0195	0,3016	-0,304	0,714	0,3742	1	0,1294
textblob_content	0,0436	0,266	0,0067	-0,227	-0,227	0,1294	1

Table 37: Correlation Audi On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,0036	-0,007	-0,025	0,0085	-0,025	-0,0089
flair_header	-0,0036	1	0,306	0,017	0,0149	-0,1703	0,0252
flair_content	-0,007	0,306	1	0,205	0,183	0,0512	0,026
vader_header	-0,025	0,0172	0,205	1	-0,1689	0,509	0,157
vader_content	0,0085	0,0149	0,183	-0,169	1	-0,25	-0,023
textblob_header	-0,025	-0,1702	0,0512	0,509	-0,253	1	0,517
textblob_content	-0,0089	0,0252	0,026	0,157	-0,023	0,517	1

Table 38: Correlation BMW On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0175	0,0183	0,017	-0,0068	-0,0175	0,003
flair_header	0,017	1	0,306	0,017	0,0149	-0,1703	0,0252
flair_content	0,0183	0,306	1	0,205	0,183	0,0512	0,026
vader_header	0,017	0,0172	0,205	1	-0,1689	0,509	0,157
vader_content	-0,0068	0,0149	0,183	-0,169	1	-0,2529	-0,0235
textblob_header	-0,0175	-0,1703	0,0512	0,509	-0,2529	1	0,5165
textblob_content	0,003	0,0252	0,026	0,157	-0,023	0,517	1

Table 39: Correlation BMW On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,0233	0,0163	0,025	0,0114	-0,0346	-0,0056
flair_header	-0,0233	1	0,4705	-0,102	0,0399	0,228	0,0279
flair_content	0,0163	0,4705	1	-0,175	0,0229	-0,111	-0,06
vader_header	0,025	-0,1016	-0,175	1	0,1618	0,262	0,2033
vader_content	0,011	0,0399	0,023	0,162	1	0,3724	0,4958
textblob_header	-0,0346	0,228	-0,111	0,262	0,3724	1	0,523
textblob_content	-0,0056	0,0279	-0,06	0,203	0,4958	0,523	1

Table 40: Correlation Daimler On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,019	0,034	-0,011	0,0079	0,0255	-0,009
flair_header	0,0197	1	0,4705	-0,102	0,0399	0,228	0,0279
flair_content	0,0348	0,4705	1	-0,175	0,0229	-0,1109	-0,06
vader_header	-0,0108	-0,1016	-0,1748	1	0,1618	0,262	0,2033
vader_content	0,008	0,0399	0,023	0,162	1	0,3724	0,4958
textblob_header	0,0255	0,228	-0,111	0,263	0,3724	1	0,523
textblob_content	-0,009	0,028	-0,06	0,203	0,4958	0,523	1

Table 41: Correlation Daimler On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,039	-0,0647	-0,005	0,035	-0,0305	0,0391
flair_header	0,0392	1	0,0525	0,445	0,2198	0,5099	-0,089
flair_content	-0,0647	0,0525	1	0,357	-0,524	0,4526	-0,5633
vader_header	-0,0047	0,445	0,357	1	0,1045	0,4213	-0,1201
vader_content	0,0348	0,2198	-0,524	0,105	1	-0,1708	0,3925
textblob_header	-0,0305	0,5098	0,453	0,421	-0,1708	1	-0,1092
textblob_content	0,0391	-0,0898	-0,563	-0,12	0,392	-0,109	1

Table 42: Correlation Ferrari On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0079	0,0492	0,0456	-0,0382	0,0433	0,0059
flair_header	0,0079	1	0,053	0,445	0,2198	0,5098	-0,0898
flair_content	0,0492	0,0525	1	0,3568	-0,524	0,4526	-0,5633
vader_header	0,0456	0,445	0,357	1	0,1045	0,4213	-0,1201
vader_content	-0,0382	0,2198	-0,524	0,105	1	-0,171	0,3925
textblob_header	0,0433	0,5098	0,4526	0,4213	-0,1708	1	-0,1092
textblob_content	0,0059	-0,0899	-0,5633	-0,1201	0,3925	-0,109	1

Table 43: Correlation Ferrari On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0114	0,0079	0,0208	0,0057	-0,0153	-0,0108
flair_header	0,0114	1	0,8629	-0,083	0,2563	-0,2835	-0,5551
flair_content	0,0079	0,8629	1	-0,476	0,0812	-0,4175	-0,4093
vader_header	0,0208	-0,0832	-0,4764	1	0,4971	0,216	-0,0057
vader_content	0,0057	0,256	0,081	0,497	1	-0,068	0,0235
textblob_header	-0,0153	-0,2835	-0,4175	0,216	-0,068	1	0,471
textblob_content	-0,011	-0,5551	-0,4093	-0,0057	0,0235	0,471	1

Table 44: Correlation Fiat Chrysler On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
Volume	1	-0,0356	-0,0369	0,0155	-0,025	0,0334	0,0545
flair_header	-0,0356	1	0,8629	-0,083	0,2563	-0,2835	-0,5551
flair_content	-0,037	0,8629	1	-0,4764	0,0812	-0,4175	-0,4093
vader_header	0,0155	-0,0832	-0,4764	1	0,49713	0,216	-0,0057
vader_content	-0,025	0,2563	0,081	0,4971	1	-0,0675	0,0235
textblob_header	0,0334	-0,2835	-0,4175	0,216	-0,0675	1	0,471
textblob_content	0,0545	-0,5551	-0,4093	-0,0057	0,0235	0,471	1

Table 45: Correlation Fiat Chrysler On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,03229	0,0312	0,0392	0,0298	0,0521	0,0133
flair_header	0,0323	1	0,5987	0,1522	-0,3328	0,2644	-0,4185
flair_content	0,0312	0,5987	1	-0,1028	-0,487	-0,1857	-0,3112
vader_header	0,0392	0,1522	-0,1028	1	0,3522	0,5432	0,0326
vader_content	0,0298	-0,3328	-0,487	0,352	1	0,324	0,6475
textblob_header	0,05209	0,2644	-0,1857	0,5433	0,324	1	0,1972
textblob_content	0,0133	-0,4185	-0,3112	0,033	0,6475	0,1972	1

Table 46: Correlation Peugeot On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0776	0,1239	-0,0544	-0,0962	-0,0423	-0,0656
flair_header	0,07758	1	0,5987	0,1522	-0,3328	0,2644	-0,4185
flair_content	0,1239	0,599	1	-0,1028	-0,4873	-0,1857	-0,3112
vader_header	-0,0544	0,1522	-0,1028	1	0,352	0,5433	0,0326
vader_content	-0,0962	-0,3328	-0,4873	0,352	1	0,324	0,64759
textblob_header	-0,042	0,2644	-0,1857	0,5433	0,324	1	0,1972
textblob_content	-0,0656	-0,4185	-0,3112	0,033	0,6475	0,1972	1

Table 47: Correlation Peugeot On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0061	-0,0271	-0,0363	-0,019	-0,0227	0,0543
flair_header	0,0061	1	0,3768	0,0092	0,356	0,3253	0,0517
flair_content	-0,0271	0,3768	1	-0,3189	-0,0135	-0,1889	-0,3862
vader_header	-0,0363	0,0092	-0,3189	1	0,2857	0,7272	-0,3995
vader_content	-0,019	0,356	-0,0135	0,2857	1	-0,0687	-0,2997
textblob_header	-0,0227	0,3253	-0,1889	0,727	-0,0687	1	-0,11311
textblob_content	0,0543	0,0517	-0,3862	-0,3995	-0,2997	-0,1131	1

Table 48: Correlation Porsche On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0199	0,0397	-0,0242	0,0129	-0,0248	0,0112
flair_header	0,0199	1	0,3768	0,0092	0,356	0,3253	0,0517
flair_content	0,0397	0,3768	1	-0,3189	-0,0135	-0,1889	-0,3862
vader_header	-0,0241	0,0092	-0,3189	1	0,2857	0,7272	-0,3995
vader_content	0,0129	0,356	-0,0135	0,2857	1	-0,0687	-0,2997
textblob_header	-0,0249	0,3253	-0,1889	0,7272	-0,0687	1	-0,1131
textblob_content	0,0112	0,0517	-0,3862	-0,3995	-0,2997	-0,1131	1

Table 49: Correlation Porsche On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0073	0,0283	0,0413	0,0335	0,0296	0,0482
flair_header	0,0074	1	0,6921	0,411	0,2878	0,1868	-0,0129
flair_content	0,0283	0,692	1	0,3279	0,4999	0,2081	-0,0129
vader_header	0,0413	0,411	0,3279	1	0,5025	0,347	-0,1362
vader_content	0,0335	0,2878	0,4999	0,5025	1	0,2959	0,1645
textblob_header	0,0296	0,1868	0,2081	0,347	0,2959	1	0,0526
textblob_content	0,0482	-0,0129	-0,0129	-0,136	0,1645	0,0526	1

Table 50: Correlation Renault On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0179	-0,0109	-0,0725	-0,057	-0,0558	-0,0114
flair_header	-0,0179	1	0,6921	0,411	0,2878	0,1868	-0,0129
flair_content	-0,0109	0,6921	1	0,328	0,4999	0,2081	-0,0129
vader_header	-0,0725	0,4109	0,3279	1	0,5025	0,347	-0,1362
vader_content	-0,0574	0,2878	0,4999	0,5025	1	0,2959	0,1645
textblob_header	-0,0558	0,1868	0,2081	0,347	0,2959	1	0,0526
textblob_content	-0,0114	-0,0129	-0,0129	-0,136	0,1645	0,0526	1

Table 51: Correlation Renault On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,0015	-0,0334	-0,017	-0,0256	0,0173	0,022
flair_header	-0,0015	1	0,2844	0,2778	0,0678	0,1304	-0,1339
flair_content	-0,0334	0,2844	1	0,0699	0,1445	-0,1009	-0,2458
vader_header	-0,0169	0,2778	0,0699	1	0,5103	0,0227	0,0256
vader_content	-0,0256	0,0678	0,1445	0,5103	1	0,0507	0,1085
textblob_header	0,0173	0,1305	-0,1009	0,0227	0,0507	1	0,1181
textblob_content	0,022	-0,1339	-0,2458	0,0266	0,1085	0,1181	1

Table 52: Correlation Volkswagen On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0265	0,0165	0,0273	0,0256	0,0105	0,0081
flair_ header	-0,0265	1	0,2844	0,2778	0,0678	0,1305	-0,1339
flair_content	0,0165	0,2844	1	0,0699	0,1445	-0,1009	-0,2458
vader_header	0,0273	0,2778	0,0699	1	0,5103	0,0227	0,0256
vader_content	0,0256	0,0678	0,1445	0,5103	1	0,0507	0,1085
textblob_header	0,0105	0,1305	-0,1009	0,0227	0,051	1	0,1181
textblob_content	0,0081	-0,1339	-0,2458	0,026	0,1085	0,1181	1

Table 53: Correlation Volkswagen On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0091	0,0113	-0,0138	-0,0175	-0,0333	-0,0009
flair_ header	0,0091	1	0,9901	0,4461	0,6796	0,4326	0,591
flair_content	0,0113	0,9901	1	0,5199	0,6846	0,4256	0,6019
vader_header	-0,0138	0,4461	0,5199	1	0,5292	-0,0011	0,3901
vader_content	-0,0175	0,6796	0,6846	0,529	1	0,4748	0,3981
textblob_header	-0,0333	0,4326	0,4255	-0,0011	0,4748	1	0,3086
textblob_content	-0,0009	0,5909	0,6019	0,39	0,3981	0,3086	1

Table 54: Correlation Volvo On Hourly Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,006	0,0024	-0,0385	-0,0228	-0,0315	-0,0074
flair_header	0,006	1	0,9901	0,4461	0,6796	0,4326	0,5909
flair_content	0,0024	0,9901	1	0,5199	0,6846	0,4255	0,6019
vader_header	-0,0385	0,4461	0,5199	1	0,5292	-0,0018	0,3901
vader_content	-0,0228	0,6796	0,6846	0,5292	1	0,4748	0,3981
textblob_header	-0,0315	0,4326	0,4255	-0,0011	0,4748	1	0,3086
textblob_content	-0,0074	0,5909	0,6019	0,3901	0,3981	0,3086	1

Table 55: Correlation Volvo On Hourly Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vade_content	textblob_header	textblob_content
return	1	0,0999	0,0493	0,0544	0,0224	0,0899	0,0193
flair_header	0,0999	1	0,6137	0,3947	0,0072	0,15	0,1394
flair_content	0,0493	0,6137	1	-0,0226	-0,395	-0,0978	-0,1335
vader_header	0,0544	0,3947	-0,0225	1	0,5072	0,4923	0,0266
vader_content	0,0224	0,0072	-0,395	0,5072	1	0,3056	0,2628
textblob_header	0,0899	0,15	-0,0978	0,4923	0,3056	1	0,1321
textblob_content	0,0193	0,1394	-0,1335	0,0266	0,2628	0,1321	1

Table 56: Correlation Audi On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,09339	-0,01	-0,0621	-0,0118	0,0063	-0,0062
flair_header	-0,0939	1	0,6137	0,3947	0,0072	0,15	0,1394
flair_content	-0,0102	0,6137	1	-0,0226	-0,395	-0,0978	-0,1335
vader_header	-0,0621	0,3947	-0,0226	1	0,5072	0,4923	0,0266
vader_content	-0,0118	0,0072	-0,395	0,5072	1	0,3056	0,2628
textblob_header	0,0063	0,15	-0,0978	0,4923	0,3056	1	0,1321
textblob_content	-0,0062	0,1394	-0,1335	0,0266	0,2628	0,1321	1

Table 57: Correlation Audi On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0399	0,0536	-0,0032	0,0464	0,042	-0,089
flair_ header	0,0399	1	0,585	0,1899	-0,1477	0,1643	-0,0506
flair_content	0,0536	0,585	1	0,3228	-0,1048	0,24	-0,0627
vader_header	-0,0032	0,1899	0,3228	1	-0,1279	0,1794	0,0981
vader_content	0,0464	-0,1477	-0,1048	-0,1279	1	-0,0927	0,1482
textblob_header	0,0421	0,1643	0,24	0,1794	-0,0927	1	0,3759
textblob_content	-0,089	-0,0506	-0,0627	0,0981	0,1482	0,3759	1

Table 58: Correlation BMW On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0406	-0,0287	-0,0529	0,0879	-0,0266	-0,0241
flair_header	-0,0406	1	0,585	0,1899	-0,1477	0,1643	-0,0506
flair_content	-0,0287	0,585	1	0,3228	-0,1048	0,24	-0,0627
vader_header	-0,0529	0,1899	0,3228	1	-0,1279	0,1794	0,0981
vader_content	0,0879	-0,1477	-0,1048	-0,1279	1	-0,0927	0,1482
textblob_header	-0,0266	0,1643	0,24	0,1794	-0,0927	1	0,3759
textblob_content	-0,0241	-0,0506	-0,0627	0,0981	0,1482	0,3759	1

Table 59: Correlation BMW On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0144	-0,0012	-0,0473	0,056	-0,0185	-0,0886
flair_header	0,0144	1	0,5894	0,0591	0,2361	0,1643	-0,0693
flair_content	-0,0012	0,5894	1	-0,1789	0,0382	-0,0715	-0,1705
compound_header	-0,0473	0,0591	-0,1789	1	0,2669	0,5532	0,5369
vader_content	0,056	0,2361	0,0382	0,2669	1	0,2119	0,2524
textblob_header	-0,0185	0,1643	-0,0715	0,5532	0,2119	1	0,5334
textblob_content	-0,0886	-0,0693	-0,1705	0,5369	0,2525	0,5334	1

Table 60: Correlation Daimler On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,1219	0,0788	0,0249	0,1081	0,0357	0,0204
flair_header	0,1219	1	0,5894	0,0591	0,2361	0,1643	-0,0693
flair_content	0,0788	0,5894	1	-0,1789	0,0382	-0,0715	-0,1705
vader_header	0,0249	0,0591	-0,1789	1	0,2669	0,5532	0,5369
vader_content	0,1081	0,2361	0,0382	0,2669	1	0,2119	0,2525
textblob_header	0,0357	0,1643	-0,0715	0,5532	0,2119	1	0,5334
textblob_content	0,0204	-0,0693	-0,1705	0,5369	0,2524	0,5334	1

Table 61: Correlation Daimler On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0687	0,0031	-0,0189	0,0071	-0,0139	0,0534
flair_header	0,0687	1	0,4997	-0,4792	-0,4338	-0,2329	-0,3668
flair_content	0,0032	0,4997	1	-0,3112	-0,9049	0,0433	-0,6638
vader_header	-0,0189	-0,4792	-0,3112	1	0,5258	0,74	0,2585
vader_content	0,0071	-0,4338	-0,9049	0,5258	1	0,2533	0,6883
textblob_header	-0,0139	-0,2329	0,0433	0,74	0,2533	1	0,2048
textblob_content	0,0534	-0,3668	-0,6638	0,2585	0,6883	0,2048	1

Table 62: Correlation Ferrari On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0381	-0,082	0,044	0,0858	-0,0079	0,0288
flair_header	-0,0381	1	0,49971	-0,4792	-0,4338	-0,2329	-0,3668
flair_content	-0,082	0,4997	1	-0,3112	-0,9049	0,0433	-0,6638
vader_header	0,0441	-0,4792	-0,3112	1	0,5258	0,74	0,2585
vader_content	0,0858	-0,4338	-0,9049	0,5258	1	0,2533	0,6883
textblob_header	-0,0079	-0,2329	0,0433	0,74	0,2533	1	0,2048
textblob_content	0,0288	-0,3668	-0,6638	0,2585	0,6883	0,2048	1

Table 63: Correlation Ferrari On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,1365	0,1057	0,0586	-0,0346	-0,0638	-0,0598
flair_header	0,1365	1	0,7951	0,2782	-0,0879	-0,1347	-0,2739
flair_content	0,1057	0,7951	1	-0,031	-0,2733	-0,3534	-0,2615
vader_header	0,0586	0,2782	-0,0309	1	0,2435	0,0447	-0,0105
vader_content	-0,0346	-0,0879	-0,2733	0,2435	1	0,1393	0,2903
textblob_header	-0,0638	-0,1347	-0,3534	0,0447	0,1393	1	0,2738
textblob_content	-0,0598	-0,2739	-0,2615	-0,01	0,2903	0,2738	1

Table 64: Correlation Fiat Chrysler On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0412	0,0188	0,0021	0,0241	0,0693	-0,0675
flair header	0,0412	1	0,7951	0,2782	-0,0879	-0,1347	-0,2739
flair_content	0,0188	0,7951	1	-0,0309	-0,2733	-0,3534	-0,2615
vader_header	0,0021	0,2782	-0,0309	1	0,2435	0,0447	-0,0105
vader content	0,0241	-0,0879	-0,2733	0,2435	1	0,1393	0,2903
textblob header	0,0693	-0,1347	-0,3534	0,0447	0,1393	1	0,2738
textblob_content	-0,0675	-0,2739	-0,2615	-0,0105	0,2903	0,2738	1

Table 65: Correlation Fiat Chrysler On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0634	0,0854	-0,0335	-0,0416	0,0832	-0,0149
flair_header	0,0634	1	0,6044	0,2015	-0,1058	0,1456	-0,2878
flair_content	0,0854	0,6044	1	-0,2202	-0,5137	-0,1529	-0,1869
vader_header	-0,0335	0,2015	-0,2202	1	0,1023	0,28	-0,1529
vader_content	-0,0416	-0,1058	-0,5137	0,1023	1	0,3869	0,3493
textblob header	0,0832	0,1456	-0,1529	0,28	0,3869	1	0,2425
textblob_content	-0,0149	-0,2878	-0,1869	-0,1529	0,3493	0,2425	1

Table 66: Correlation Peugeot On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0633	0,0082	0,0872	-0,0633	-0,0395	-0,0686
flair_header	0,0633	1	0,6044	0,2015	-0,1058	0,1456	-0,2878
flair_content	0,0082	0,6044	1	-0,2202	-0,5137	-0,1529	-0,1869
vader_header	0,0872	0,2015	-0,2202	1	0,1023	0,28	-0,1529
vader_content	-0,0633	-0,1057	-0,5137	0,1023	1	0,3869	0,3493
textblob_header	-0,039	0,1456	-0,1529	0,28	0,3869	1	0,2425
textblob_content	-0,0686	-0,2878	-0,1869	-0,1529	0,3493	0,2425	1

Table 67: Correlation Peugeot On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,0134	-0,0209	0,0318	0,0521	-0,0781	0,0716
flair_header	-0,0134	1	0,3738	0,1677	-0,0202	0,3977	-0,0165
flair_content	-0,0209	0,3738	1	-0,207	-0,1216	-0,1358	-0,2278
vader_header	0,0318	0,1677	-0,207	1	0,3835	0,4516	-0,0602
vader_content	0,0521	-0,0202	-0,1216	0,3835	1	-0,1086	0,1679
textblob_header	-0,0781	0,3977	-0,1358	0,4516	-0,1086	1	-0,0938
textblob_content	0,0716	-0,0165	-0,2278	-0,0602	0,1679	-0,0938	1

Table 68: Correlation Porsche On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,1123	-0,0355	-0,0374	-0,0182	-0,0705	0,1181
flair_header	-0,1123	1	0,3738	0,1677	-0,0202	0,3977	-0,0165
flair_content	-0,0355	0,3738	1	-0,207	-0,1216	-0,1358	-0,2278
vader_header	-0,0374	0,1677	-0,207	1	0,3835	0,4516	-0,0602
vader content	-0,0182	-0,0202	-0,1216	0,3835	1	-0,1086	0,1679
polarity_textblob_header	-0,0705	0,3977	-0,1358	0,4516	-0,1086	1	-0,09375
textblob_content	0,1181	-0,0165	-0,2278	-0,0602	0,1679	-0,09375	1

Table 69: Correlation Porsche On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	-0,0519	0,0071	0,0477	0,07	0,027	-0,0028
flair_header	-0,0519	1	0,6596	0,2979	0,3255	0,1225	-0,1025
flair_content	0,0071	0,6596	1	0,2404	0,6969	0,2879	-0,1964
vader_header	0,047	0,2979	0,2404	1	0,4905	0,4344	0,0492
vader_content	0,07	0,3255	0,6969	0,4905	1	0,4506	0,1082
textblob_header	0,027	0,1225	0,2879	0,4344	0,4506	1	0,0358
textblob_content	-0,0028	-0,1025	-0,1963	0,0492	0,1082	0,0358	1

Table 70: Correlation Renault On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
Volume	1	-0,0481	-0,0161	0,0205	0,0946	0,1017	0,1359
flair_header	-0,0481	1	0,6596	0,2979	0,3255	0,1225	-0,1025
flair_content	-0,0161	0,6596	1	0,2404	0,6969	0,2879	-0,1963
vader_header	0,0205	0,2979	0,2404	1	0,4905	0,4344	0,0492
vader_content	0,0946	0,3255	0,6969	0,4905	1	0,4506	0,1082
textblob_header	0,1017	0,1225	0,2879	0,4344	0,4505	1	0,0358
textblob_content	0,1359	-0,1025	-0,1963	0,0492	0,1082	0,0358	1

Table 71: Correlation Renault On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
Return	1	0,0307	-0,0558	-0,0394	0,0169	-0,0182	0,0086
flair_header	0,0307	1	0,701	0,088	-0,0472	0,0562	-0,3957
flair_content	-0,0558	0,701	1	0,0077	-0,0663	-0,0035	-0,4454
vader_header	-0,0394	0,088	0,0077	1	0,3112	0,0084	0,0224
vader_content	0,0169	-0,0472	-0,0663	0,3112	1	0,0543	0,3433
textblob_header	-0,0182	0,0562	-0,0035	0,0084	0,0543	1	0,1614
textblob_content	0,0086	-0,3957	-0,4454	0,0224	0,3433	0,1614	1

Table 72: Correlation Volkswagen On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	0,0193	0,025	0,0072	0,1272	0,0951	-0,0023
flair_header	0,0193	1	0,701	0,088	-0,0472	0,0562	-0,3957
flair_content	0,025	0,701	1	0,0077	-0,0663	-0,0035	-0,4454
vader_header	0,0072	0,088	0,0077	1	0,3112	0,0084	0,0224
vader_content	0,1272	-0,0472	-0,0663	0,3112	1	0,0543	0,3433
textblob header	0,0951	0,0562	-0,0035	0,0084	0,0543	1	0,1614
textblob_content	-0,0023	-0,3957	-0,4454	0,0224	0,3433	0,1614	1

Table 73: Correlation Volkswagen On Daily Stock Price Data Volume vs. Semantics

	return	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
return	1	0,0548	0,044	0,0491	0,0556	-0,0619	0,0696
flair_header	0,0548	1	0,7717	0,3578	0,1876	0,3284	-0,0239
flair content	0,044	0,7717	1	0,1131	0,1326	0,3186	-0,0372
vader_header	0,0491	0,3578	0,1131	1	0,5596	0,48	0,2366
vader content	0,0556	0,1876	0,1324	0,5596	1	0,2856	0,3857
textblob_header	-0,0619	0,3284	0,3186	0,48	0,2856	1	-0,16287
textblob_content	0,0696	-0,0239	-0,0372	0,2366	0,3857	-0,1628	1

Table 74: Correlation Volvo On Daily Stock Price Data Return vs. Semantics

	volume	flair_header	flair_content	vader_header	vader_content	textblob_header	textblob_content
volume	1	-0,0663	-0,0732	0,0126	-0,0034	-0,0657	0,109
flair_header	-0,0663	1	0,7717	0,3578	0,1876	0,3284	-0,0239
flair_content	-0,0732	0,7717	1	0,1131	0,1324	0,3186	-0,0372
vader_header	0,0126	0,3578	0,1131	1	0,5596	0,48	0,2366
vader_content	-0,0034	0,1876	0,1324	0,5596	1	0,2856	0,3857
textblob_header	-0,0657	0,3284	0,3186	0,48	0,2856	1	-0,1628
textblob_content	0,109	-0,0239	-0,0372	0,2366	0,3857	-0,1628	1

Table 75: Correlation Volvo On Daily Stock Price Data Volume vs. Semantics

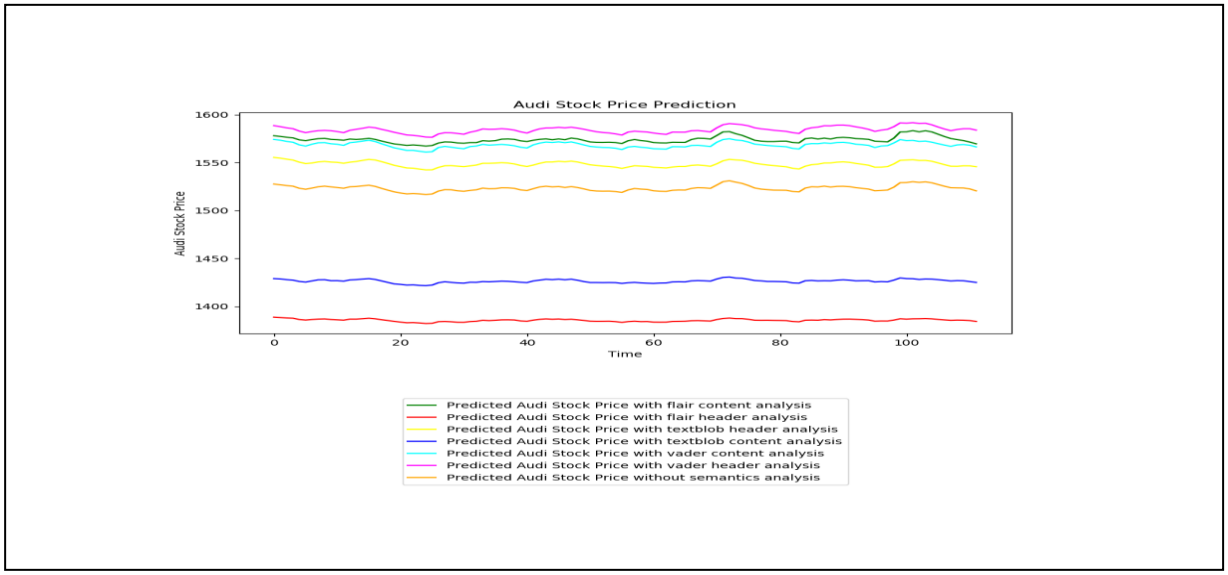


Figure 33: Stock Price Prediction Of Audi With Minutely Stock Price Data Using LSTM

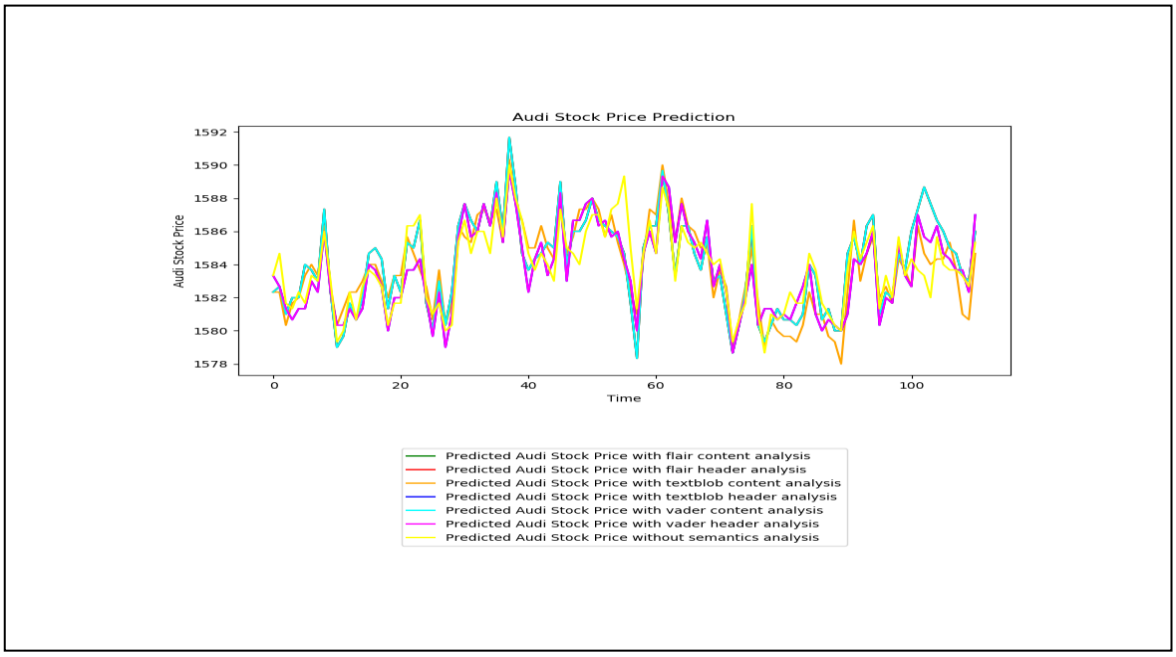


Figure 34: Stock Price Prediction Of Audi With Minutely Stock Price Data Using RandomForest Based Model

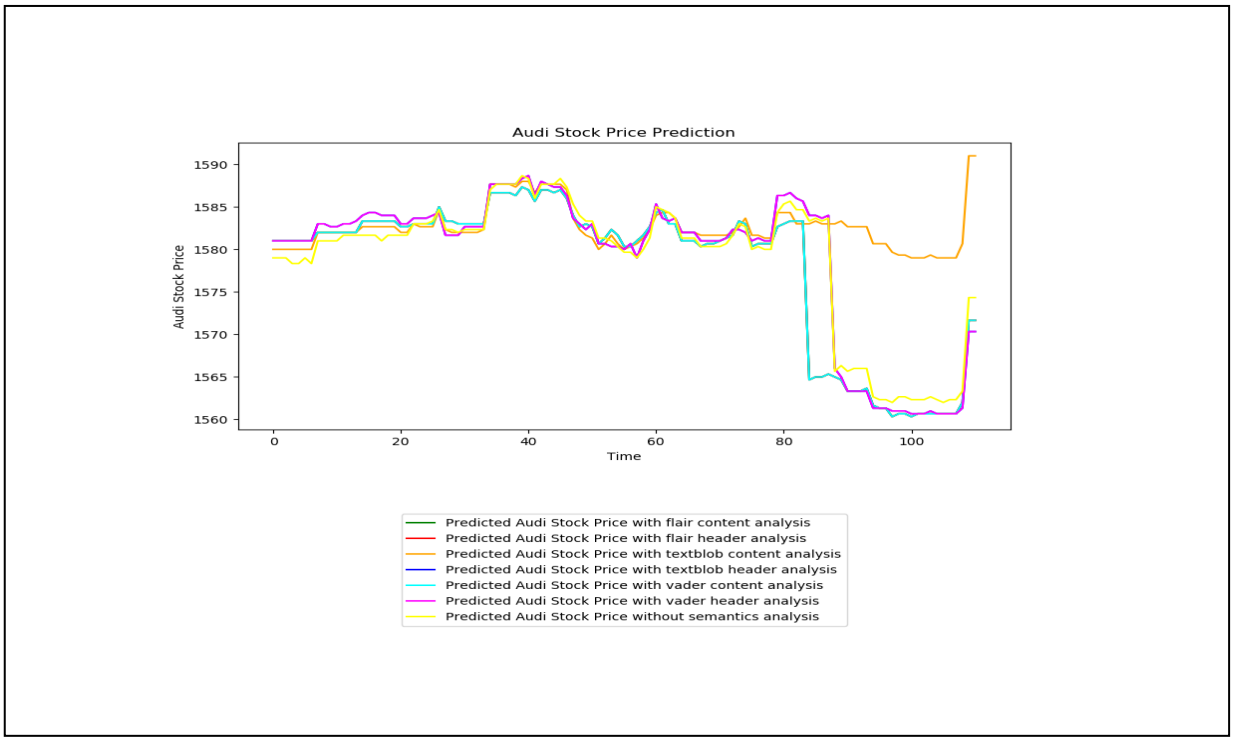


Figure 35: Stock Price Prediction Of Audi With Minutely Stock Price Data Using RandomForest Feature Model

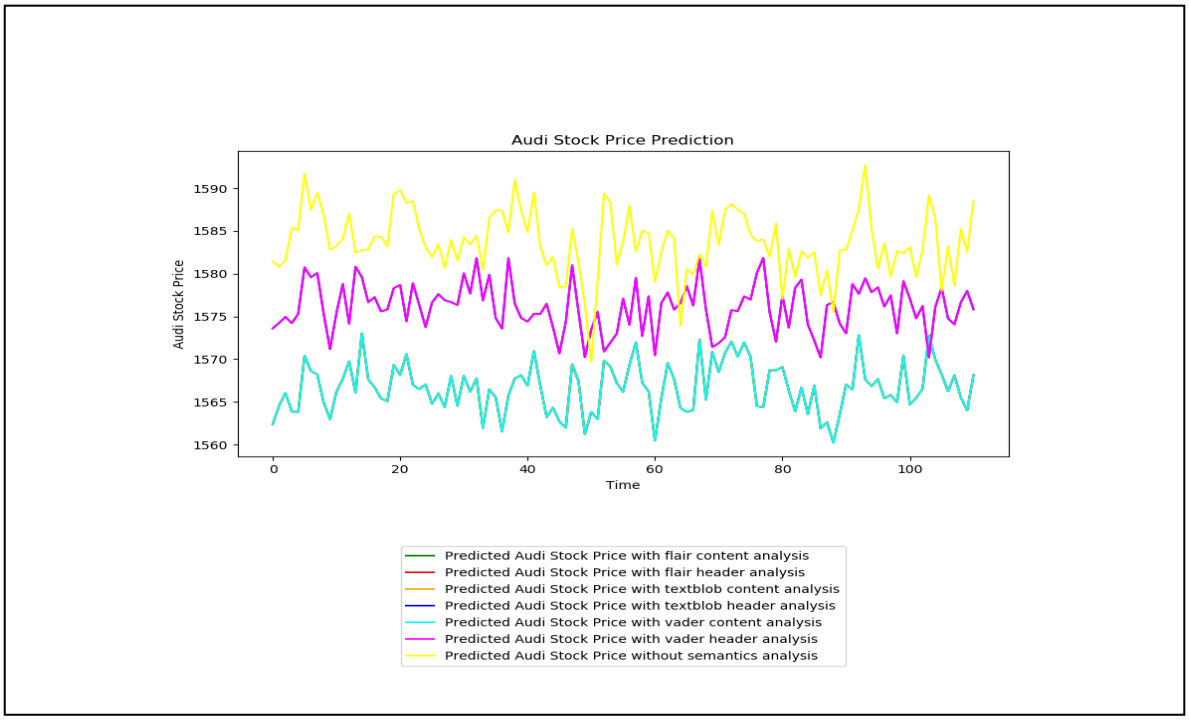


Figure 36: Stock Price Prediction Of Audi With Minutely Stock Price Data Using XGBoost

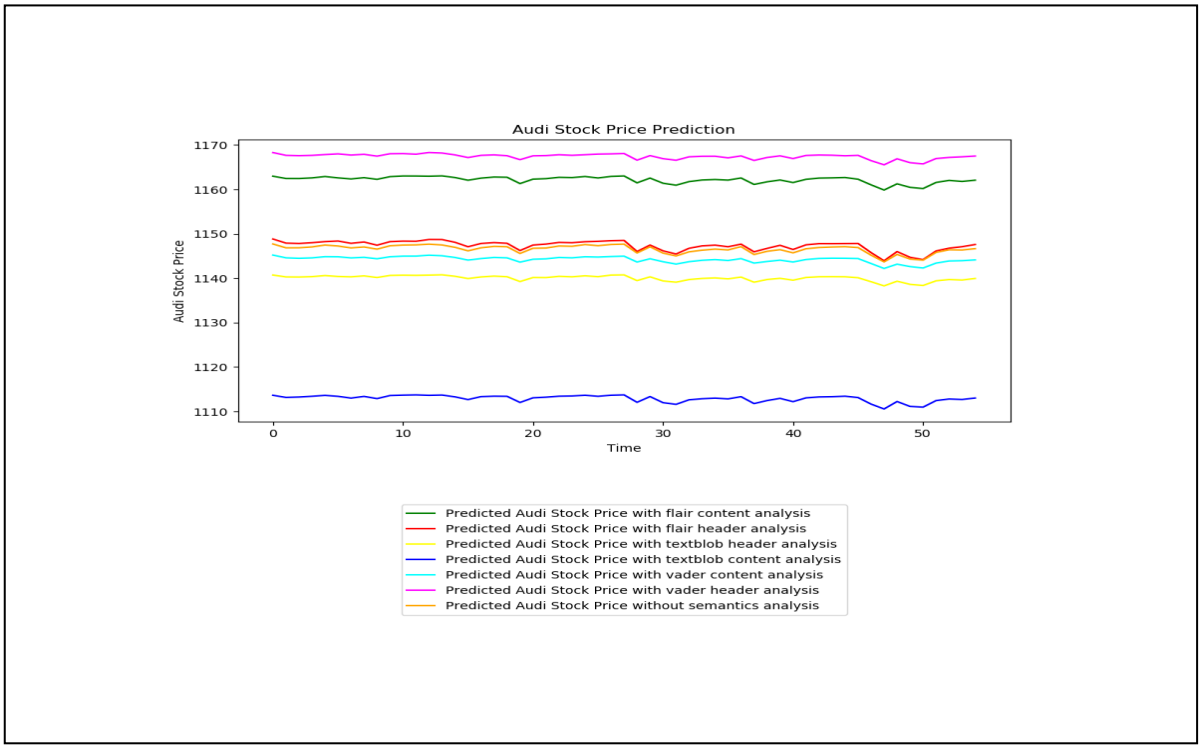


Figure 37: Stock Price Prediction Of Audi With Hourly Stock Price Data Using LSTM

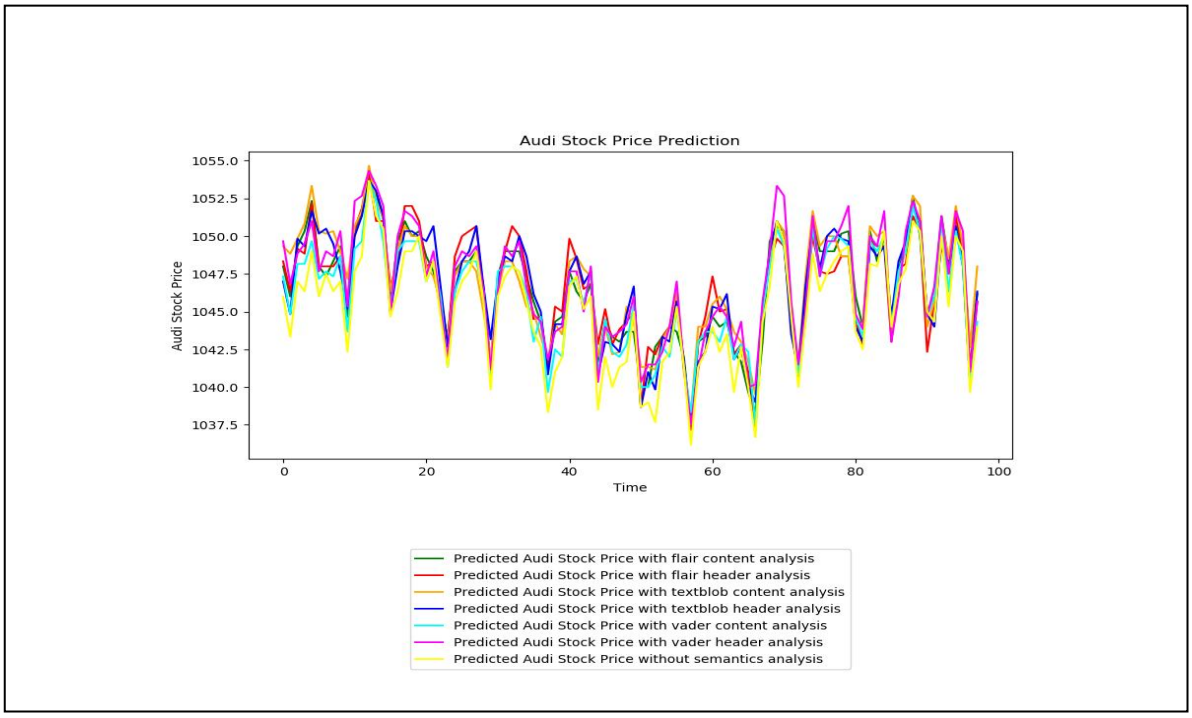


Figure 38: Stock Price Prediction Of Audi With Hourly Stock Price Data Using RandomForest Base Model

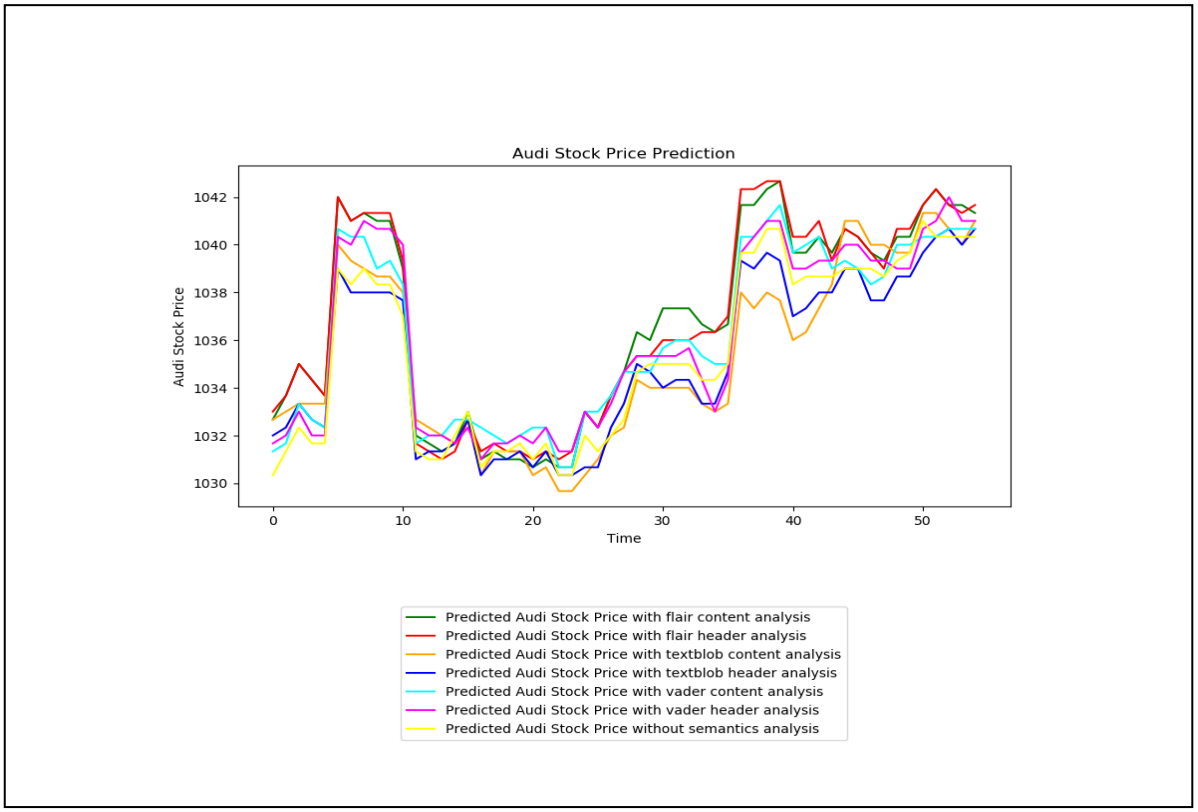


Figure 39: Stock Price Prediction Of Audi With Hourly Stock Price Data Using RandomForest Feature Model

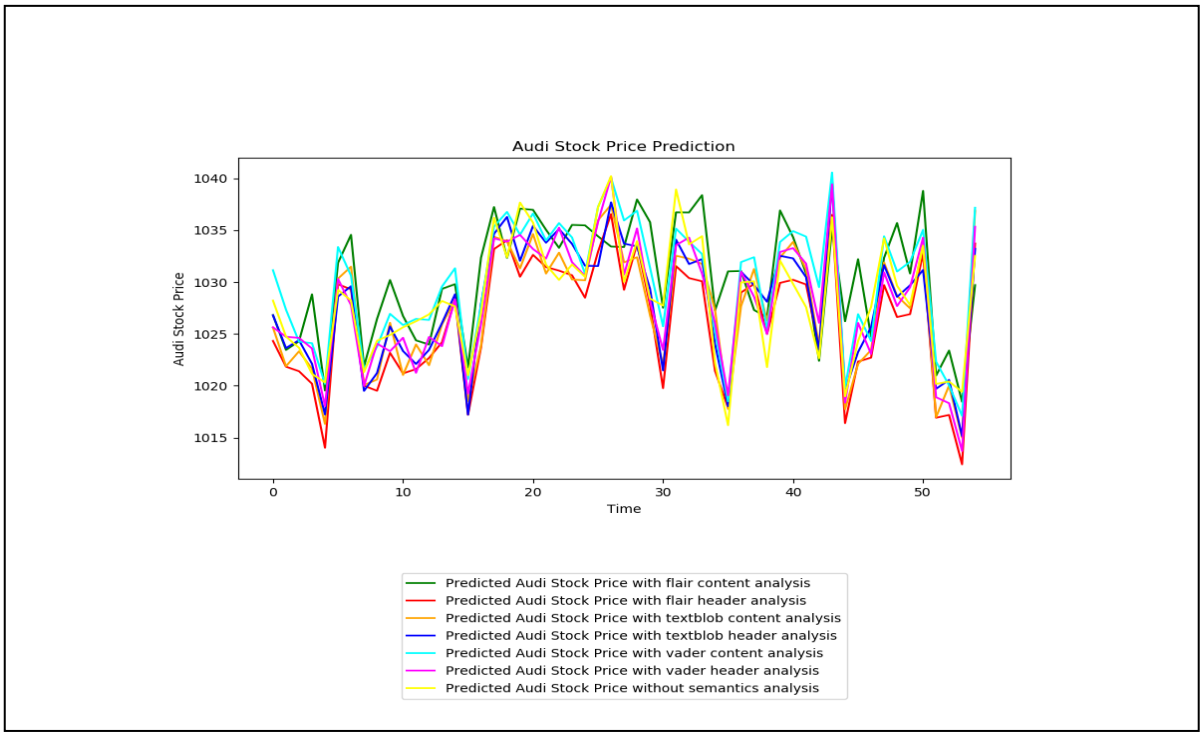


Figure 40: Stock Price Prediction Of Audi With Hourly Stock Price Data Using XGBoost

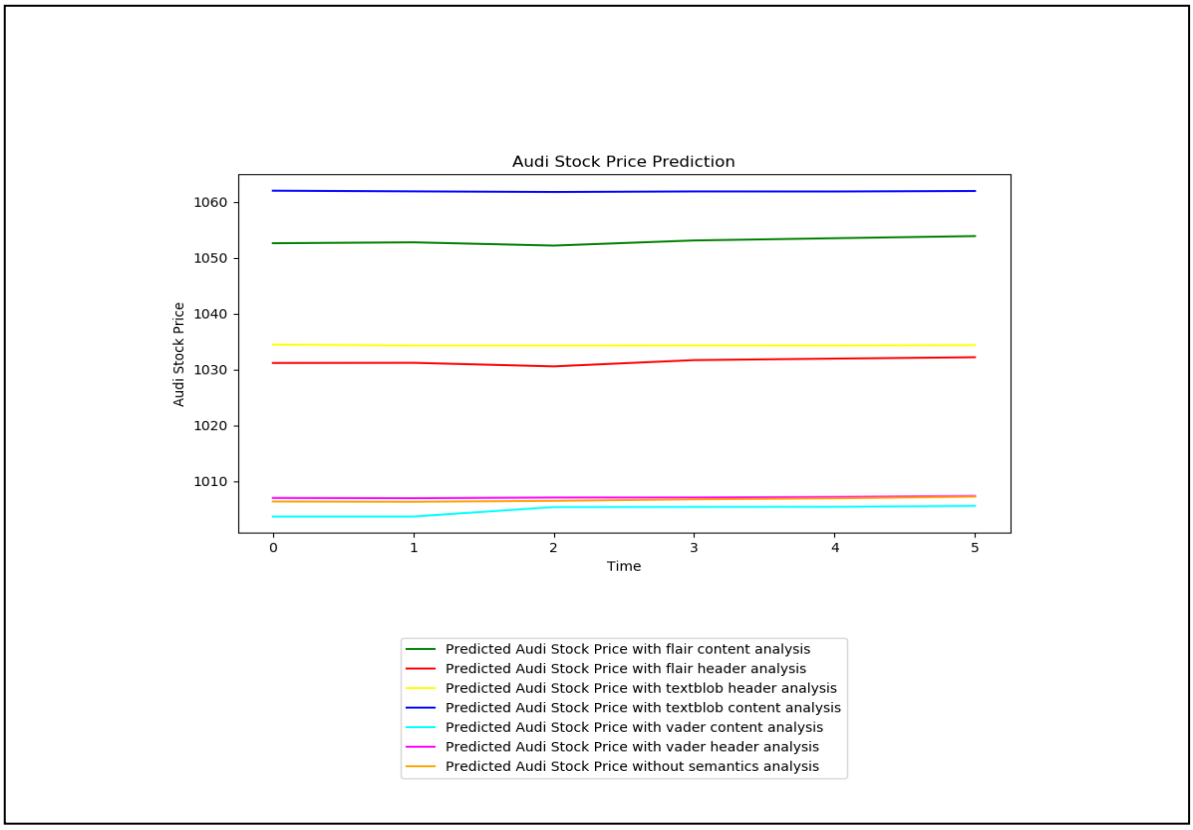


Figure 41: Stock Price Prediction Of Audi With Daily Stock Price Data Using LSTM

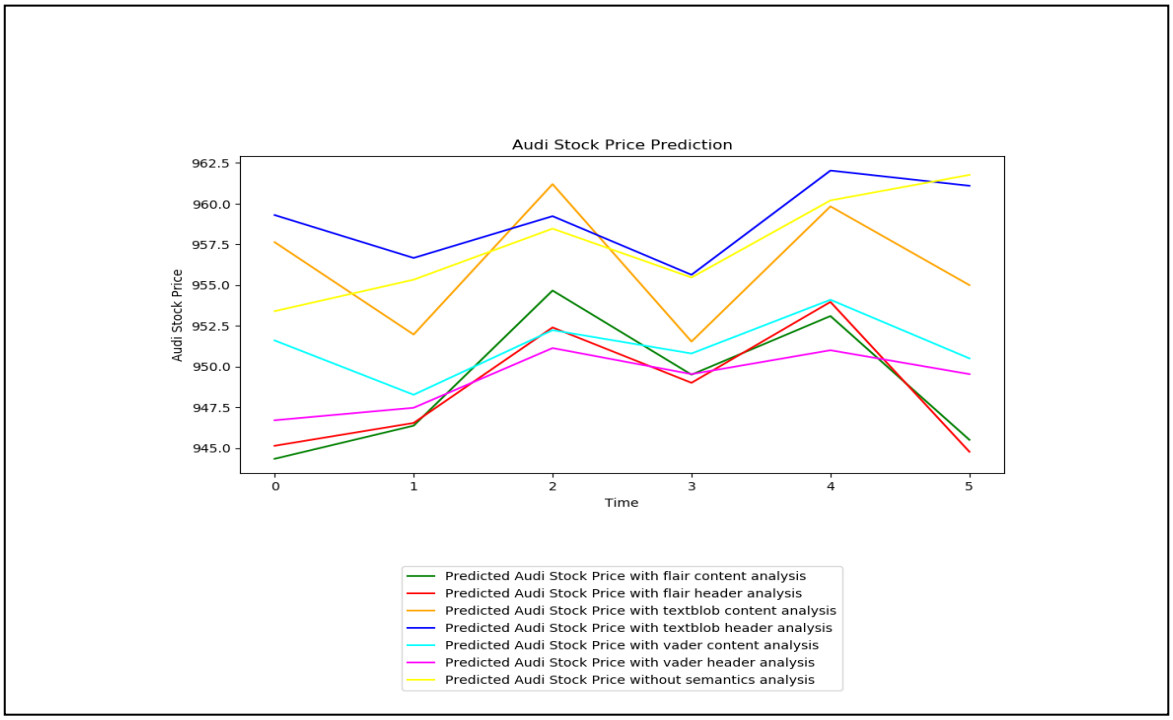


Figure 42: Stock Price Prediction Of Audi With Daily Stock Price Data Using RandomForest Base Model

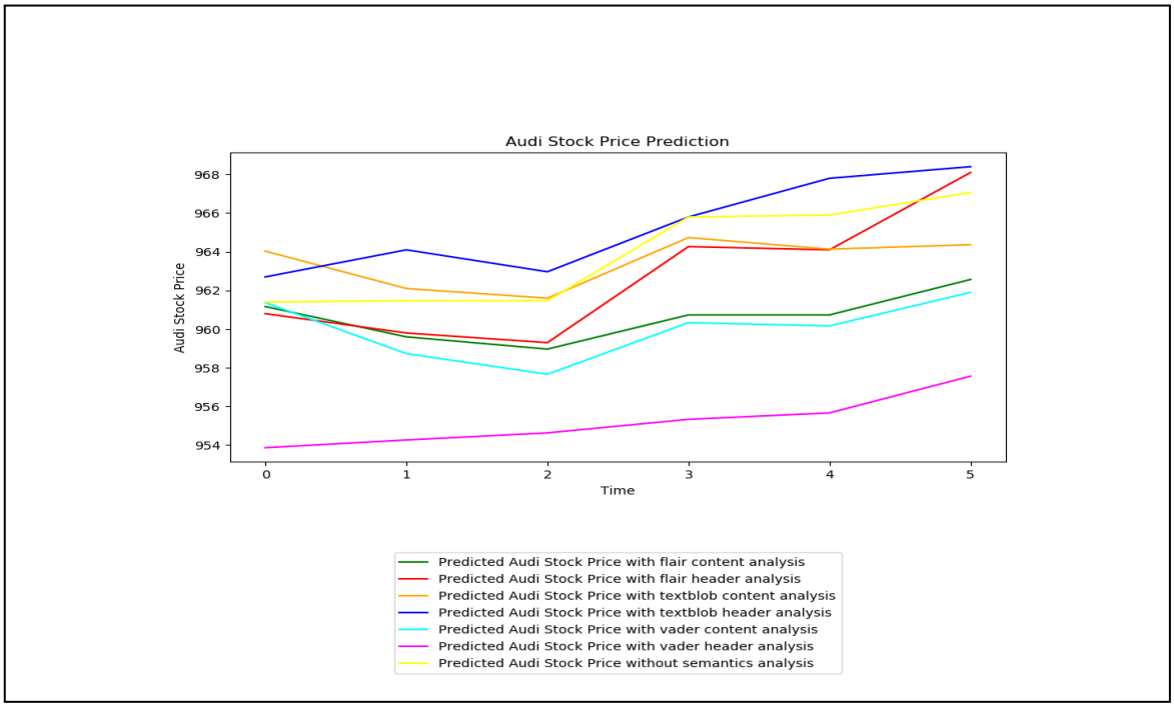


Figure 43: Stock Price Prediction Of Audi With Daily Stock Price Data Using RandomForest Feature Model

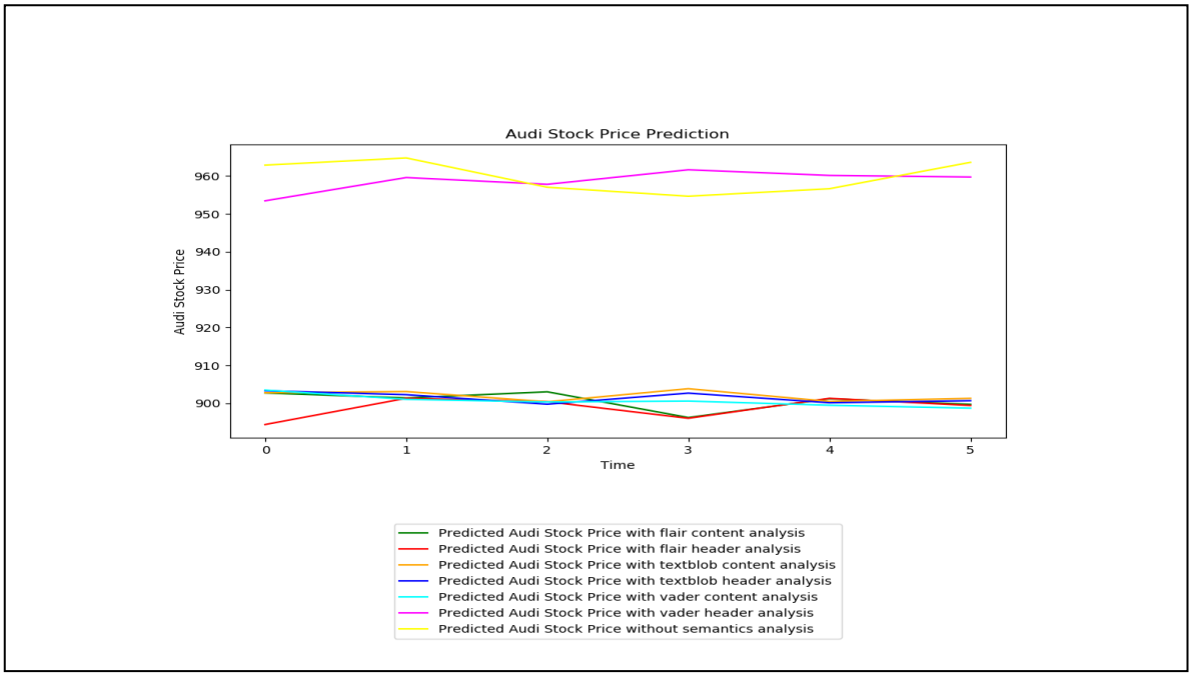


Figure 44: Stock Price Prediction Of Audi With Daily Stock Price Data Using XGBoost

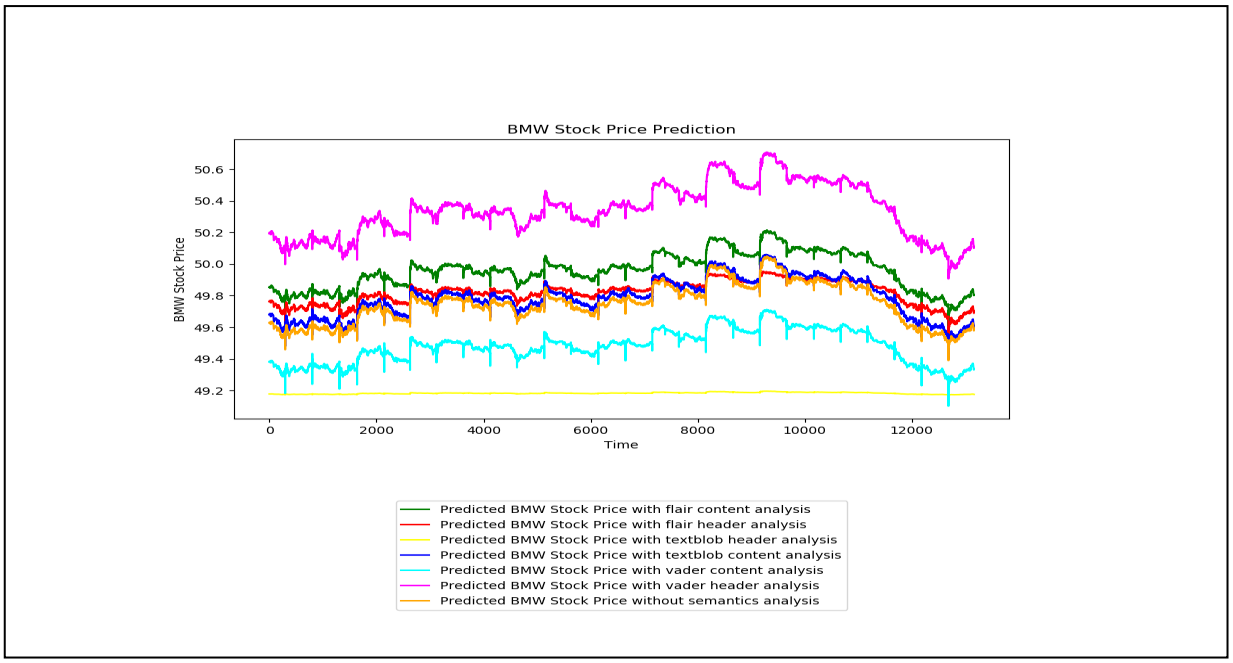


Figure 45: Stock Price Prediction Of BMW With Minutely Stock Price Data Using LSTM

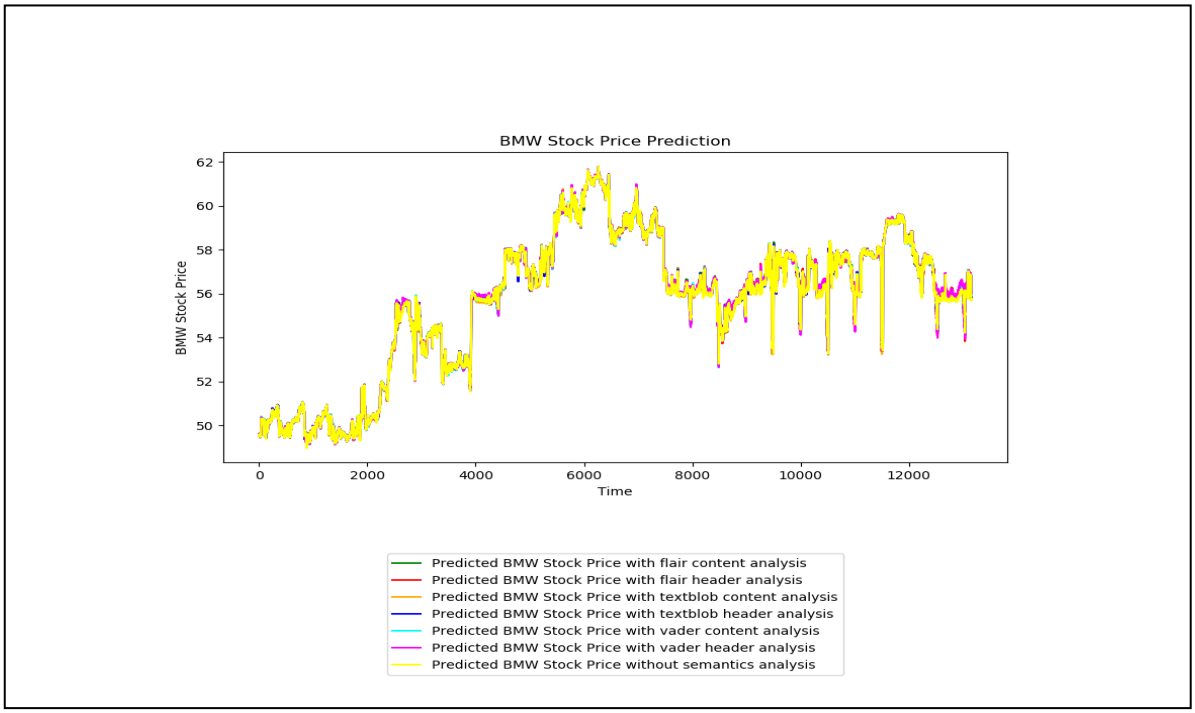


Figure 46: Stock Price Prediction Of BMW With Minutely Stock Price Data Using RandomForest Base Model

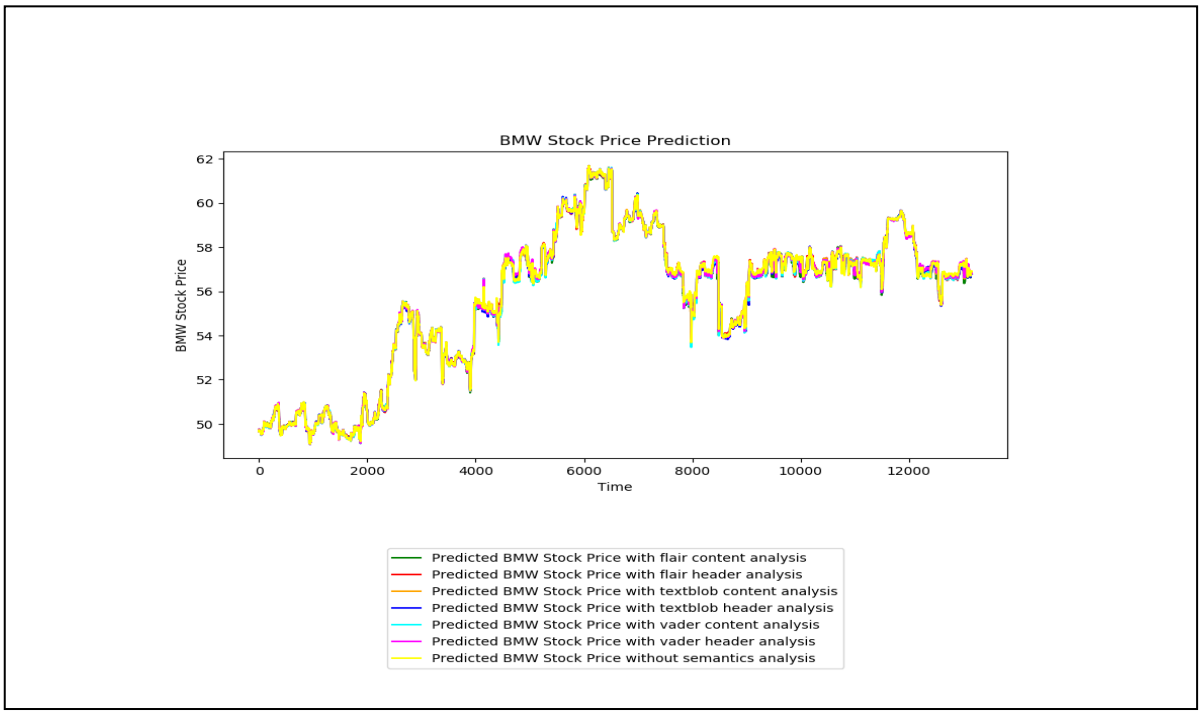


Figure 47: Stock Price Prediction Of BMW With Minutely Stock Price Data Using RandomForest Feature Model

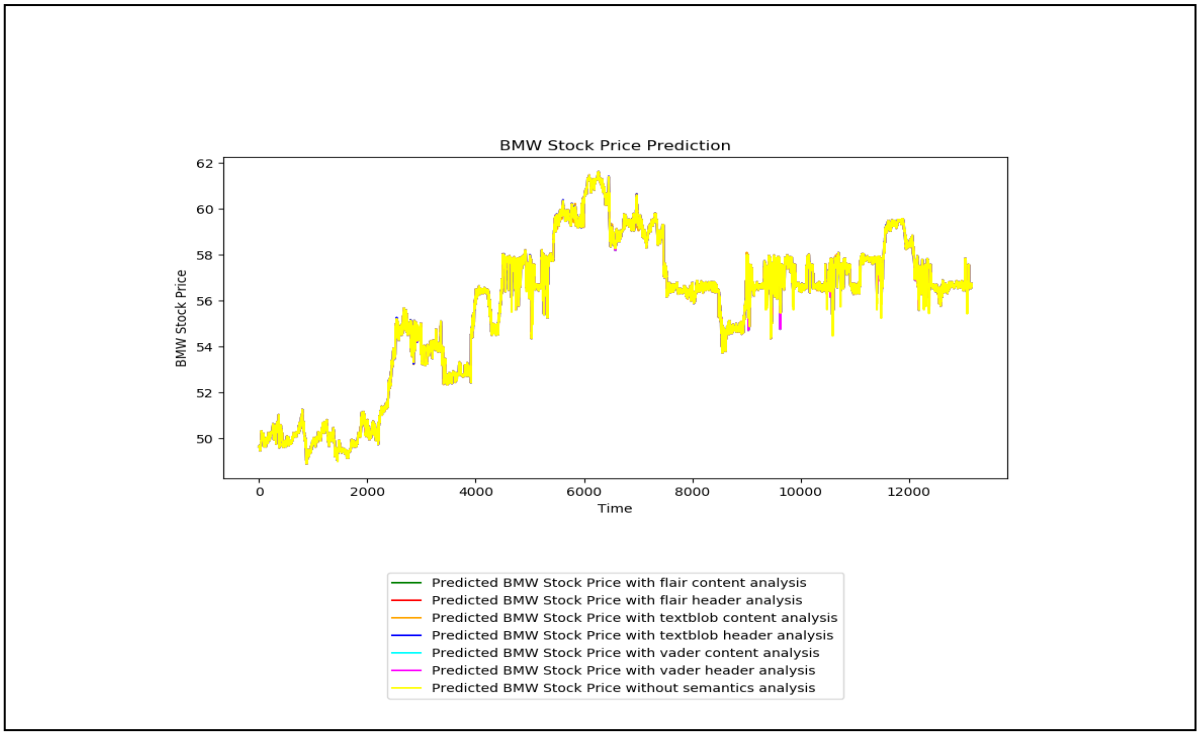


Figure 48: Stock Price Prediction Of BMW With Minutely Stock Price Data Using XGBoost

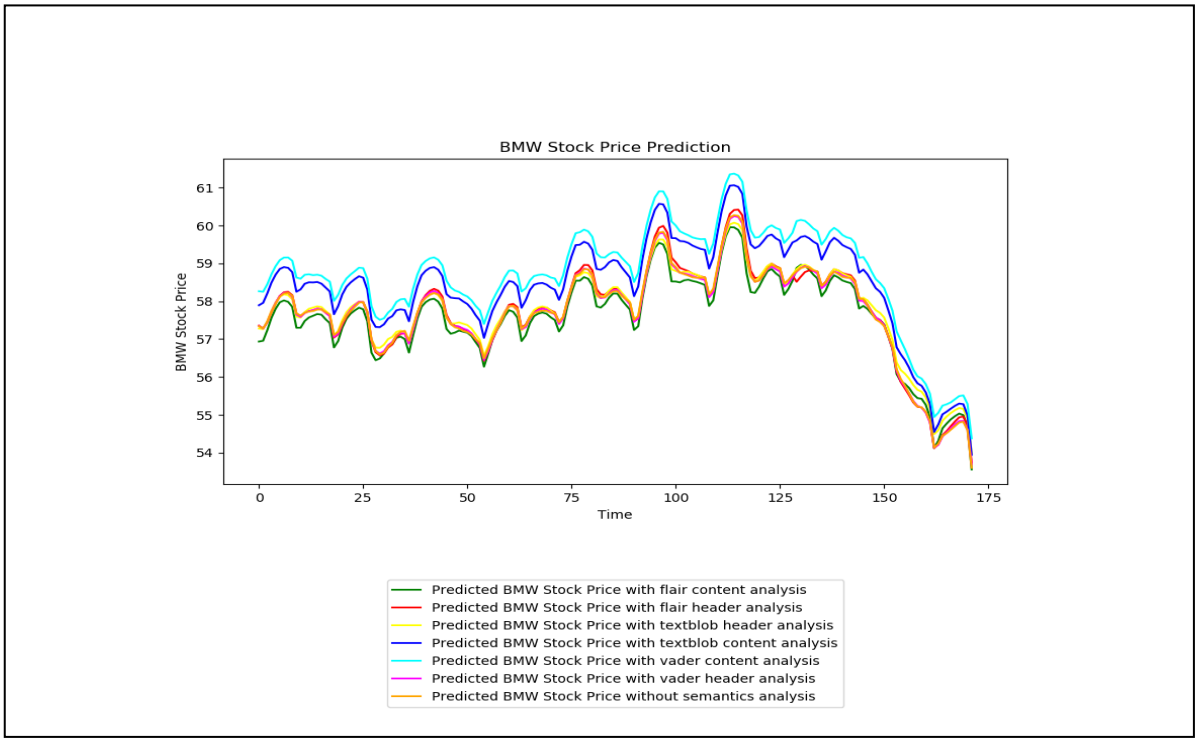


Figure 49: Stock Price Prediction Of BMW With Hourly Stock Price Data Using LSTM

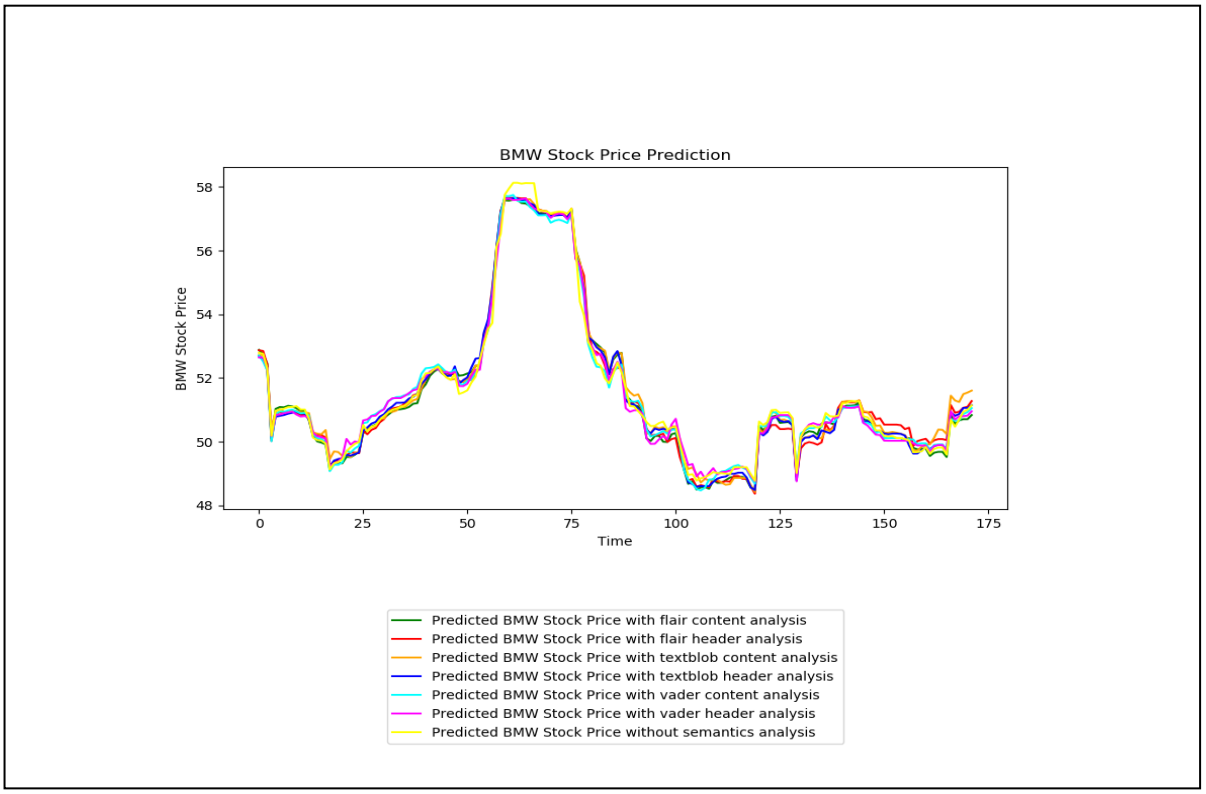


Figure 50: Stock Price Prediction Of BMW With Hourly Stock Price Data Using RandomForest Base Model

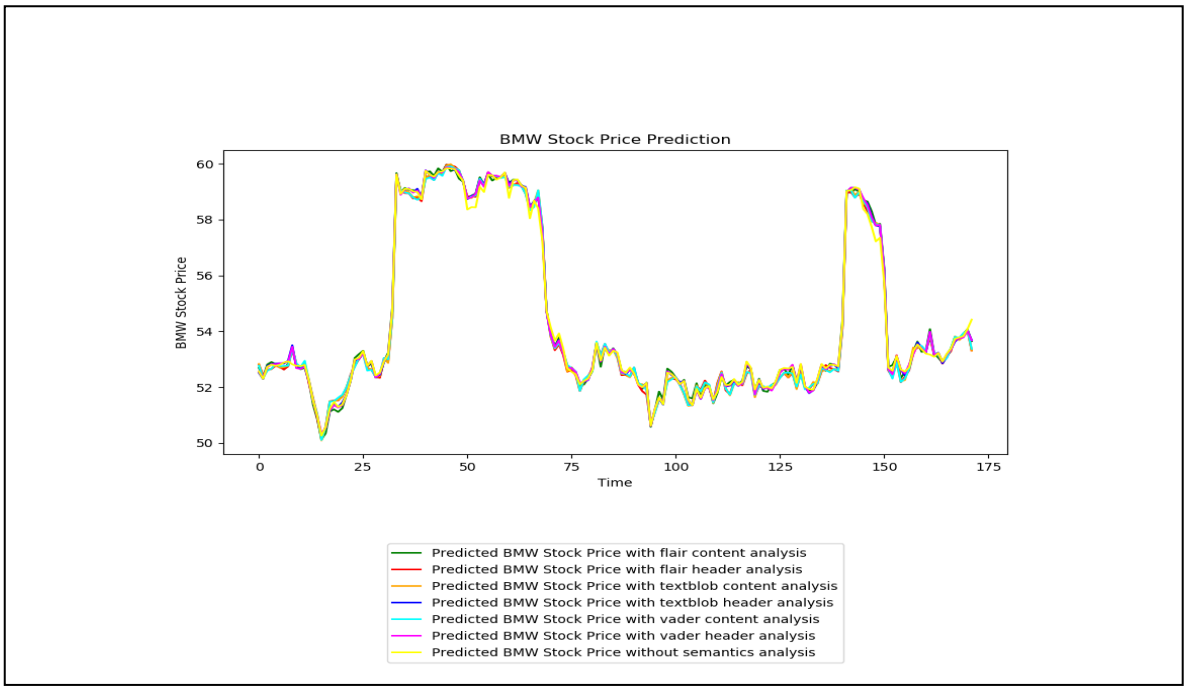


Figure 51: Stock Price Prediction Of BMW With Hourly Stock Price Data Using RandomForest Feature Model

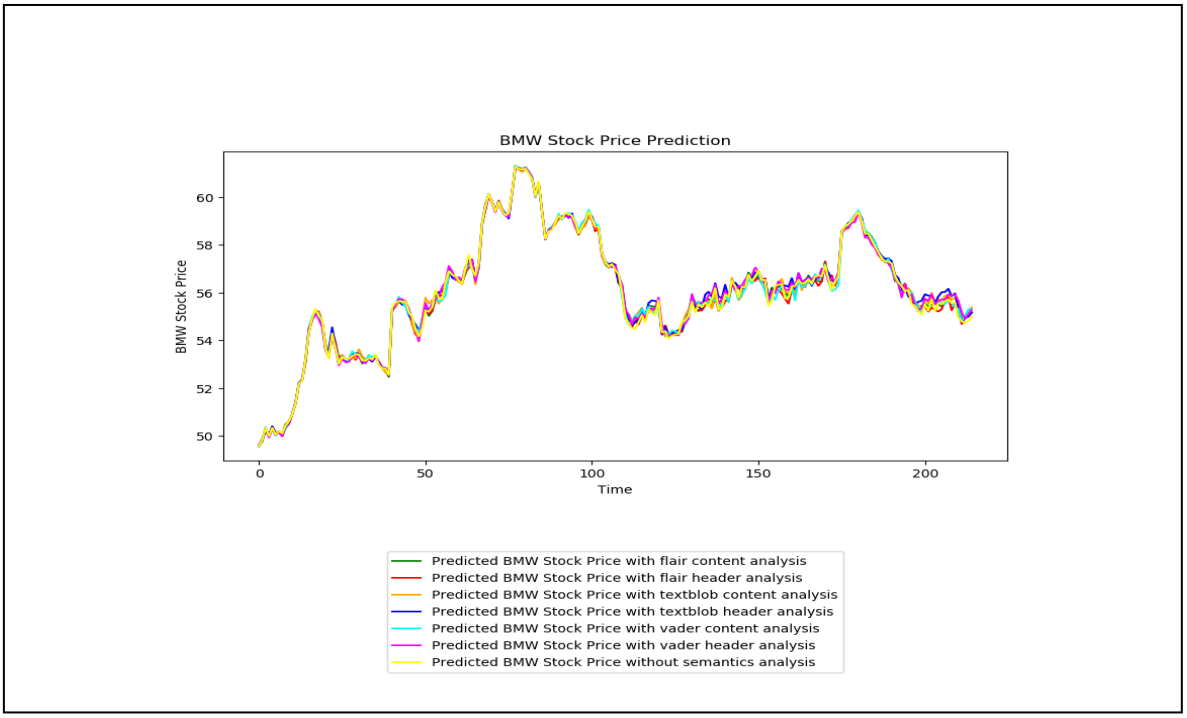


Figure 52: Stock Price Prediction Of BMW With Hourly Stock Price Data Using XGBoost

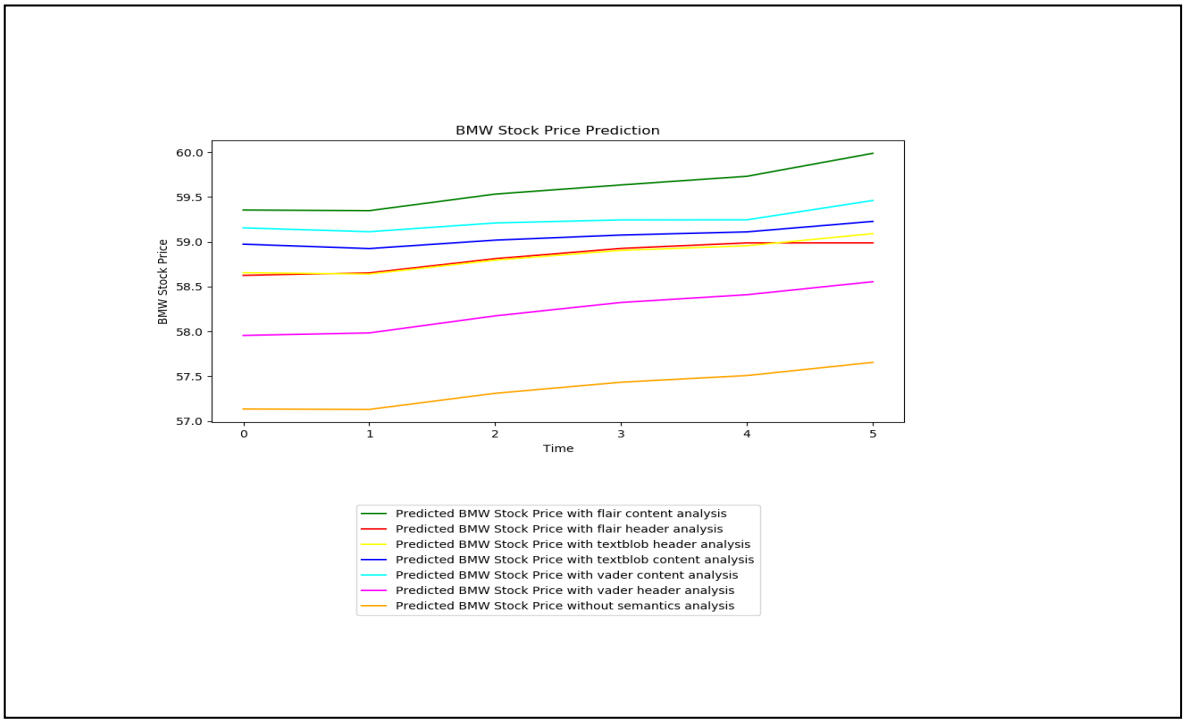


Figure 53: Stock Price Prediction Of BMW With Daily Stock Price Data Using LSTM

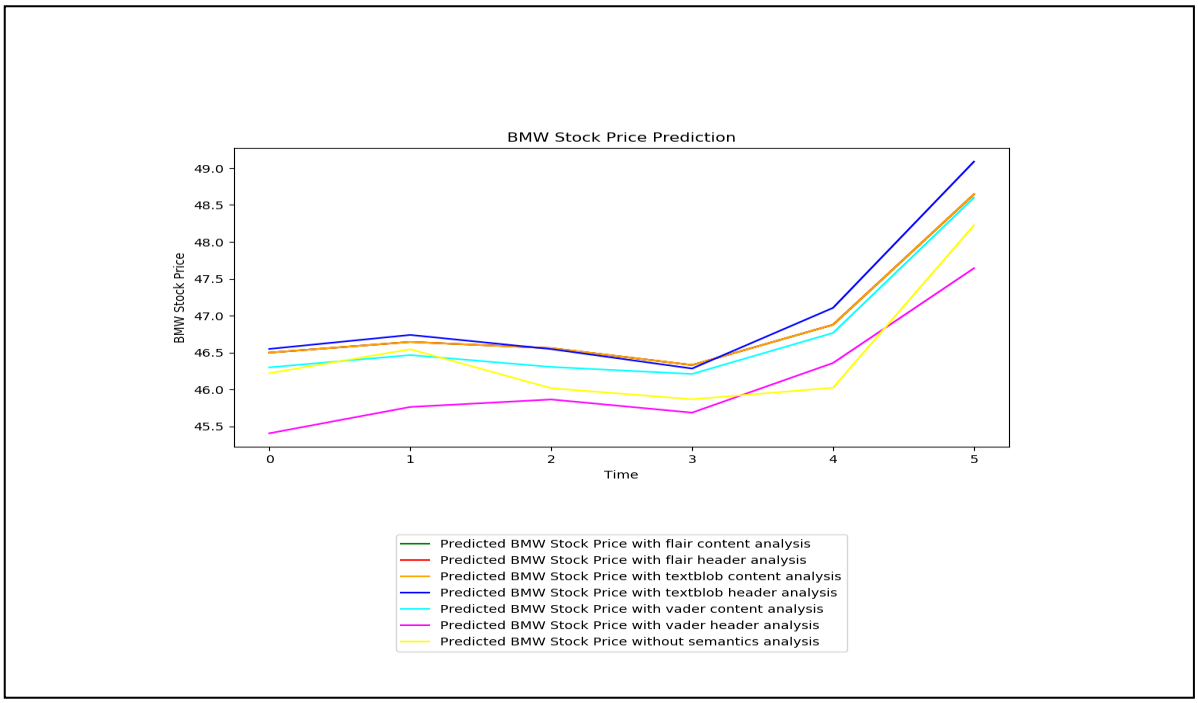


Figure 54: Stock Price Prediction Of BMW With Daily Stock Price Data Using RandomForest Base Model

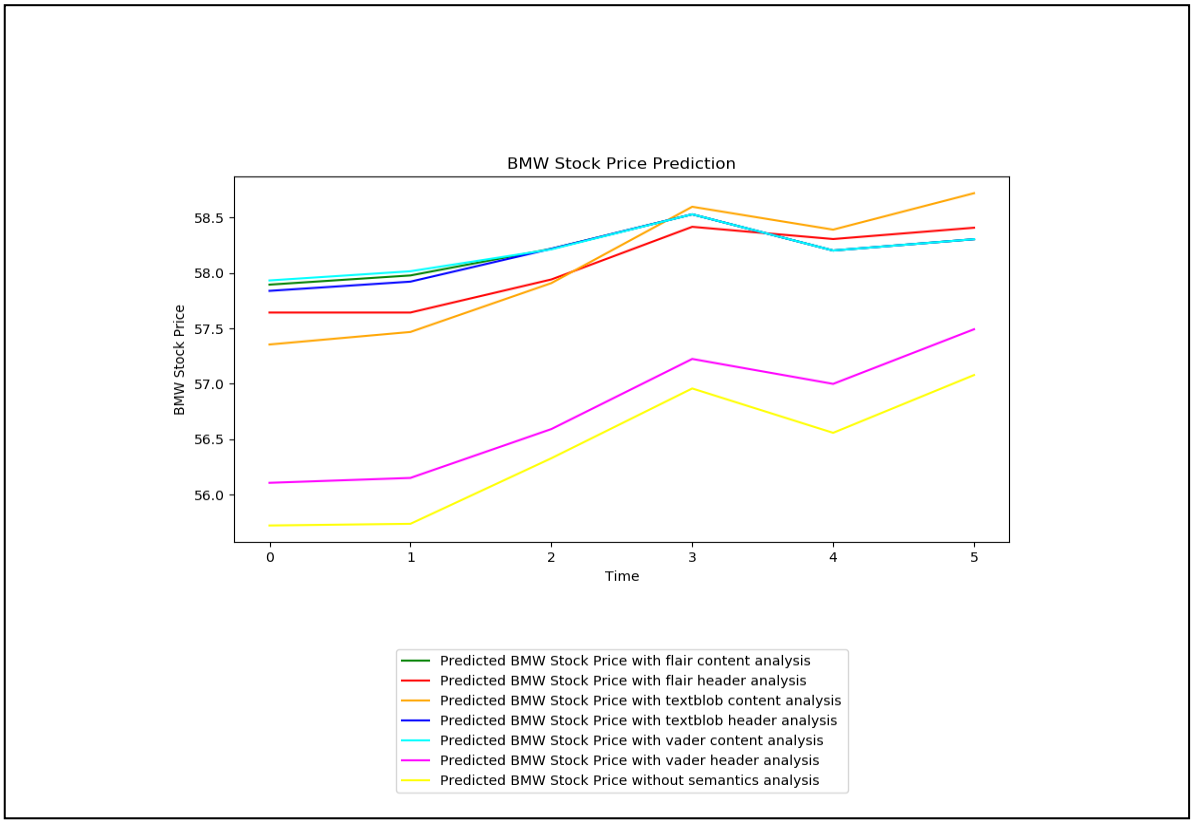


Figure 55: Stock Price Prediction Of BMW With Daily Stock Price Data Using RandomForest Feature Model

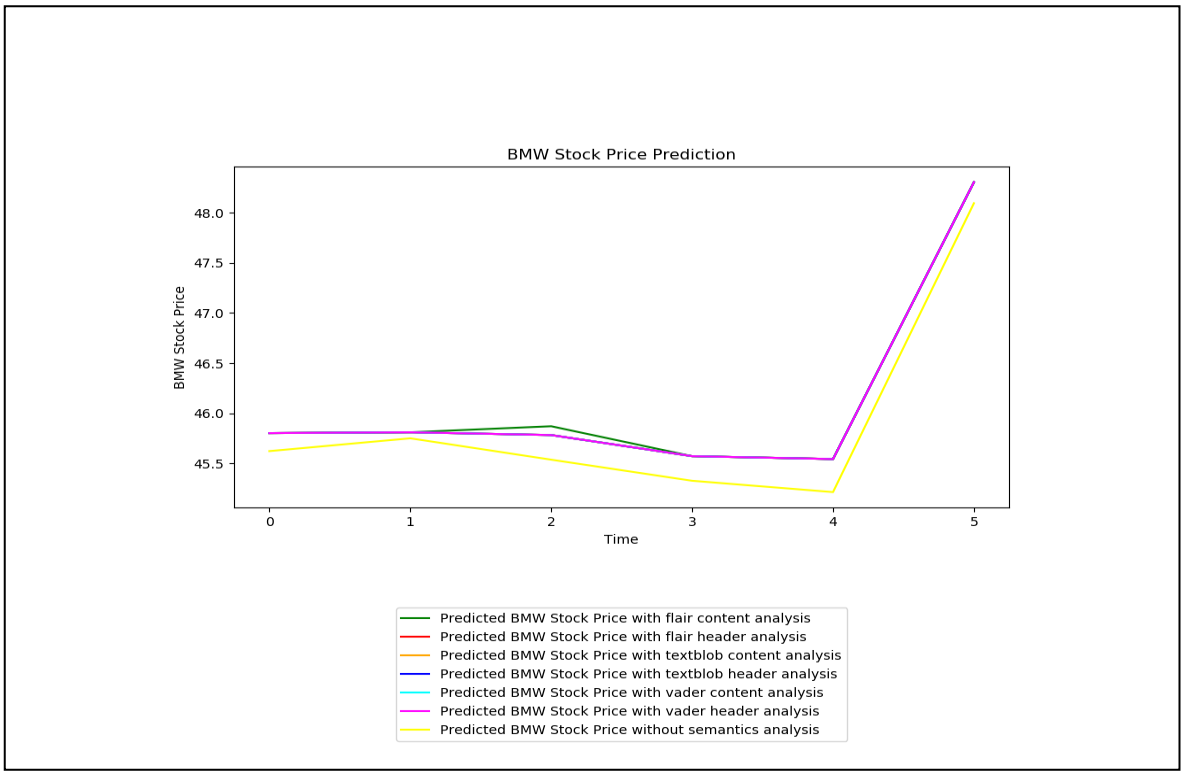


Figure 56: Stock Price Prediction Of BMW With Daily Stock Price Data Using XGBoost

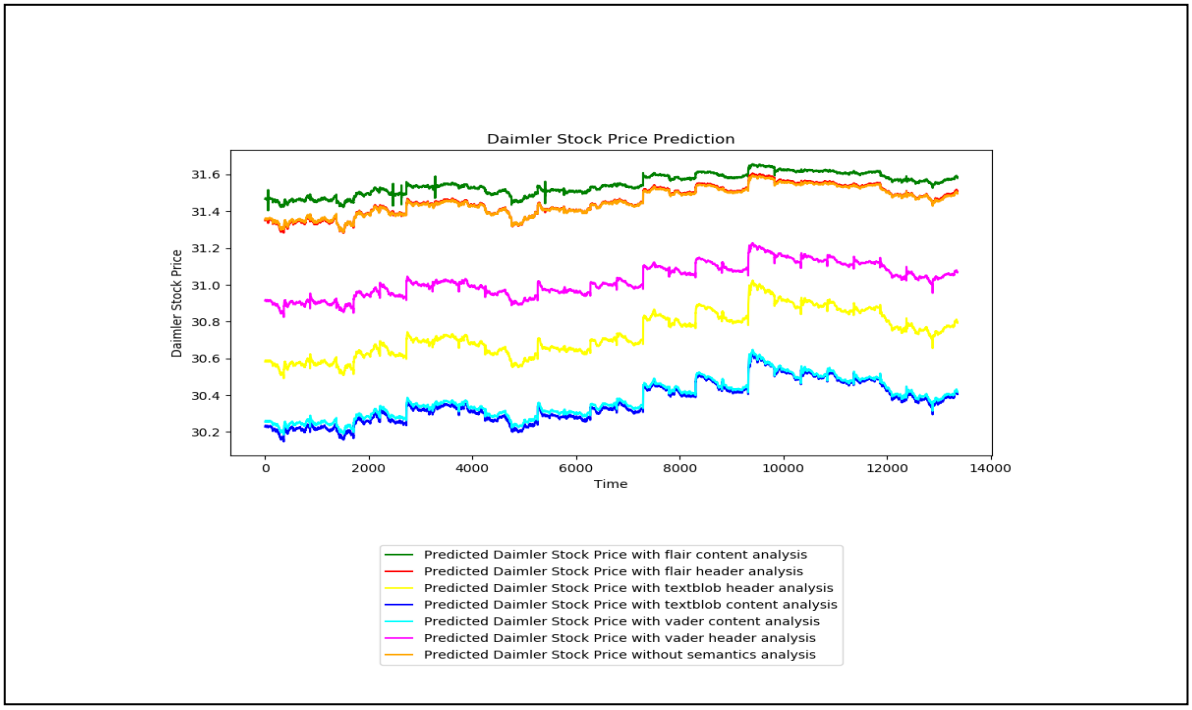


Figure 57: Stock Price Prediction Of Daimler With Minutely Stock Price Data Using LSTM

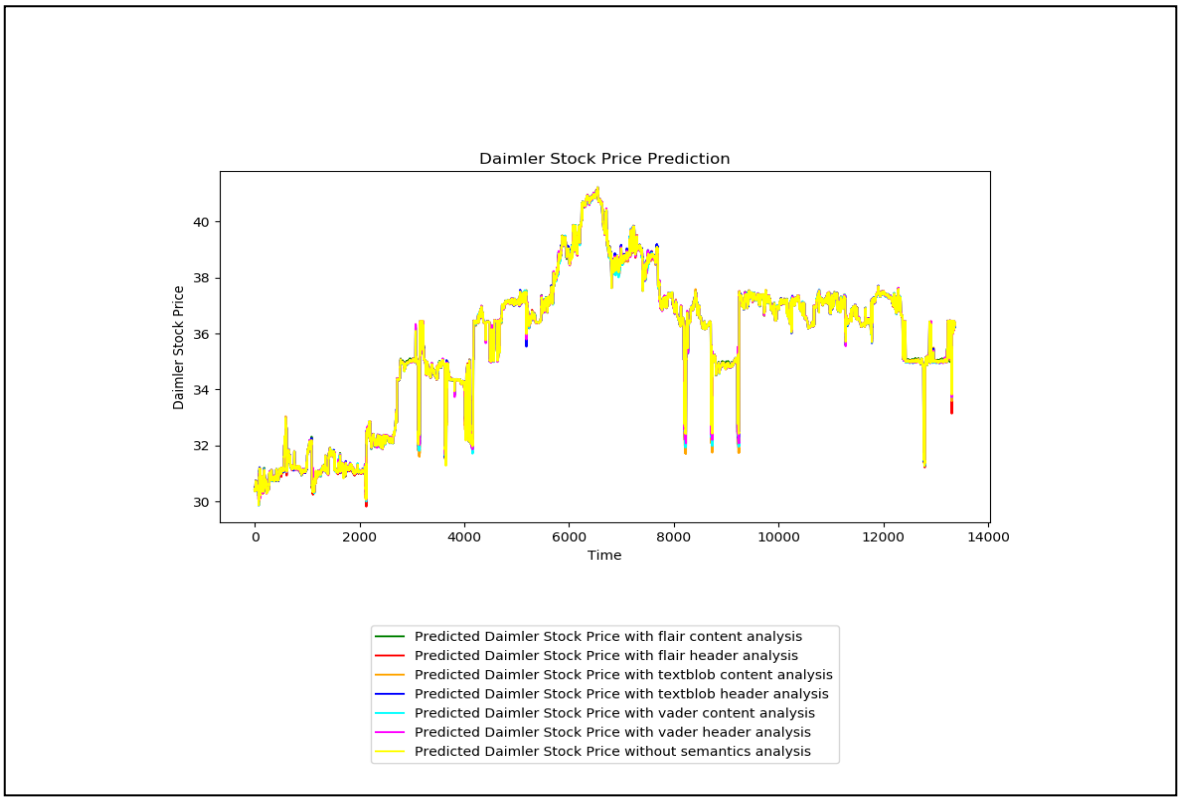


Figure 58: Stock Price Prediction Of Daimler With Minutely Stock Price Data Using Random-Forest Base Model

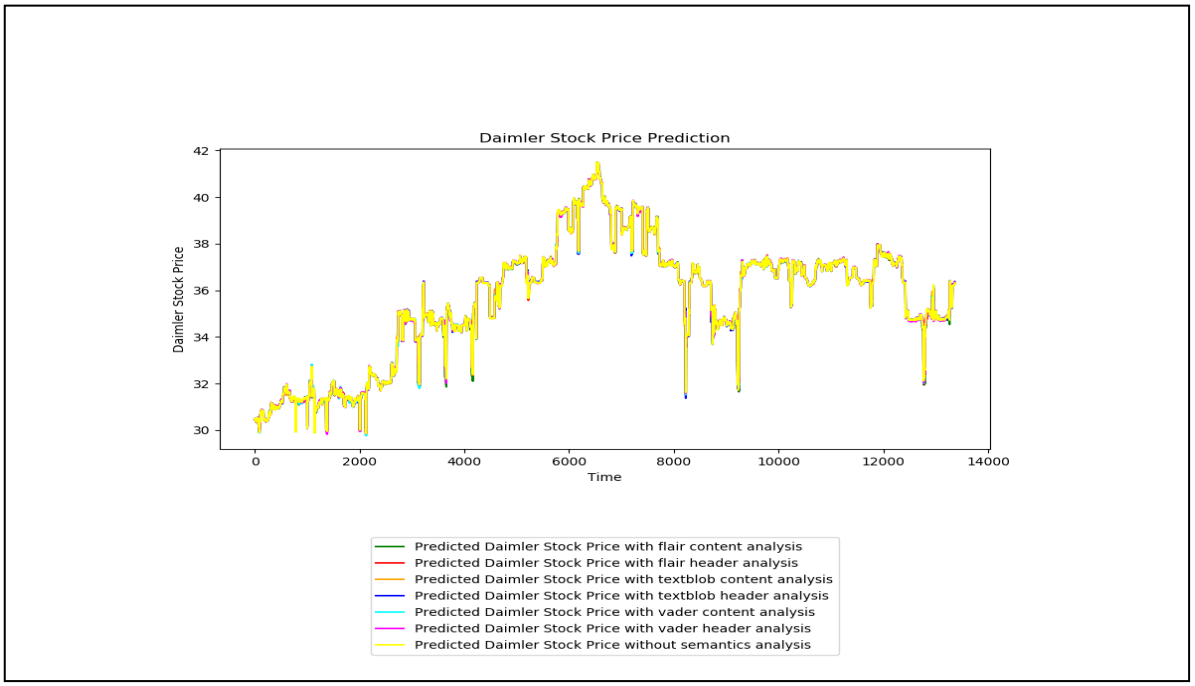


Figure 59: Stock Price Prediction Of Daimler With Minutely Stock Price Data Using Random-Forest Feature Model

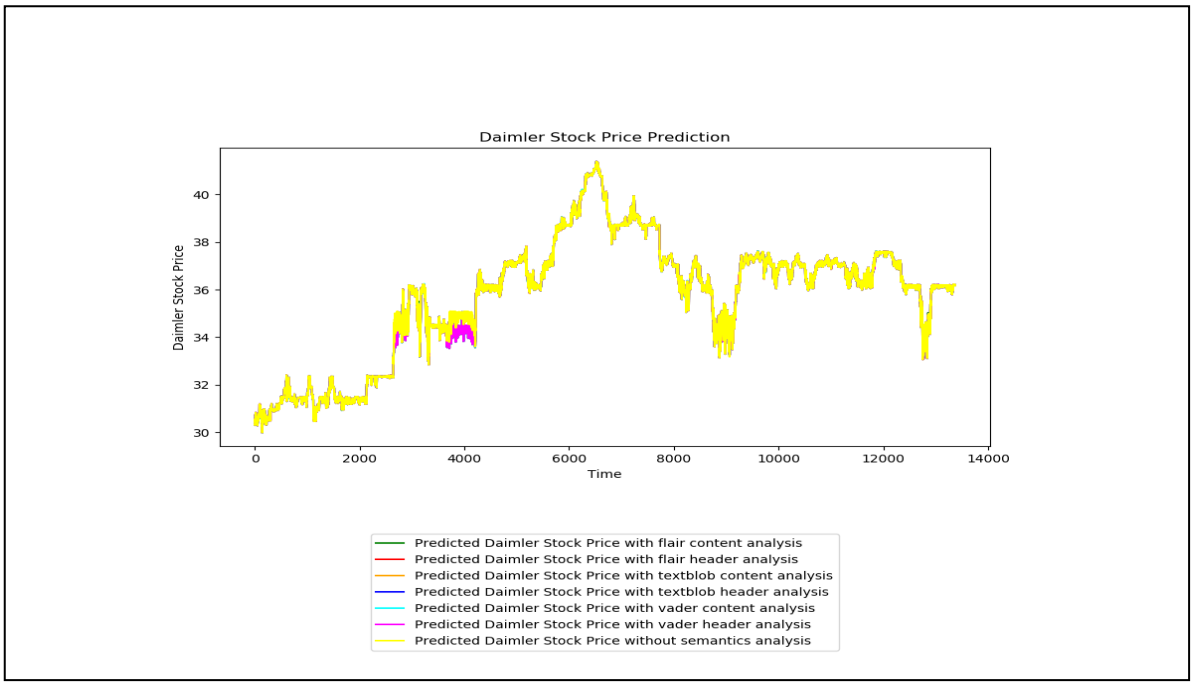


Figure 60: Stock Price Prediction Of Daimler With Minutely Stock Price Data Using XGBoost

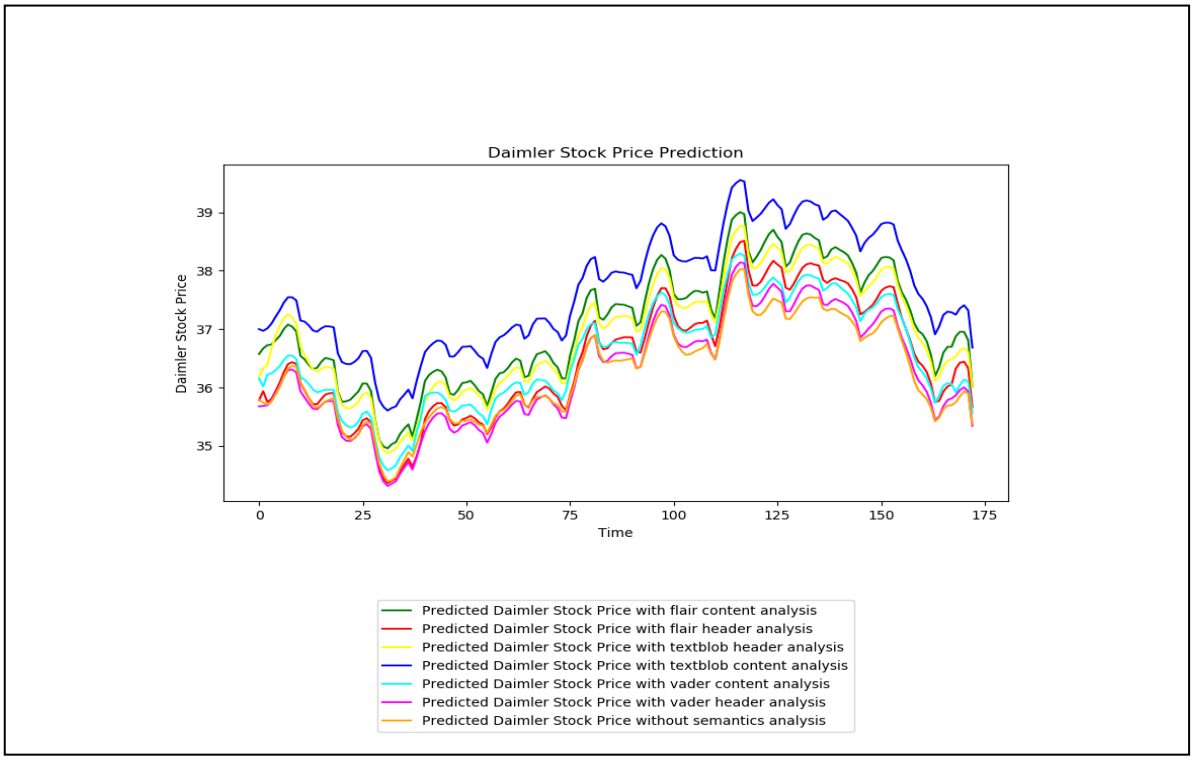


Figure 61: Stock Price Prediction Of Daimler With Hourly Stock Price Data Using LSTM

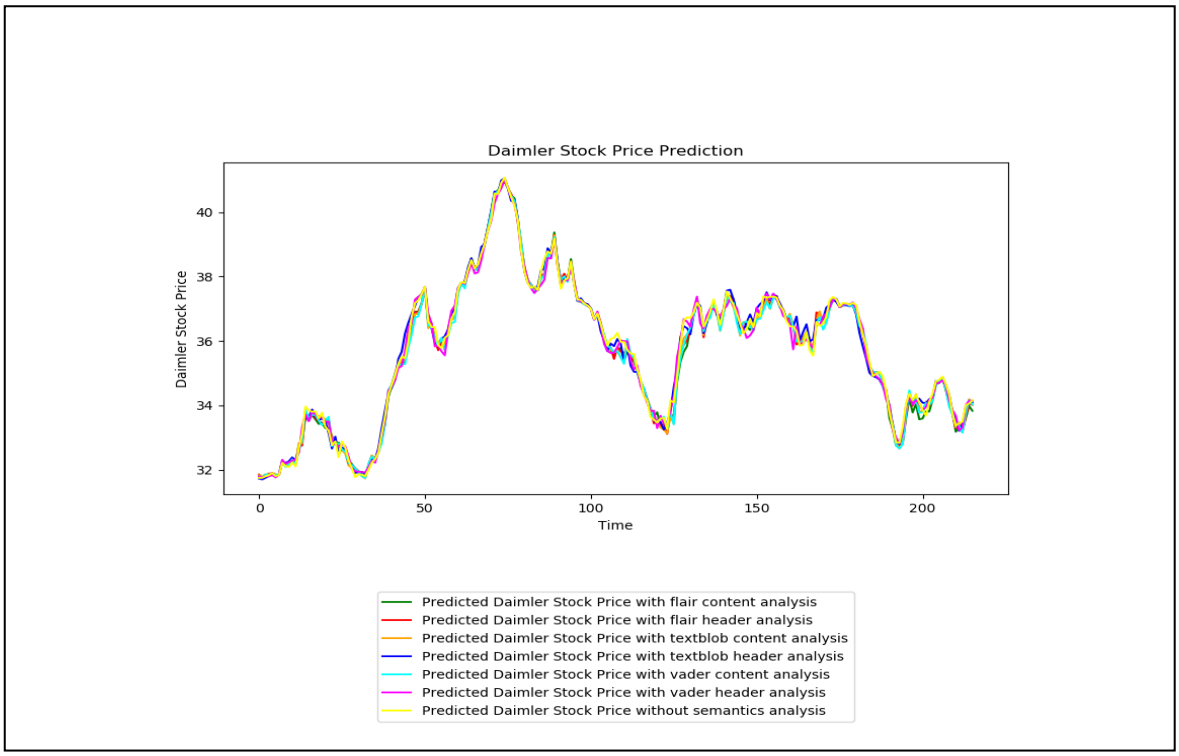


Figure 62: Stock Price Prediction Of Daimler With Hourly Stock Price Data Using RandomForest Base Model

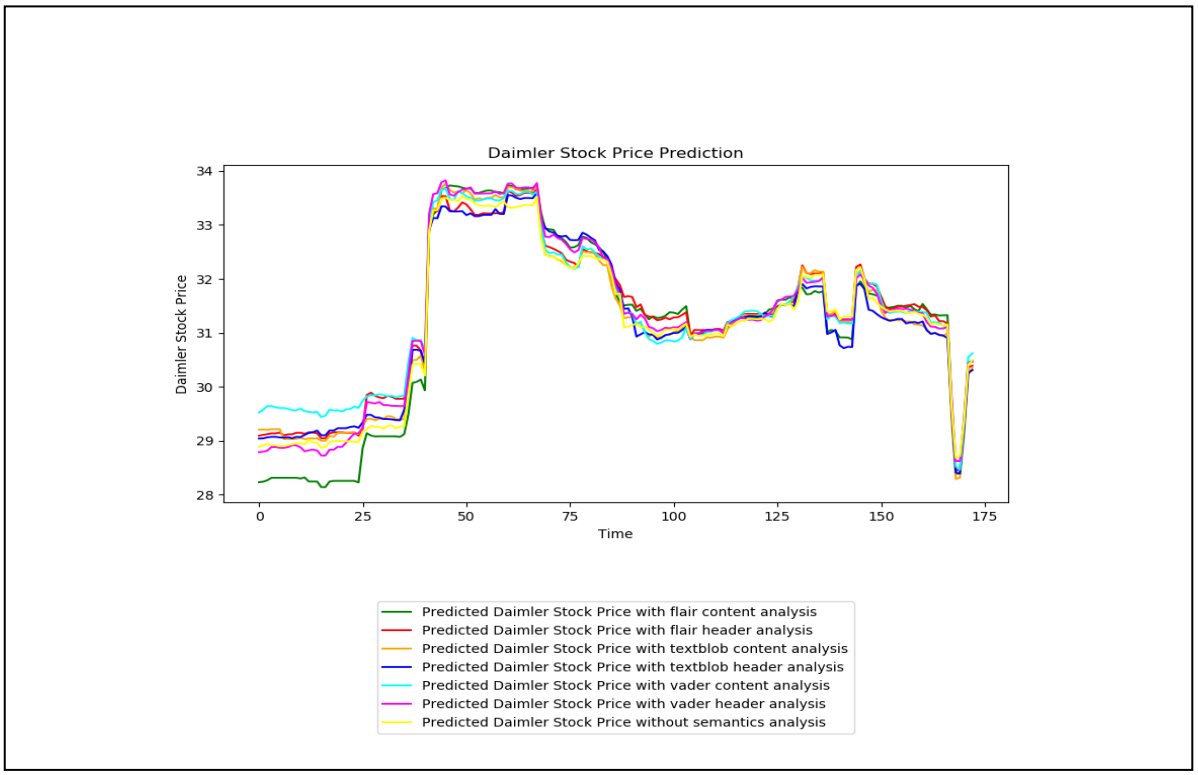


Figure 63: Stock Price Prediction Of Daimler With Hourly Stock Price Data Using RandomForest Feature Model

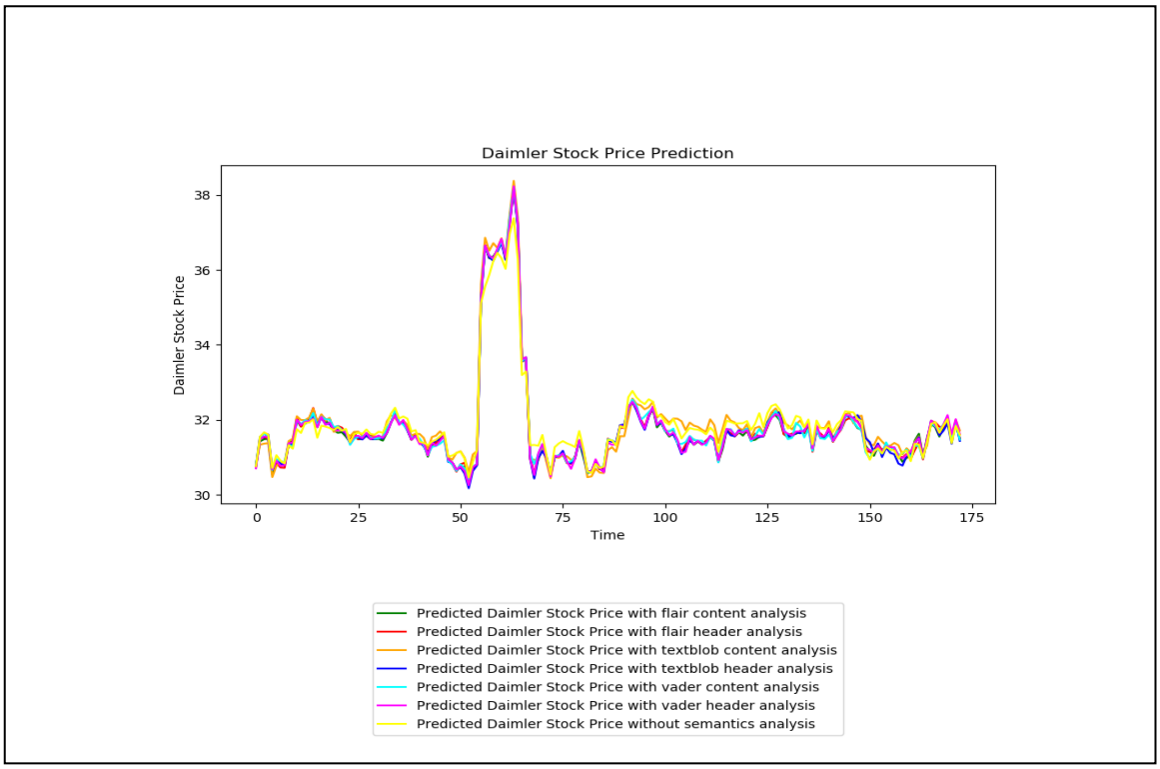


Figure 64: Stock Price Prediction Of Daimler With Hourly Stock Price Data Using XGBoost

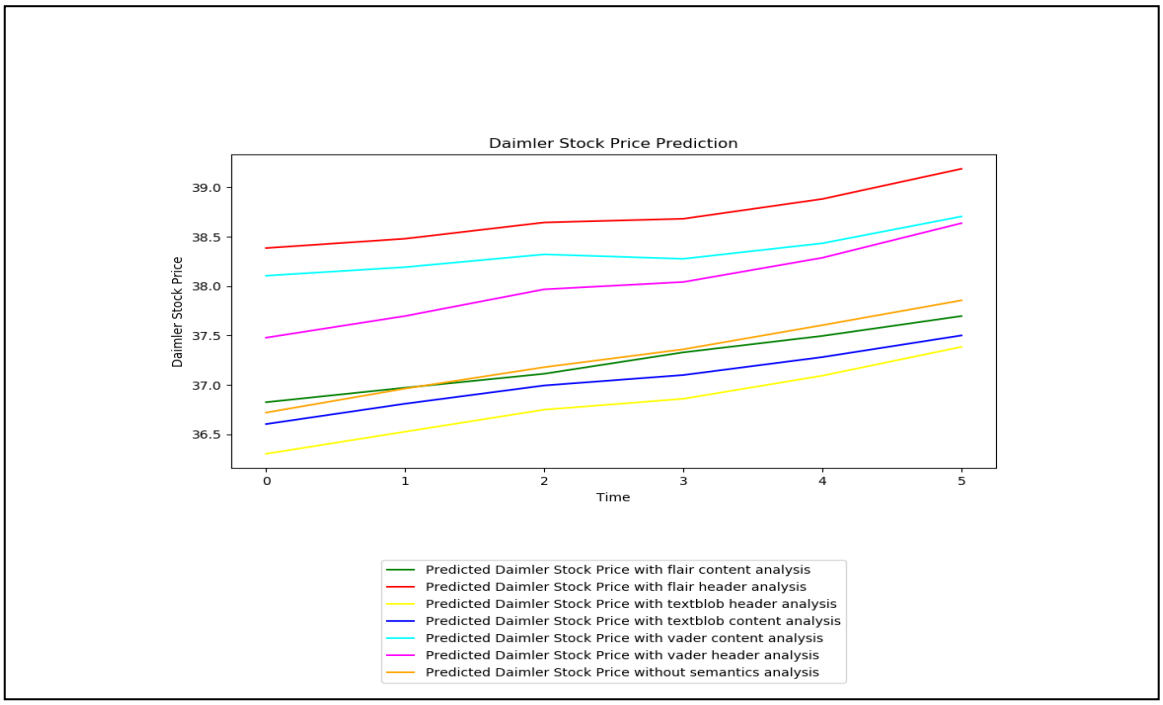


Figure 65: Stock Price Prediction Of Daimler With Daily Stock Price Data Using LSTM

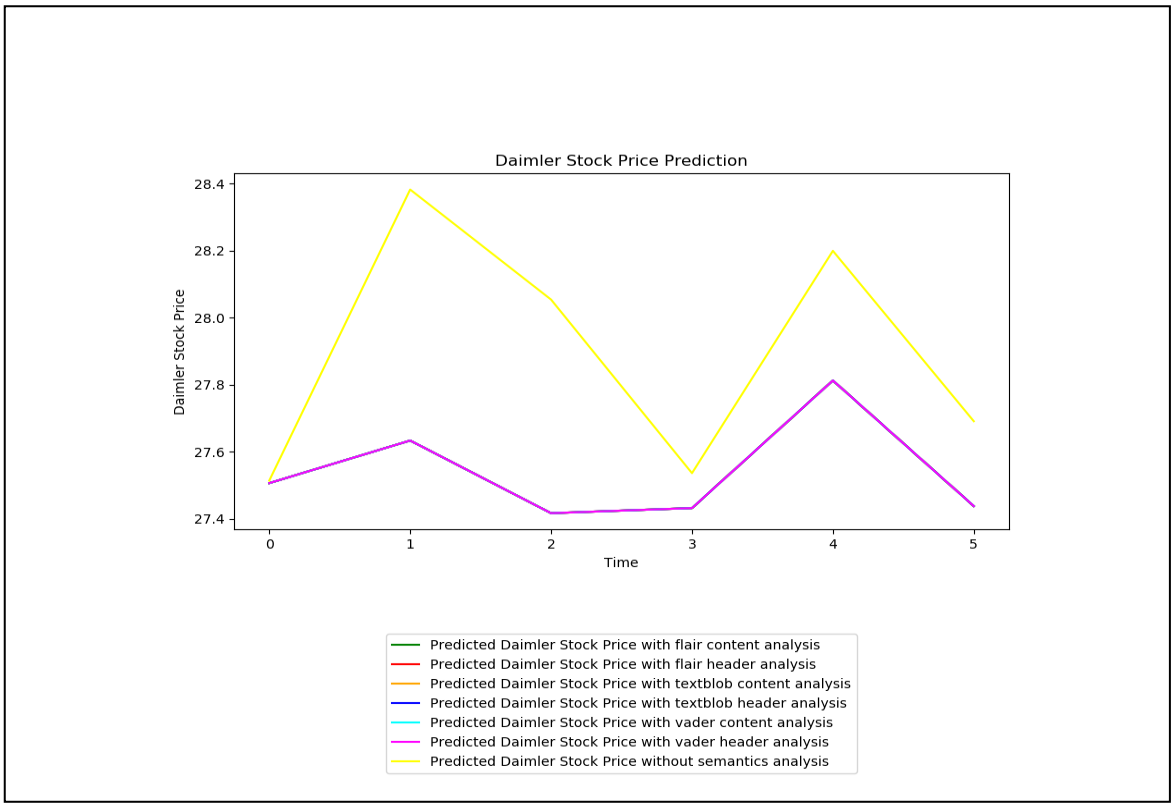


Figure 66: Stock Price Prediction Of Daimler With Daily Stock Price Data Using RandomForest Base Model

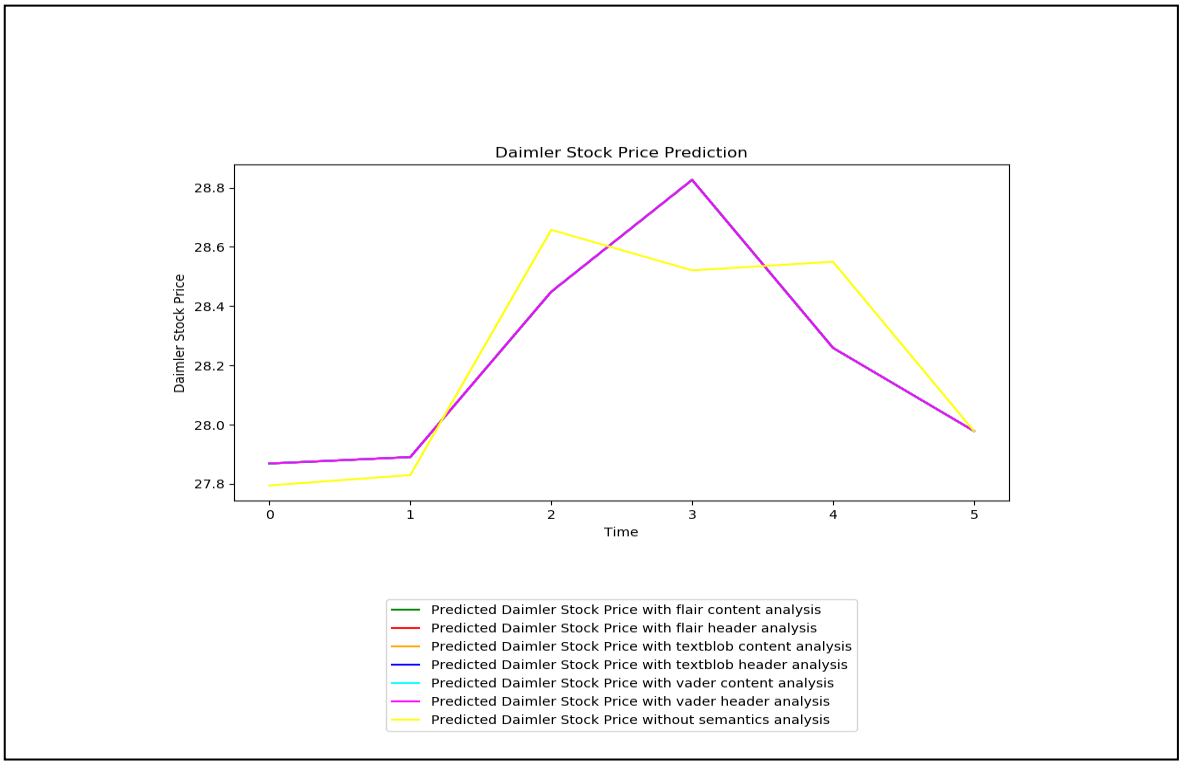


Figure 67: Stock Price Prediction Of Daimler With Daily Stock Price Data Using RandomForest Feature Model

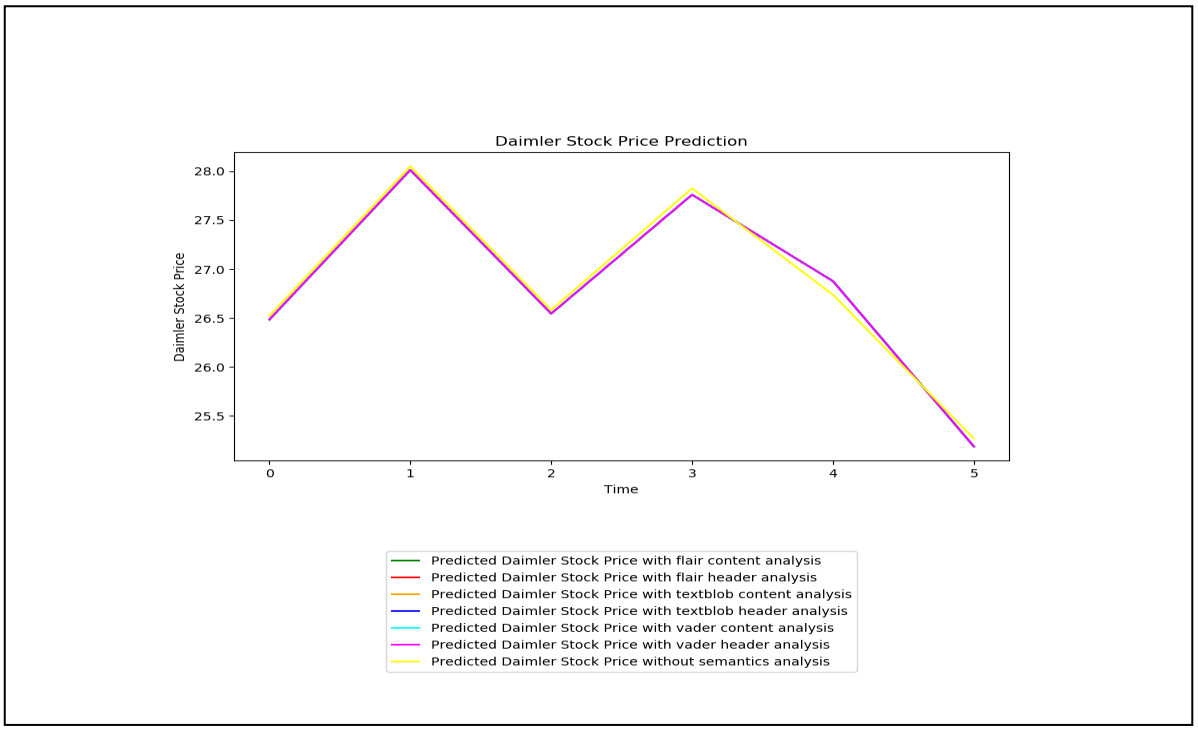


Figure 68: Stock Price Prediction Of Daimler With Daily Stock Price Data Using XGBoost

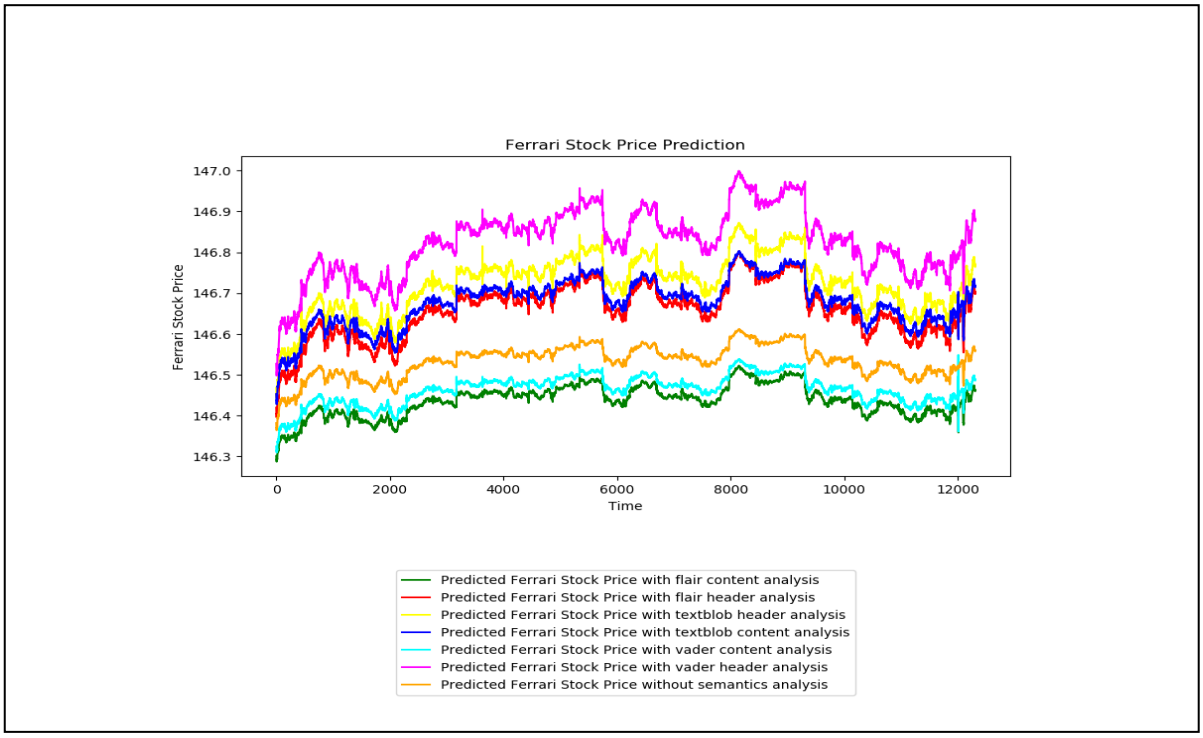


Figure 69: Stock Price Prediction Of Ferrari With Minutely Stock Price Data Using LSTM

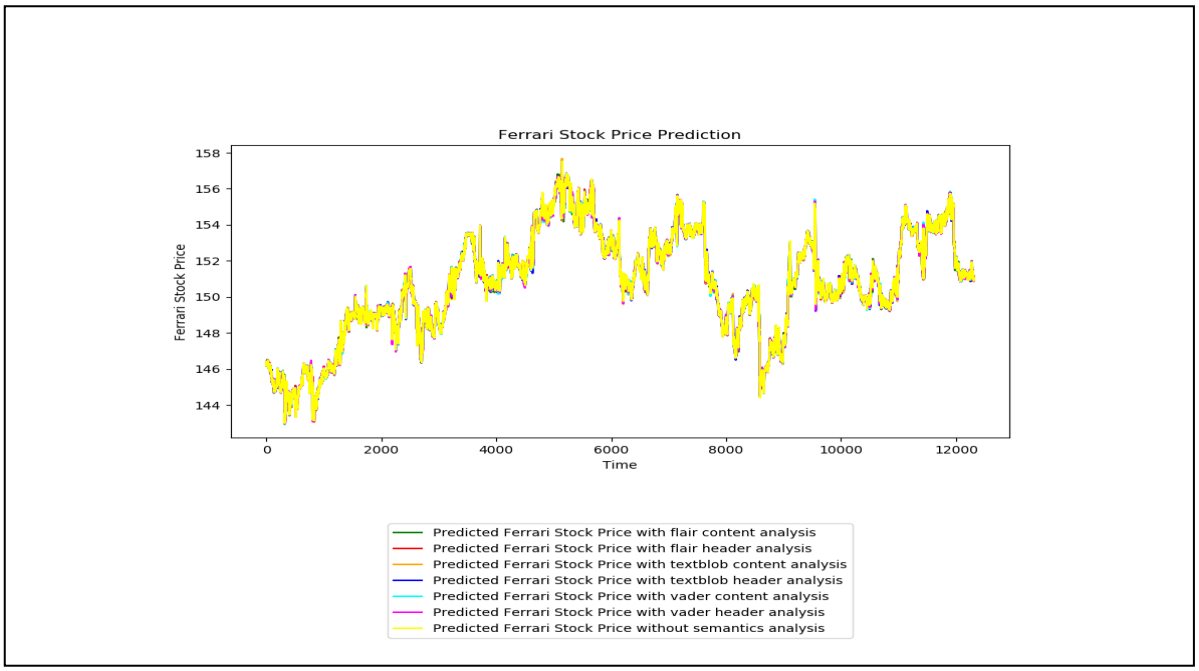


Figure 70: Stock Price Prediction Of Ferrari With Minutely Stock Price Data Using RandomForest Base Model

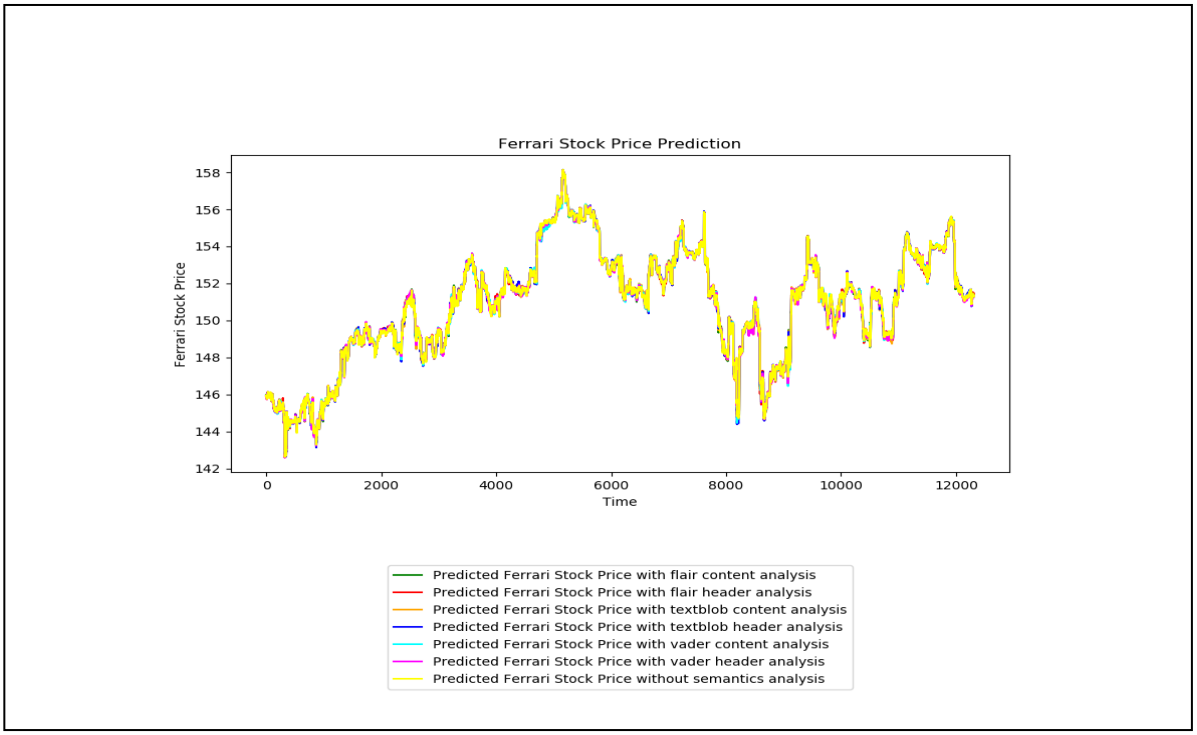


Figure 71: Stock Price Prediction Of Ferrari With Minutely Stock Price Data Using RandomForest Feature Model

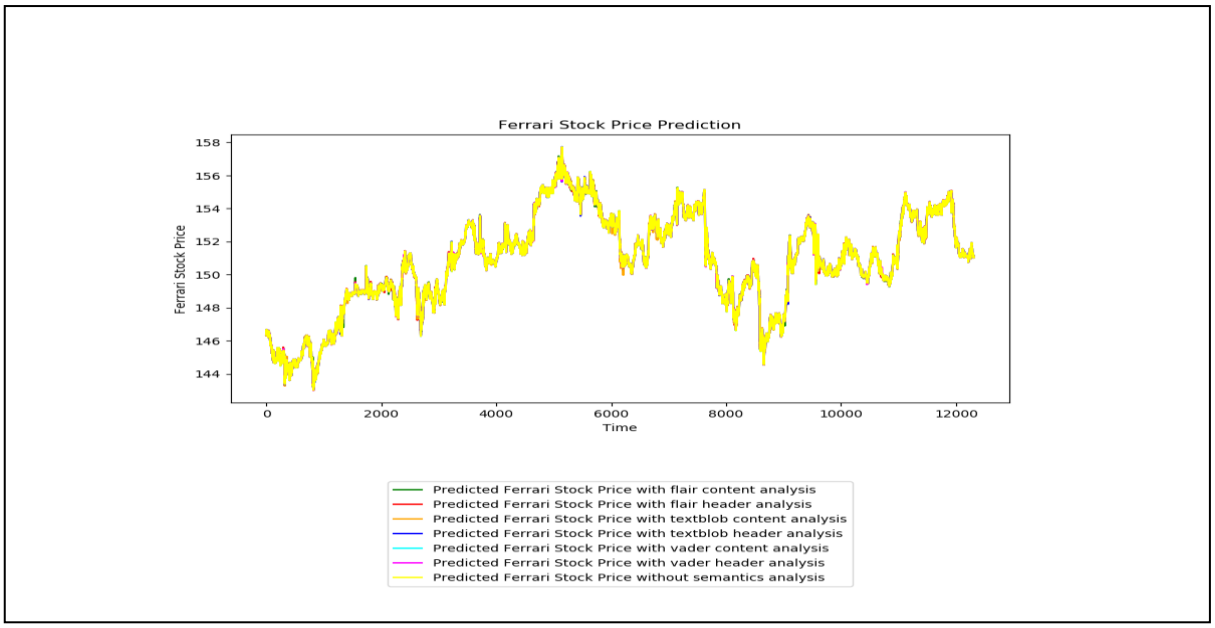


Figure 72: Stock Price Prediction Of Ferrari With Minutely Stock Price Data Using XGBoost

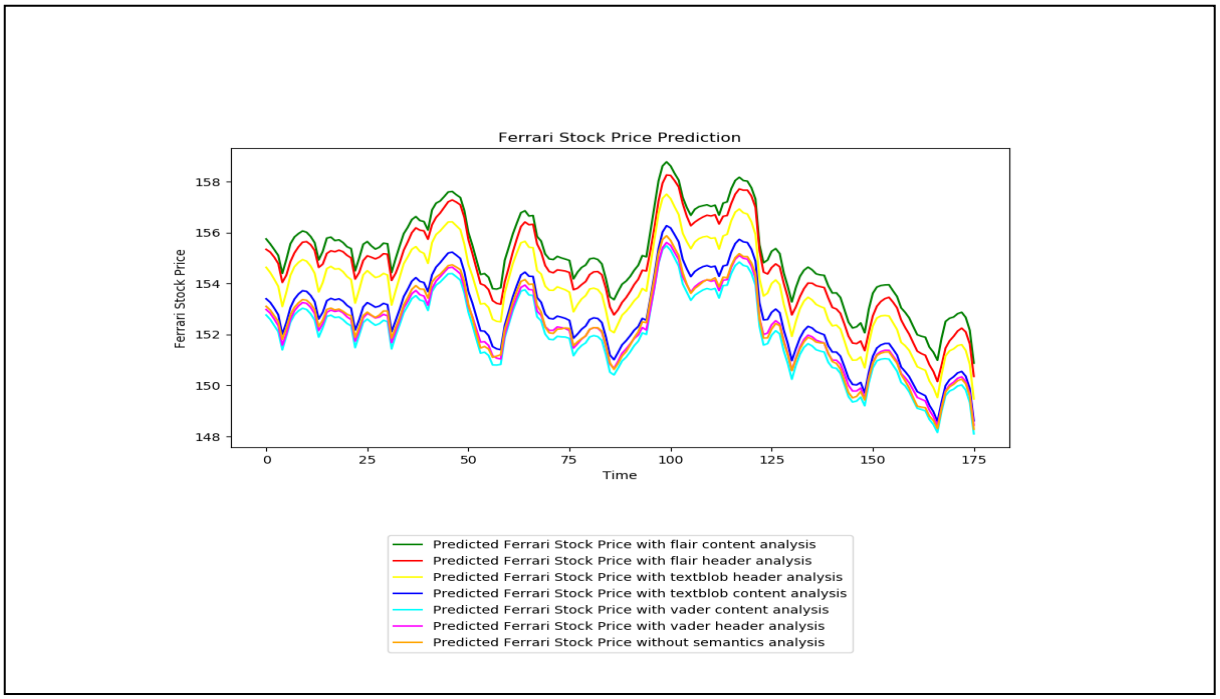


Figure 73: Stock Price Prediction Of Ferrari With Hourly Stock Price Data Using LSTM

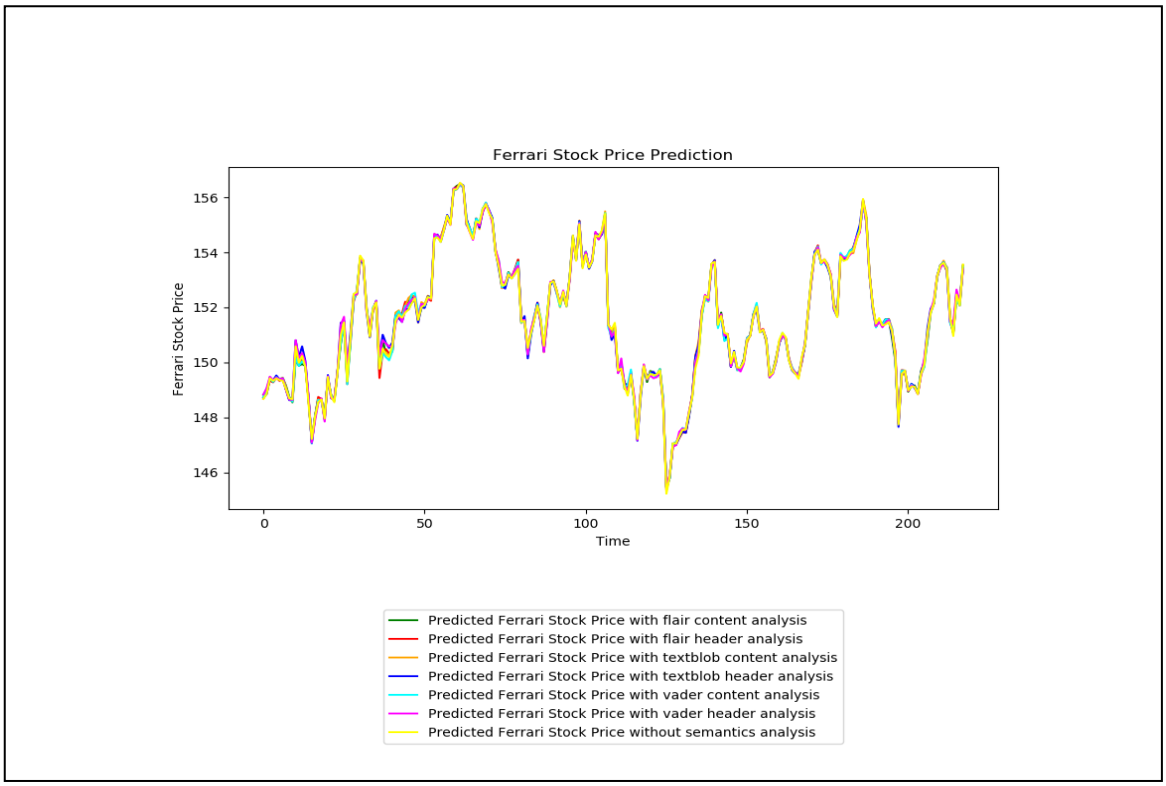


Figure 74: Stock Price Prediction Of Ferrari With Hourly Stock Price Data Using RandomForest Base Model

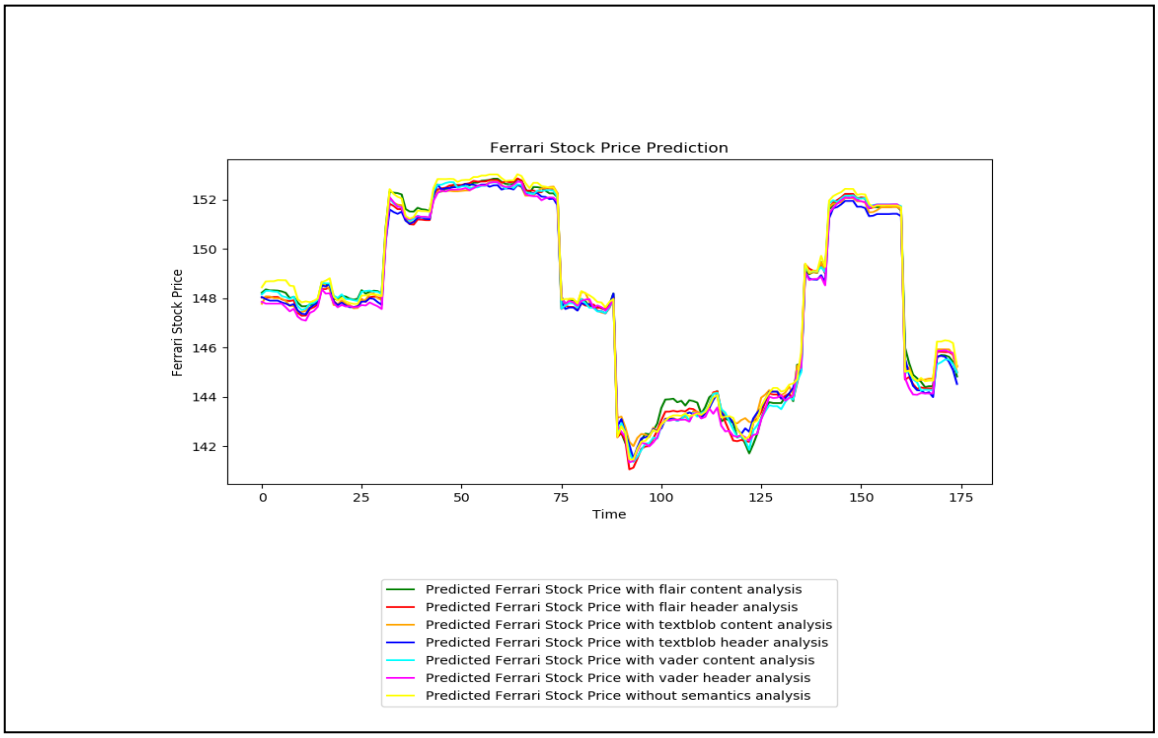


Figure 75: Stock Price Prediction Of Ferrari With Hourly Stock Price Data Using RandomForest Feature Model

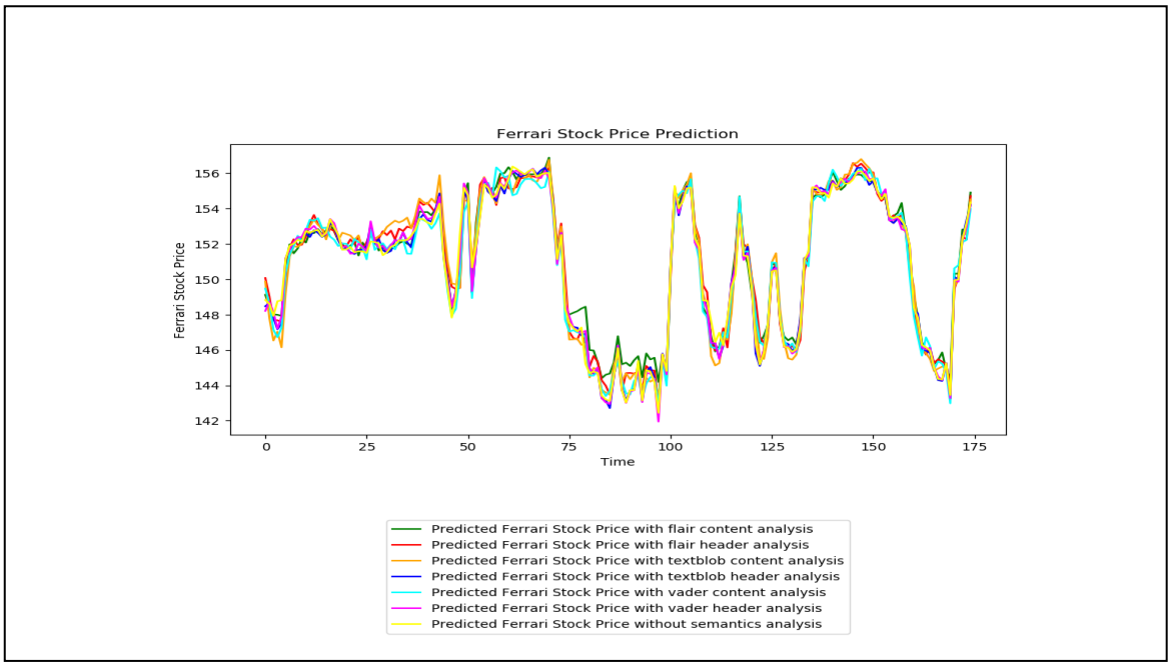


Figure 76: Stock Price Prediction Of Ferrari With Hourly Stock Price Data Using XGBoost

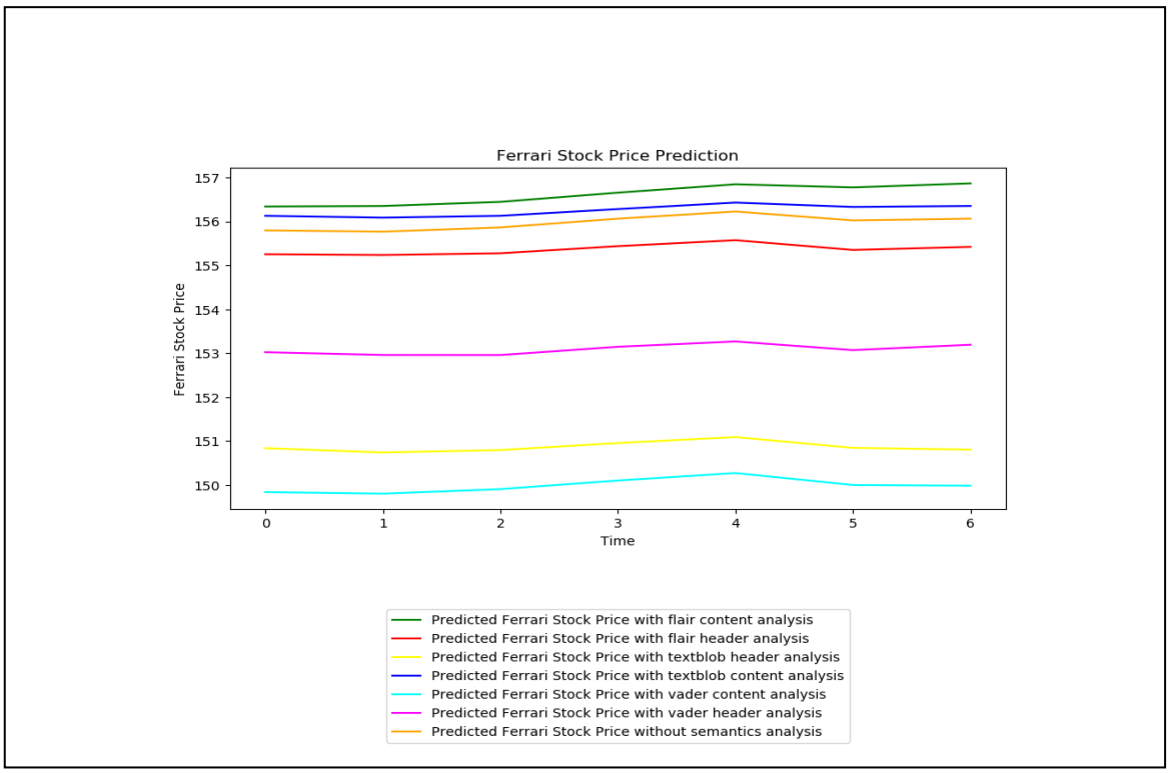


Figure 77: Stock Price Prediction Of Ferrari With Daily Stock Price Data Using LSTM

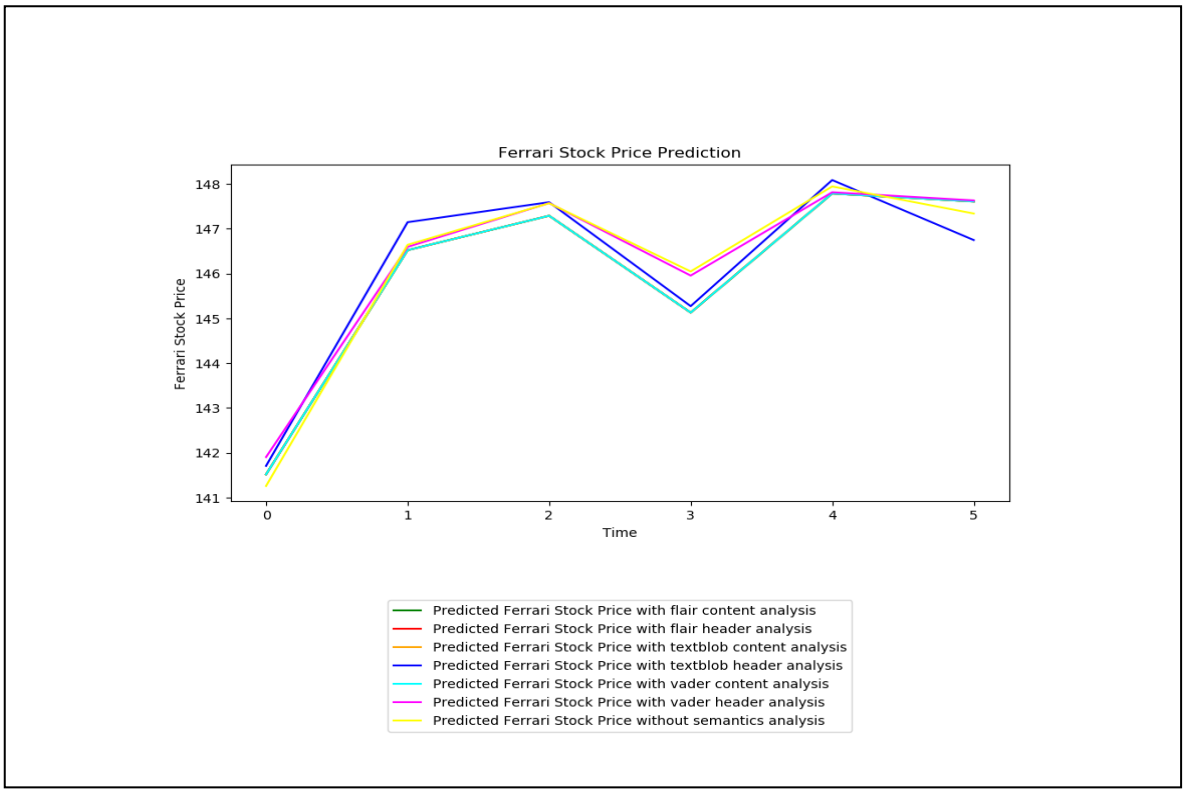


Figure 78: Stock Price Prediction Of Ferrari With Daily Stock Price Data Using RandomForest Base Model

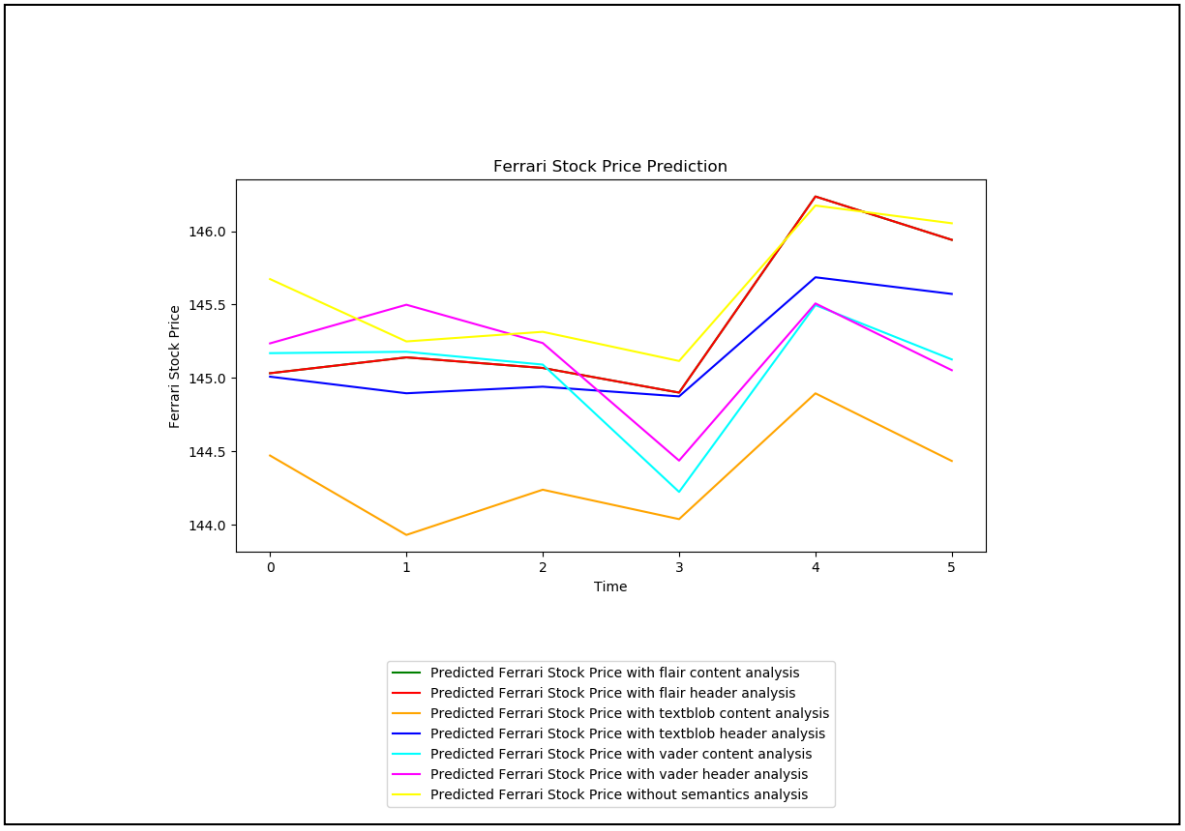


Figure 79: Stock Price Prediction Of Ferrari With Daily Stock Price Data Using RandomForest Feature Model

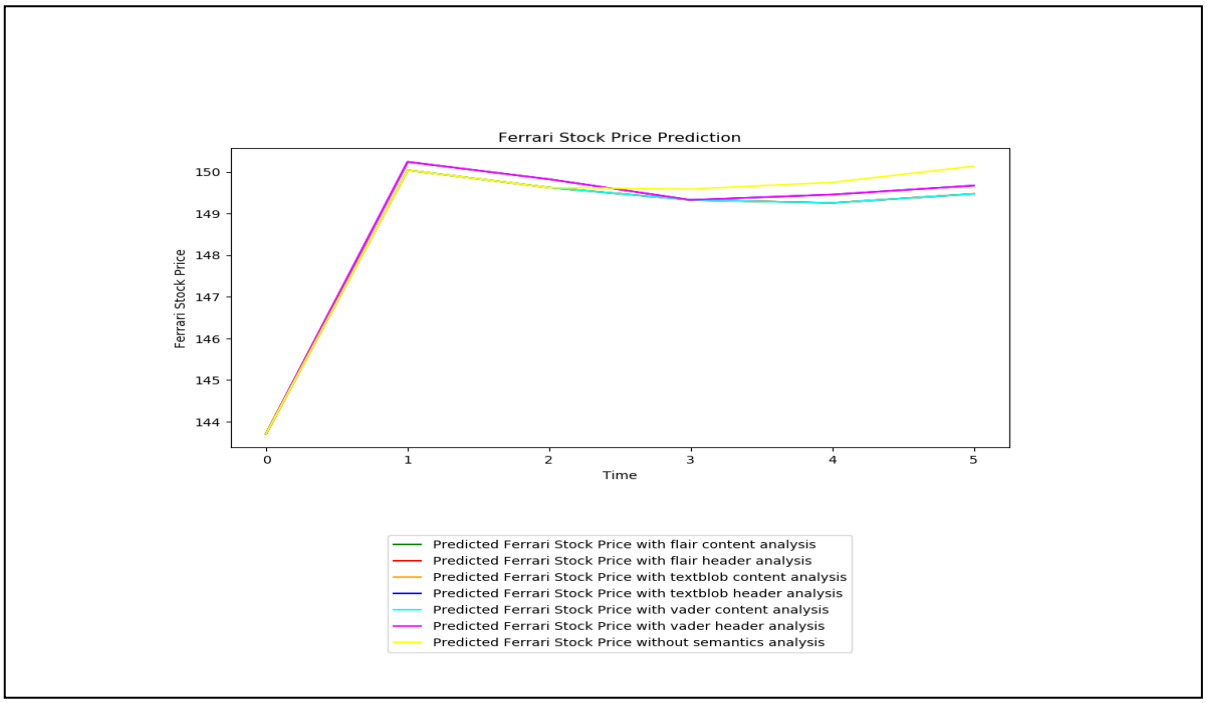


Figure 80: Stock Price Prediction Of Ferrari With Daily Stock Price Data Using XGBoost

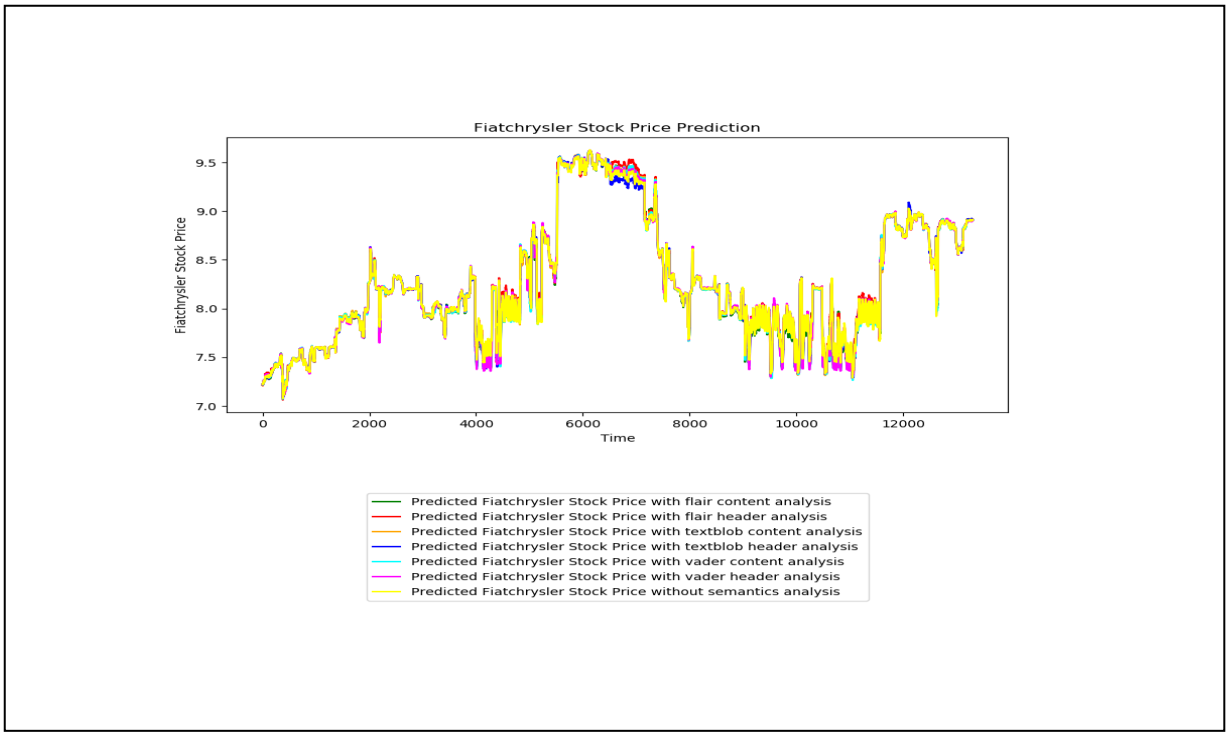


Figure 81: Stock Price Prediction Of Fiat Chrysler With Minutely Stock Price Data Using Random-Forest Feature Model

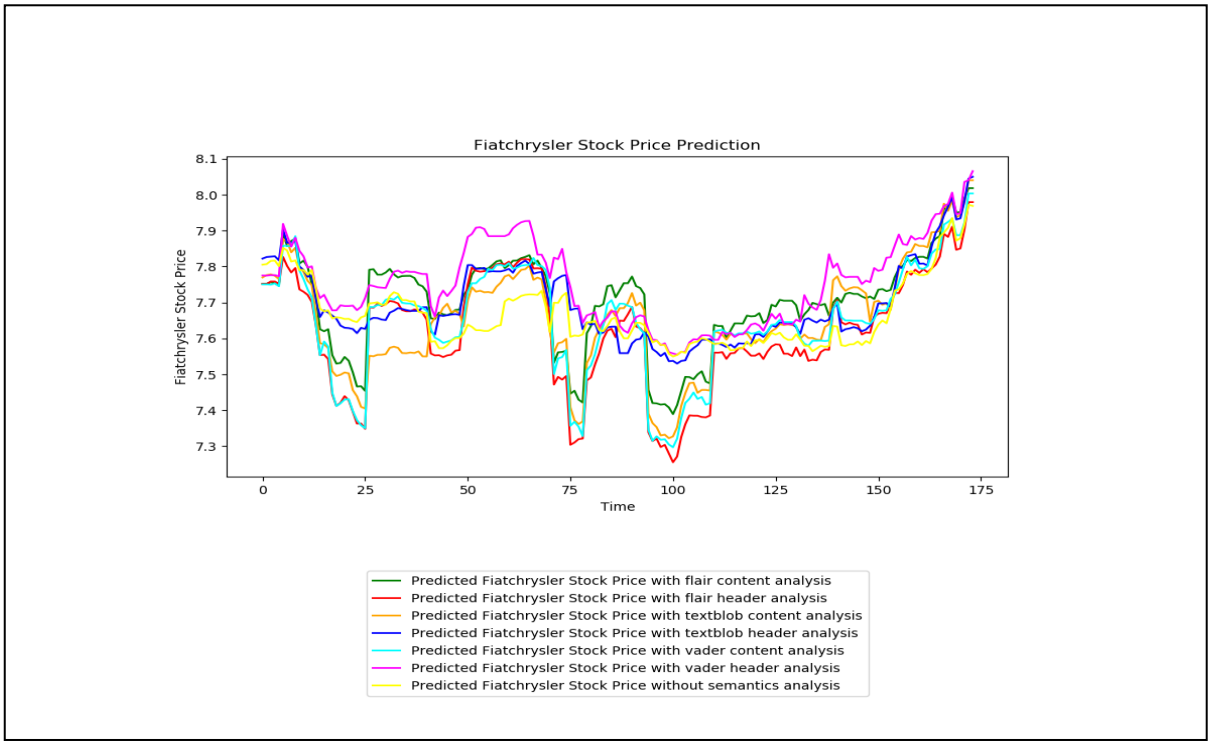


Figure 82: Stock Price Prediction Of Fiat Chrysler With Hourly Stock Price Data Using Random-Forest Feature Model

Figure 83: Stock Price Prediction Of Fiat Chrysler With Hourly Stock Price Data Using XGBoost

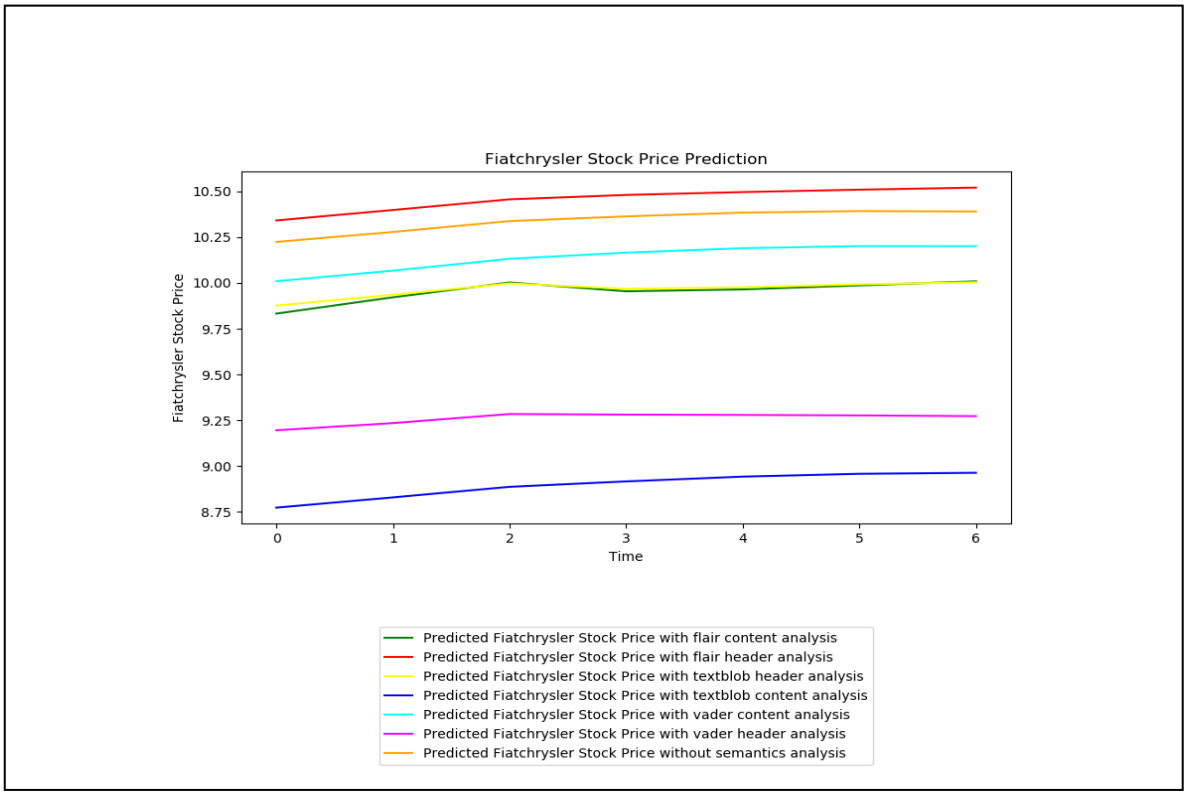


Figure 84: Stock Price Prediction Of Fiat Chrysler With Daily Stock Price Data Using LSTM

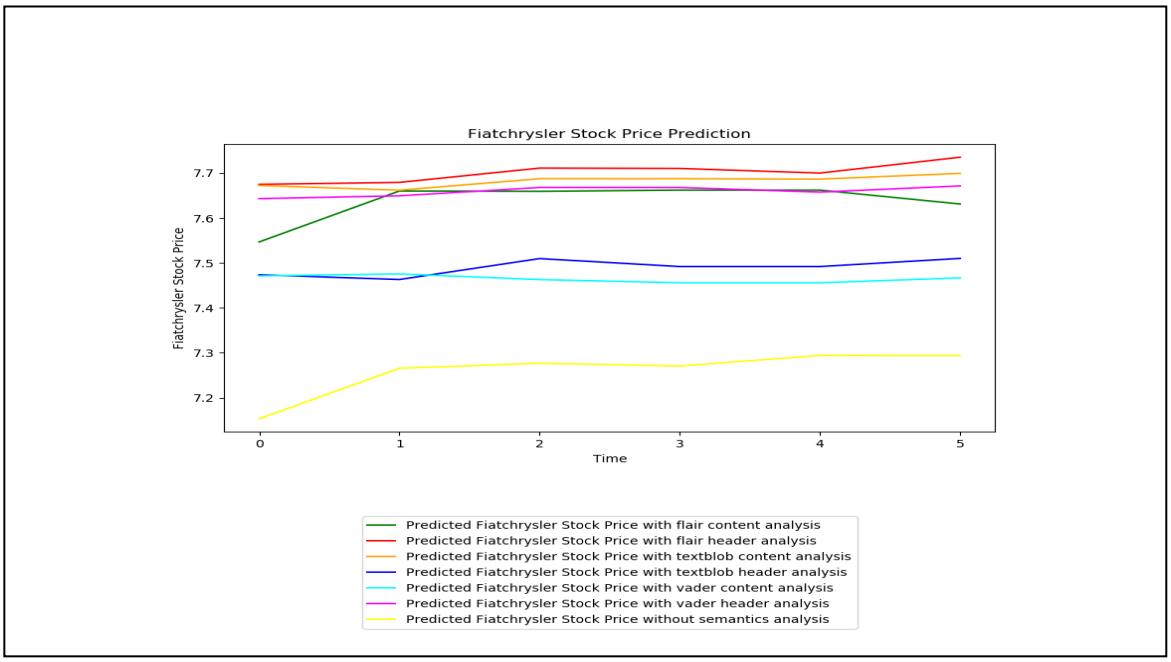


Figure 85: Stock Price Prediction Of Fiat Chrysler With Daily Stock Price Data Using Random-Forest Feature Model

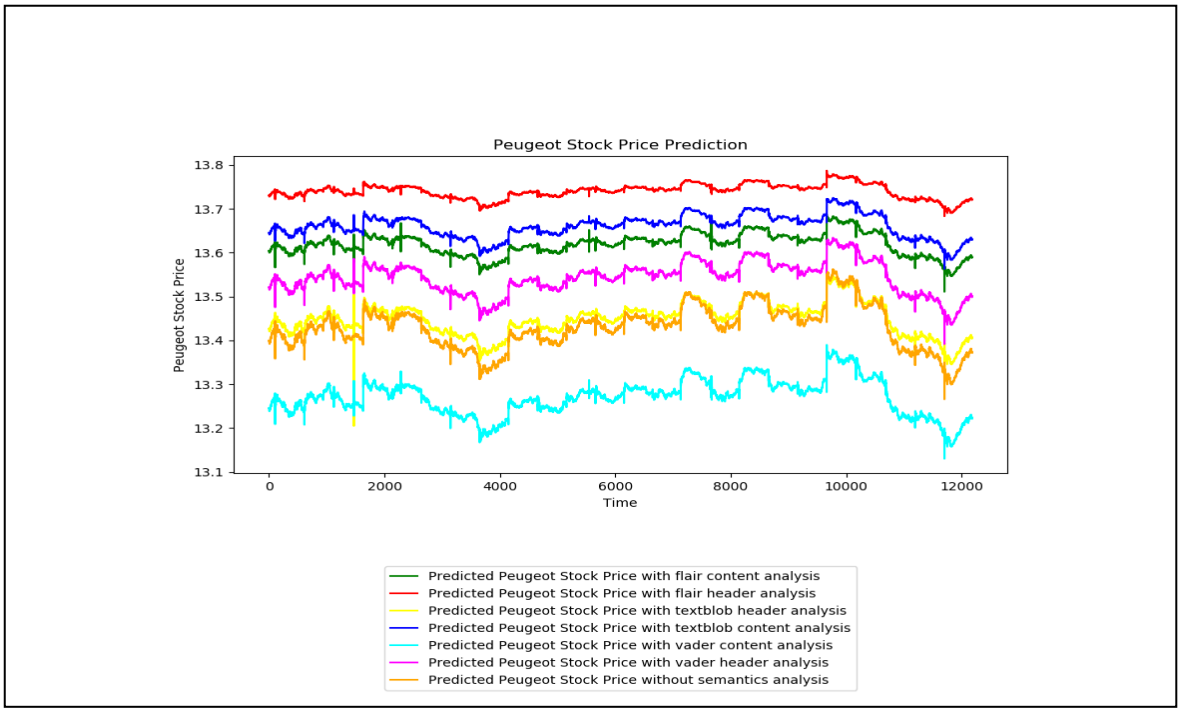


Figure 86: Stock Price Prediction Of Peugeot With Minutely Stock Price Data Using LSTM

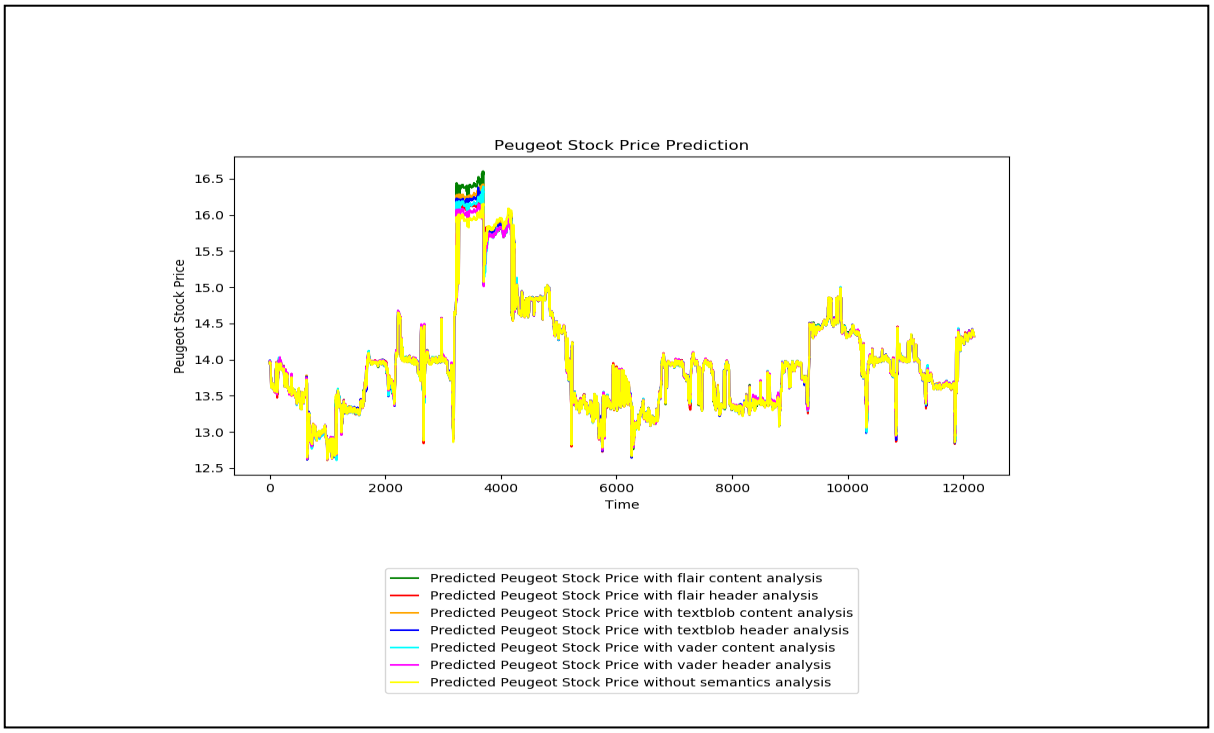


Figure 87: Stock Price Prediction Of Peugeot With Minutely Stock Price Data Using Random-Forest Base Model

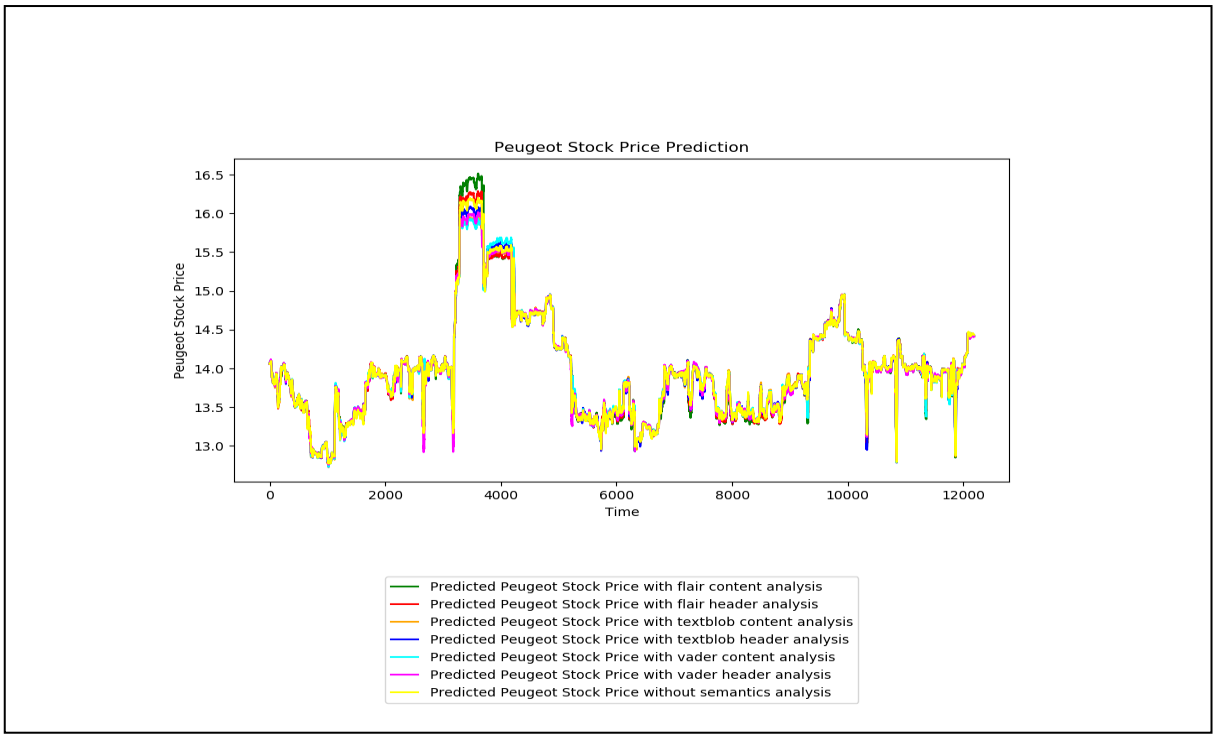


Figure 88: Stock Price Prediction Of Peugeot With Minutely Stock Price Data Using Random-Forest Feature Model

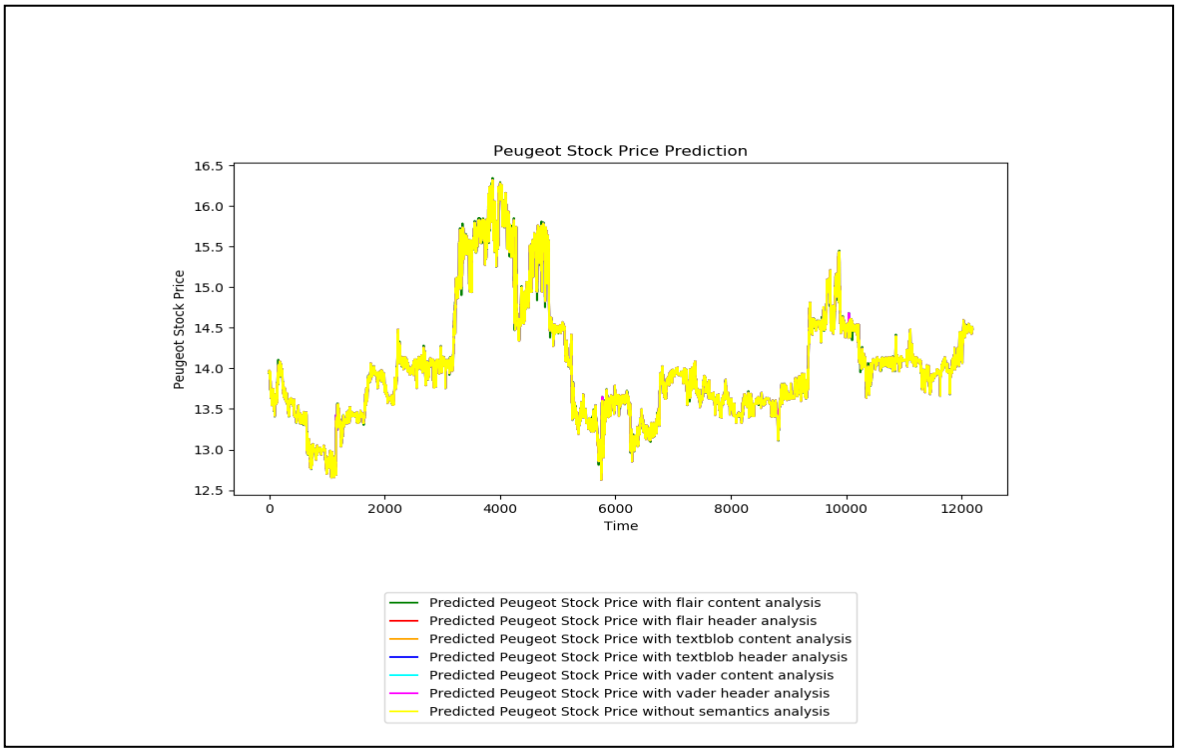


Figure 89: Stock Price Prediction Of Peugeot With Minutely Stock Price Data Using XGBoost

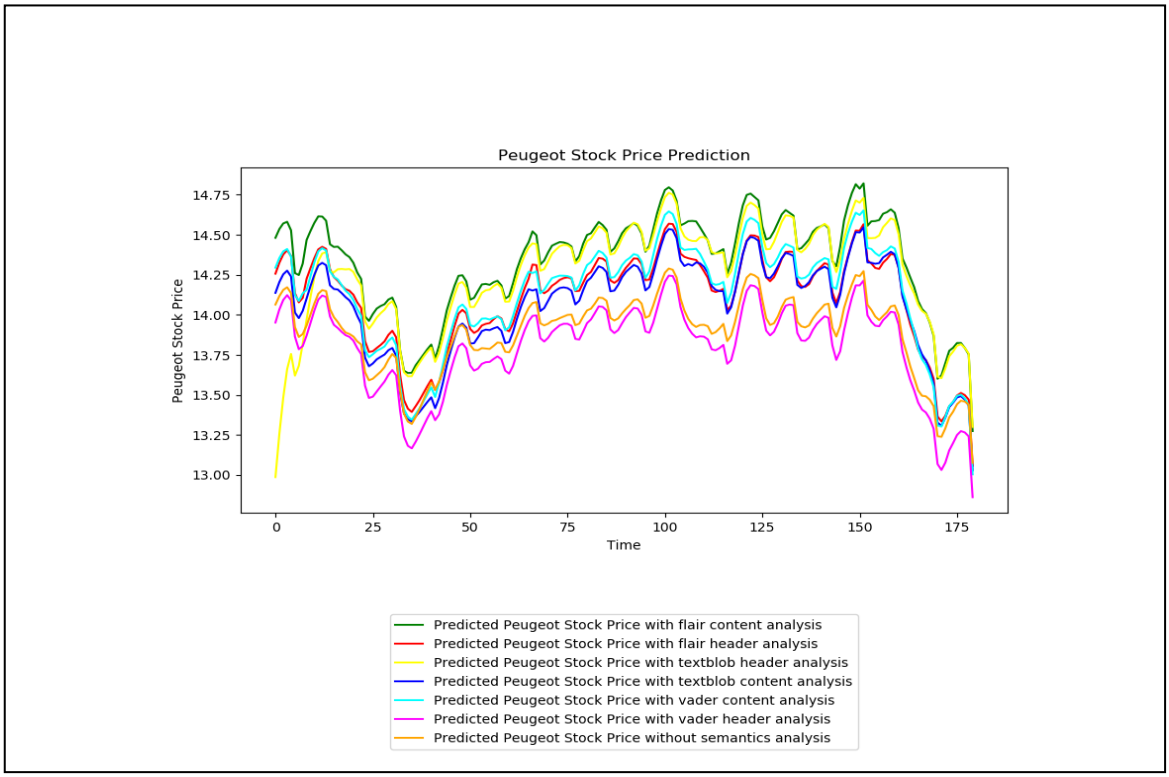


Figure 90: Stock Price Prediction Of Peugeot With Hourly Stock Price Data Using LSTM

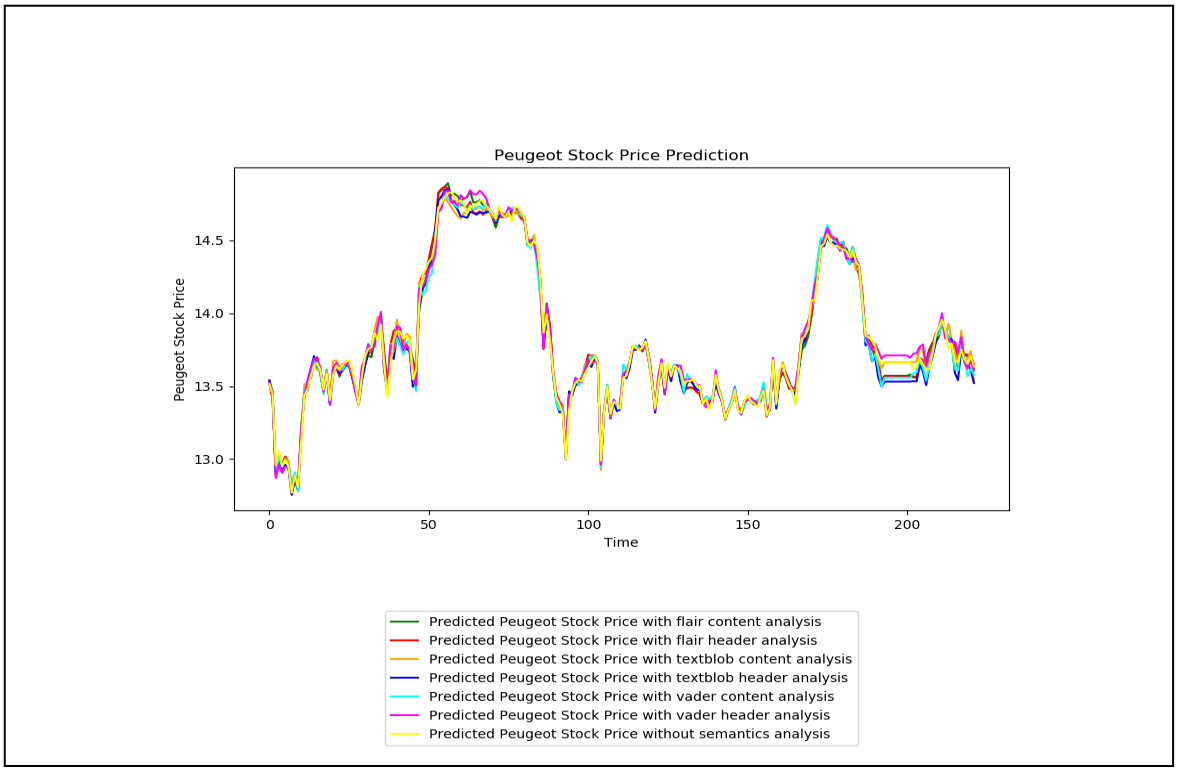


Figure 91: Stock Price Prediction Of Peugeot With Hourly Stock Price Data Using RandomForest Base Model

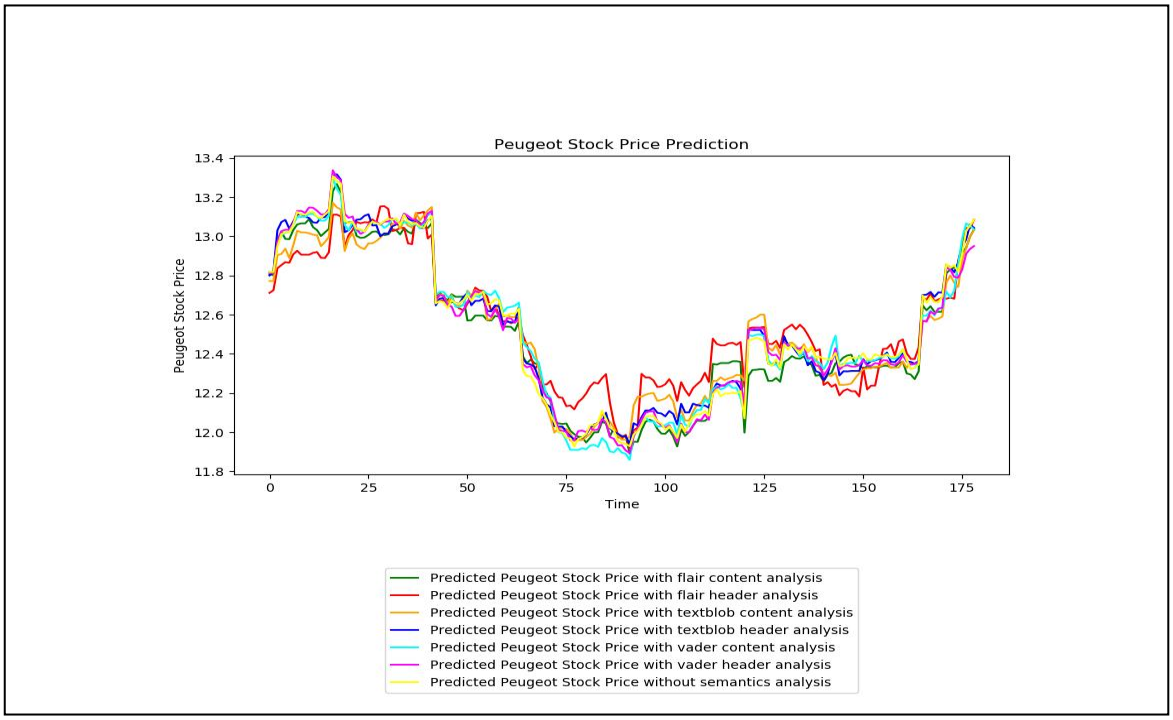


Figure 92: Stock Price Prediction Of Peugeot With Hourly Stock Price Data Using RandomForest Feature Model

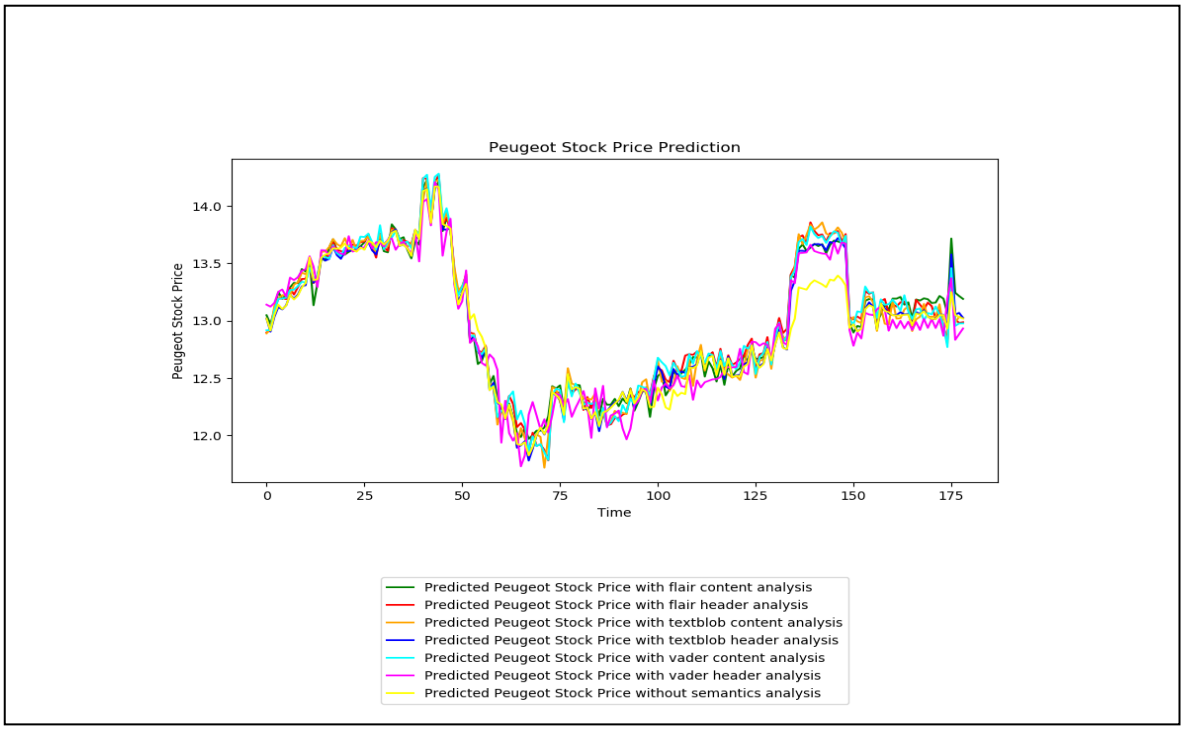


Figure 93: Stock Price Prediction Of Peugeot With Hourly Stock Price Data Using XGBoost

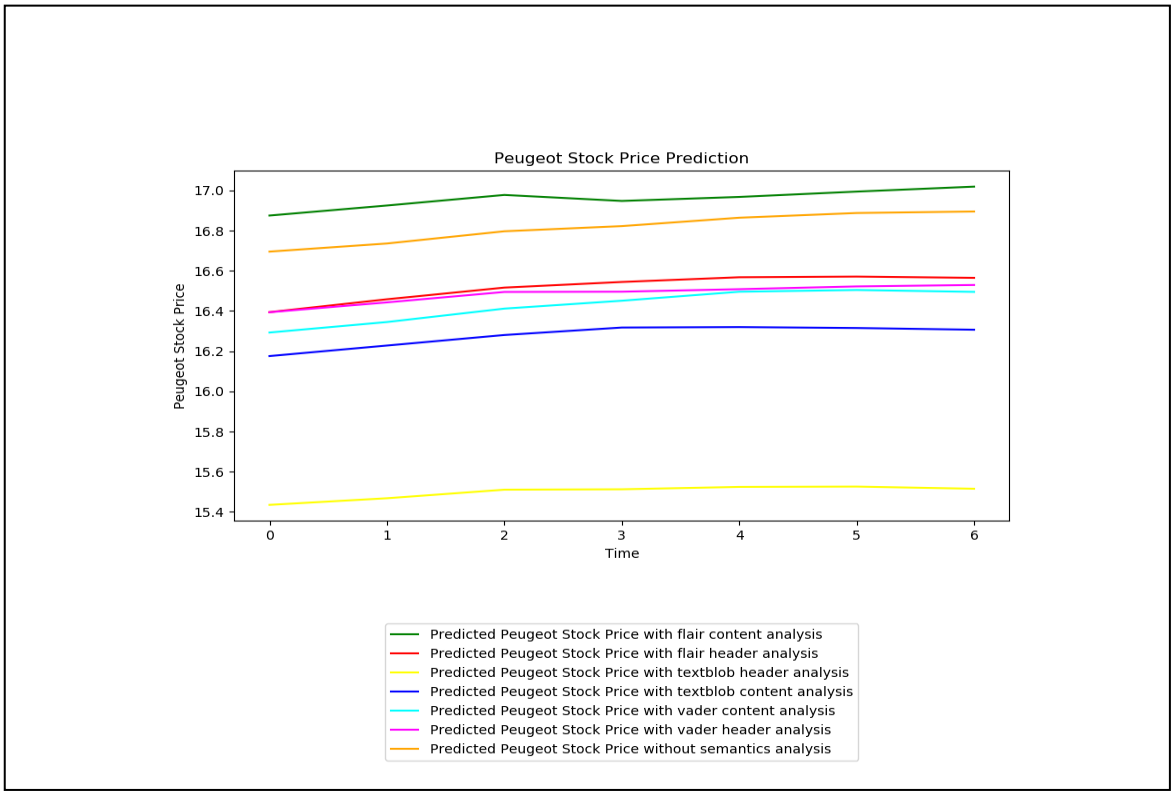


Figure 94: Stock Price Prediction Of Peugeot With Daily Stock Price Data Using LSTM

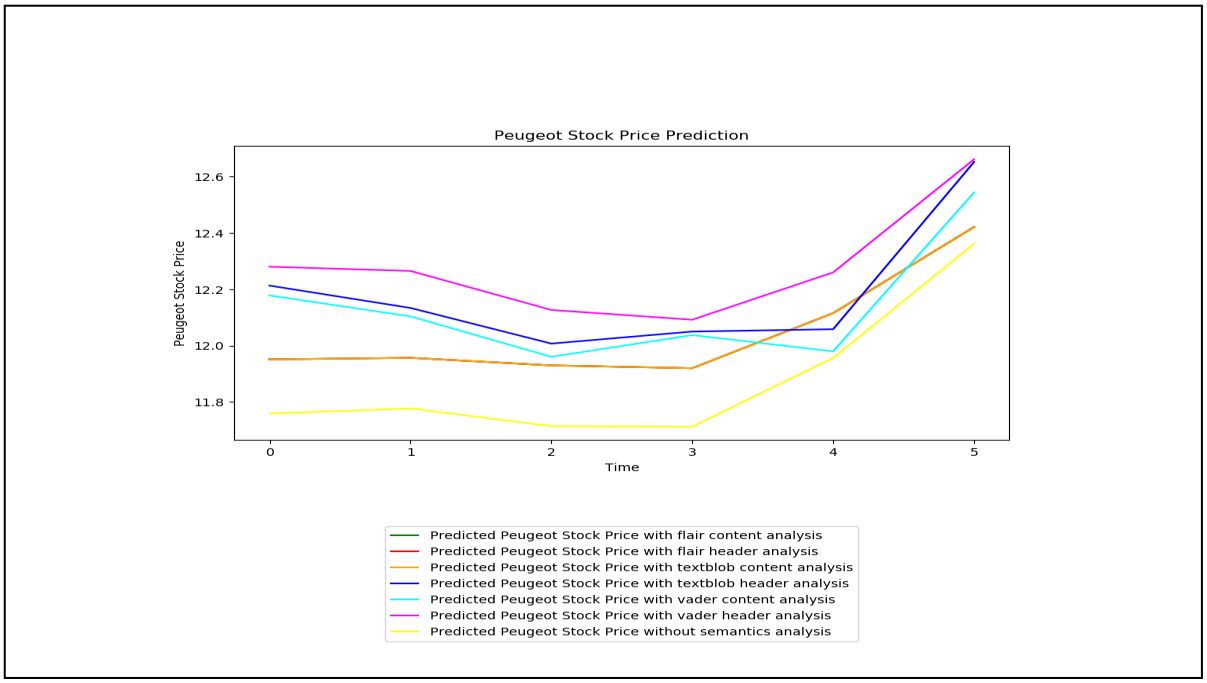


Figure 95: Stock Price Prediction Of Peugeot With Daily Stock Price Data Using RandomForest Base Model

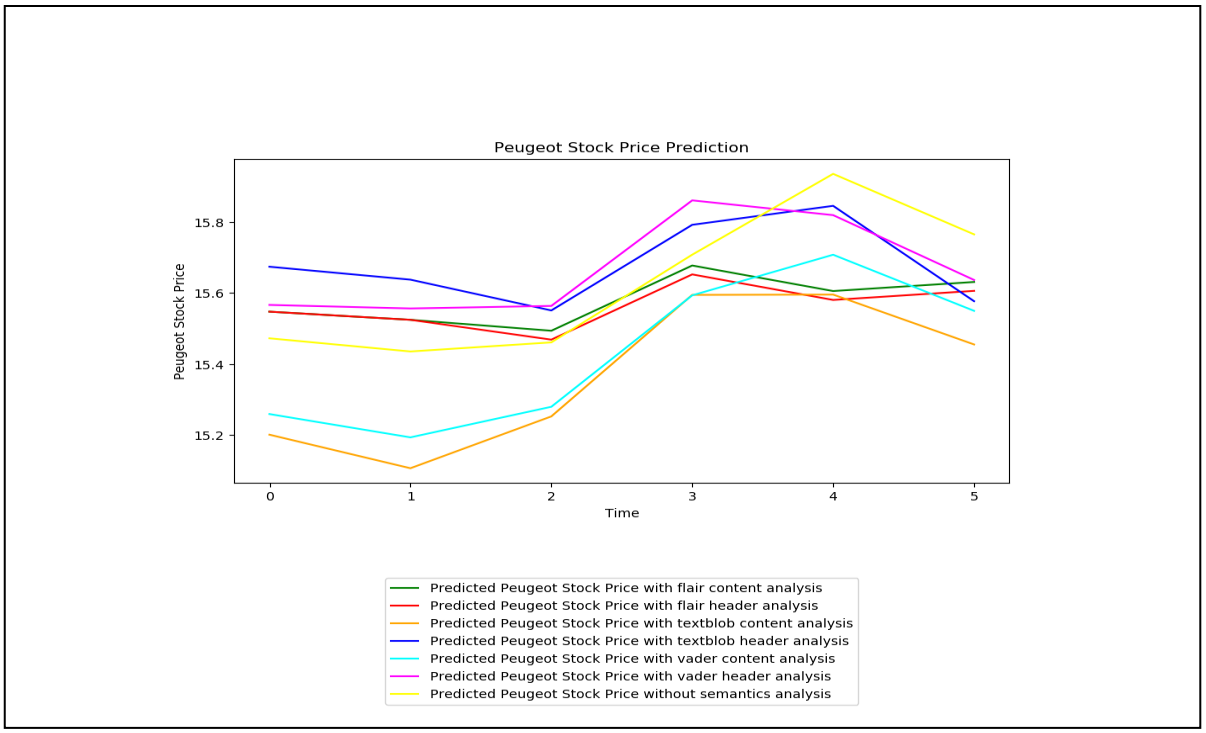


Figure 96: Stock Price Prediction Of Peugeot With Daily Stock Price Data Using RandomForest Feature Model

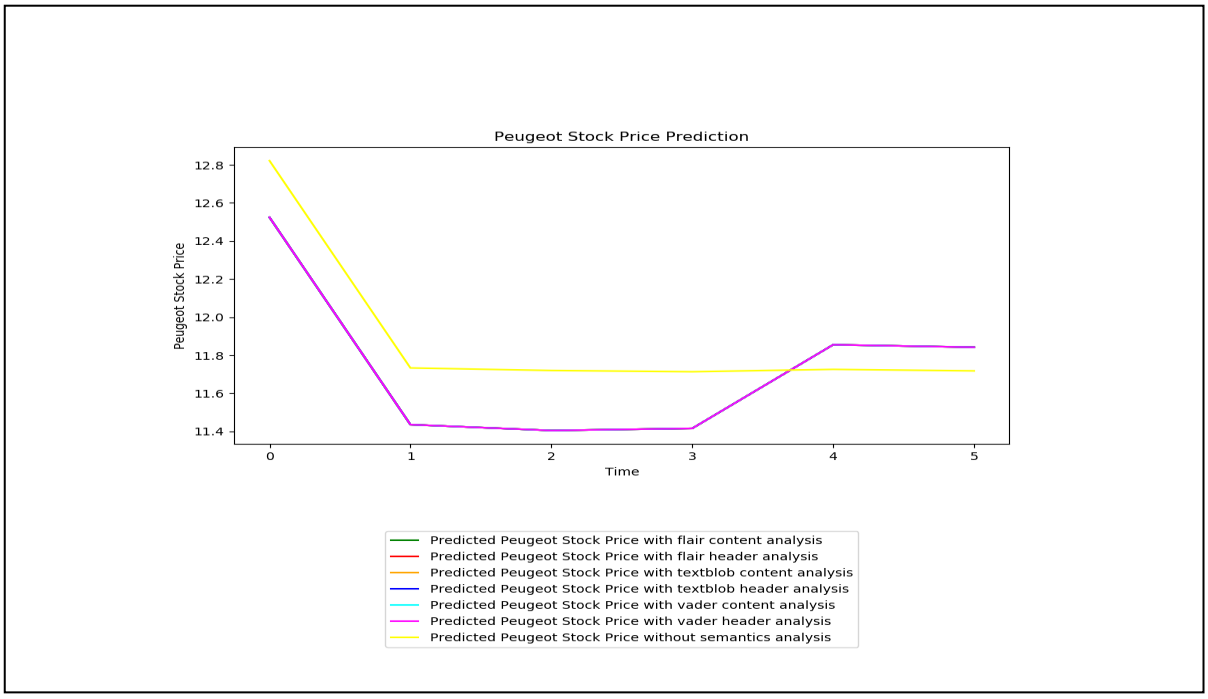


Figure 97: Stock Price Prediction Of Peugeot With Daily Stock Price Data Using XGBoost

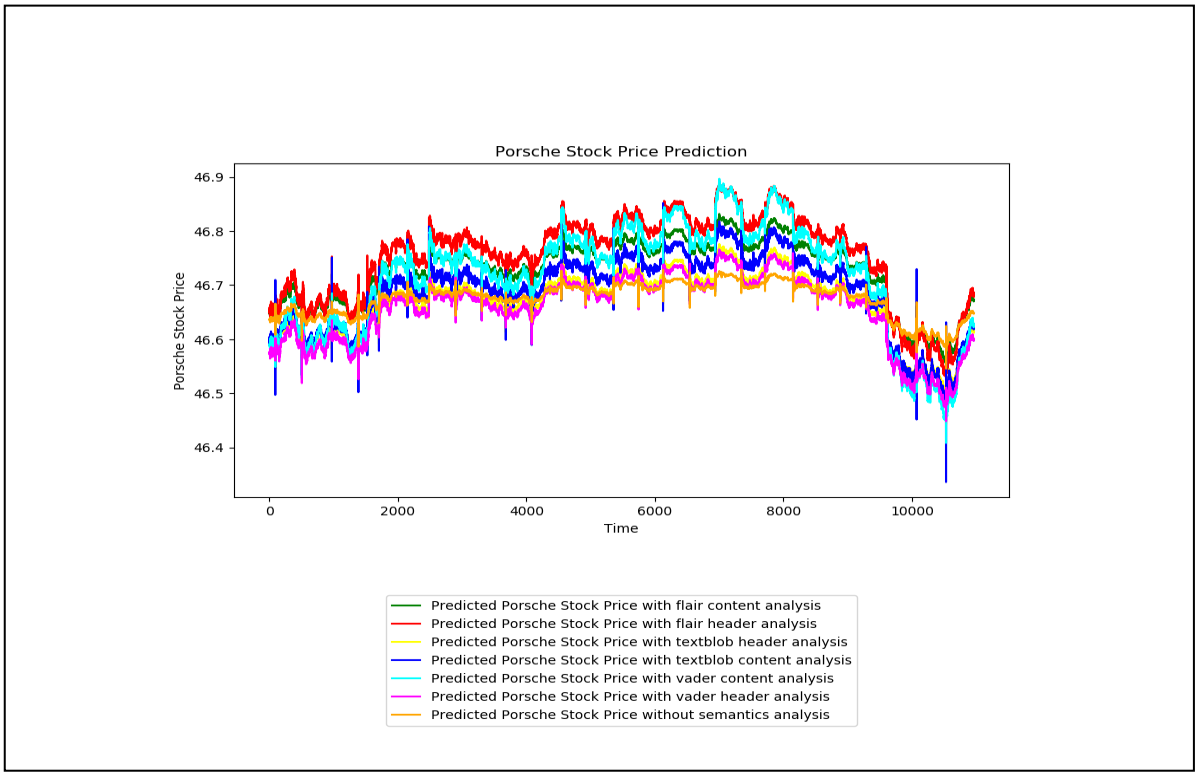


Figure 98: Stock Price Prediction Of Porsche With Minutely Stock Price Data Using LSTM

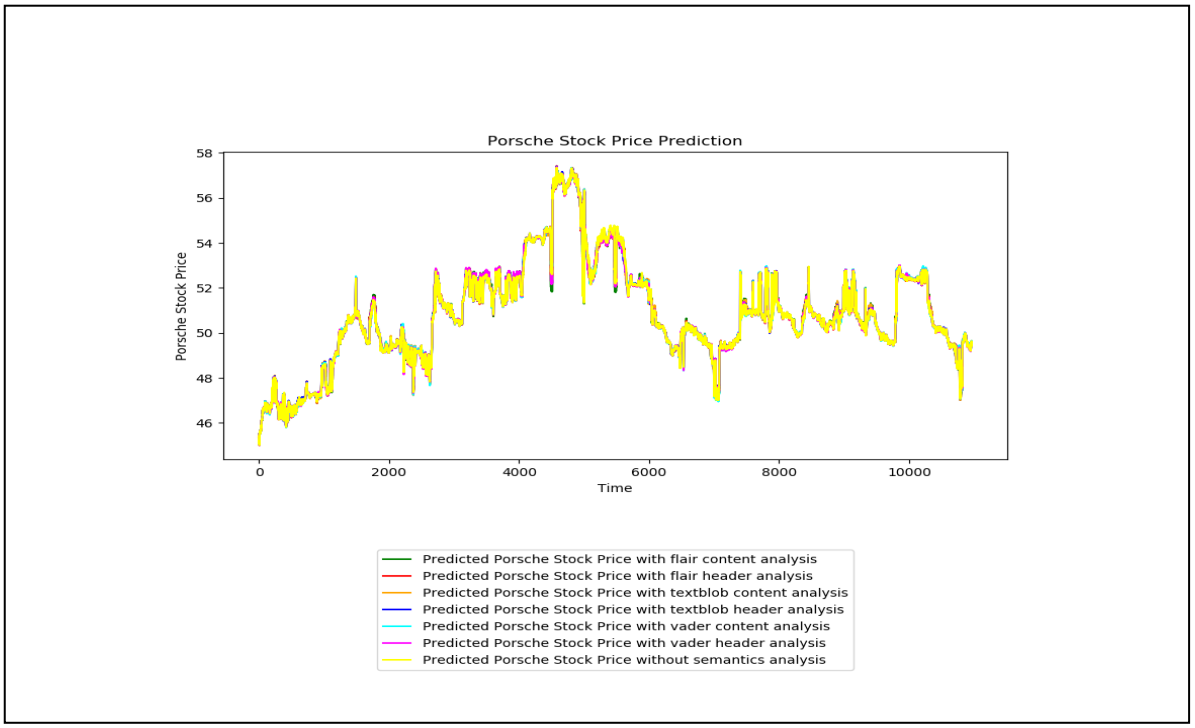


Figure 99: Stock Price Prediction Of Porsche With Minutely Stock Price Data Using Random-Forest Base Model

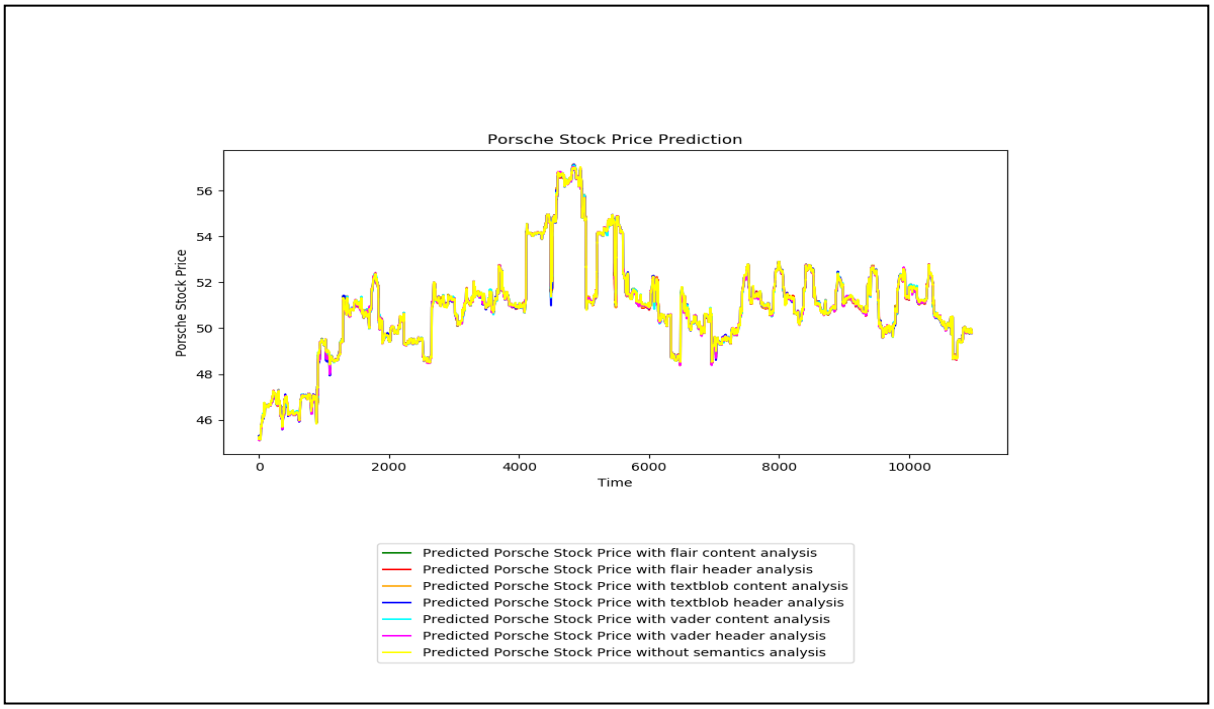


Figure 100: Stock Price Prediction Of Porsche With Minutely Stock Price Data Using Random-Forest Feature Model

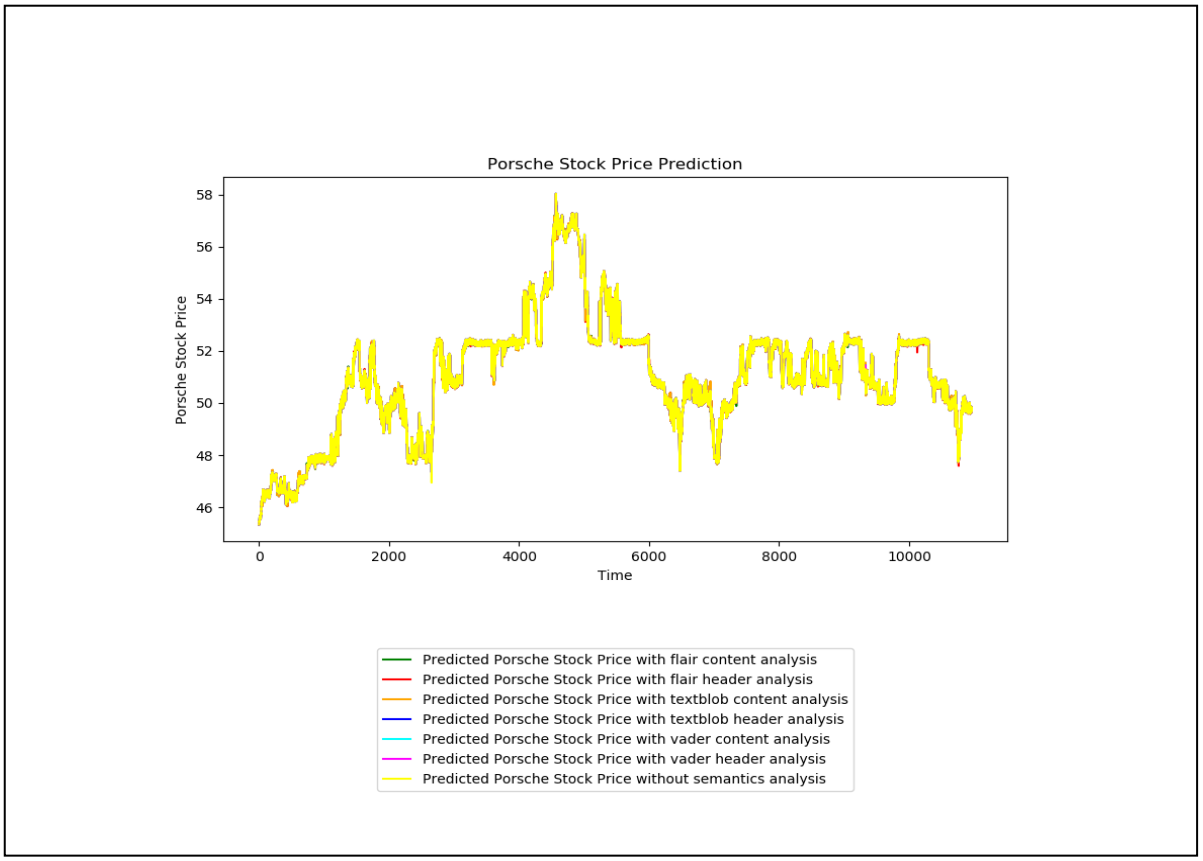


Figure 101: Stock Price Prediction Of Porsche With Minutely Stock Price Data Using XGBoost

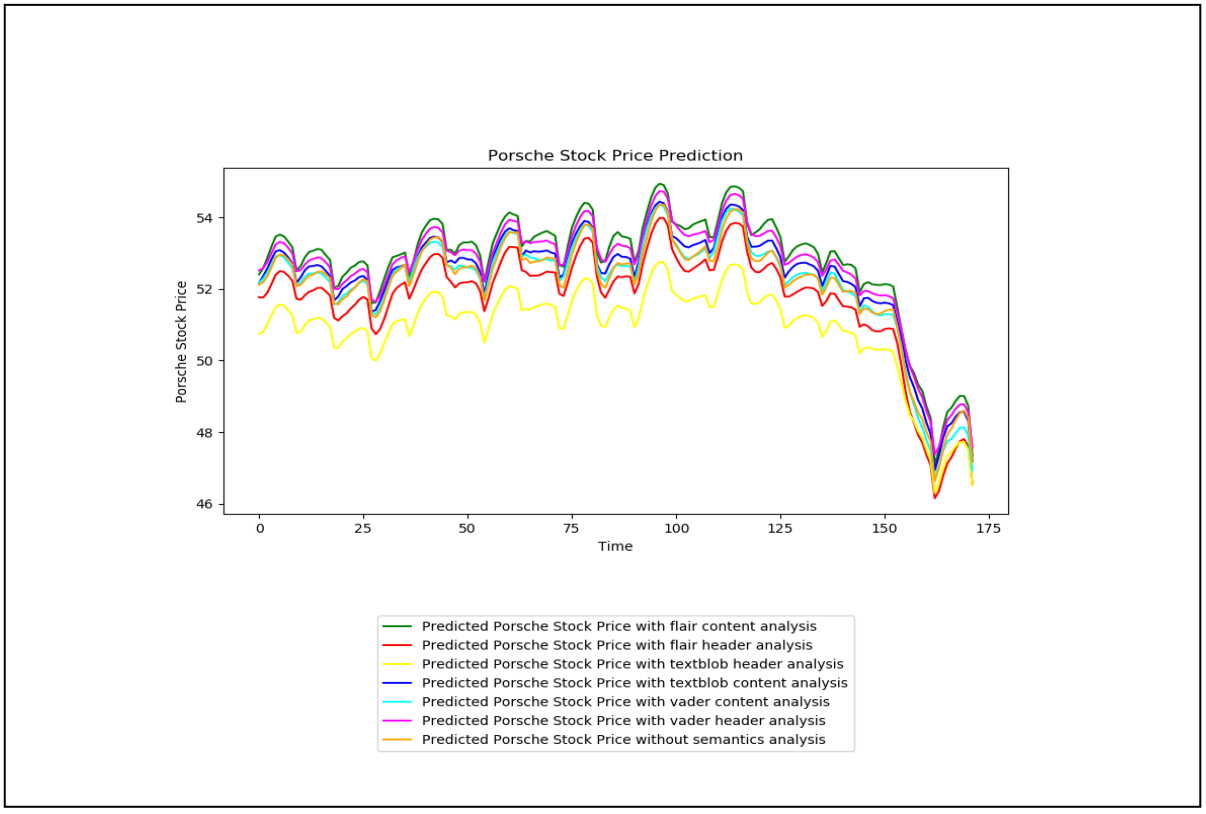


Figure 102: Stock Price Prediction Of Porsche With Hourly Stock Price Data Using LSTM

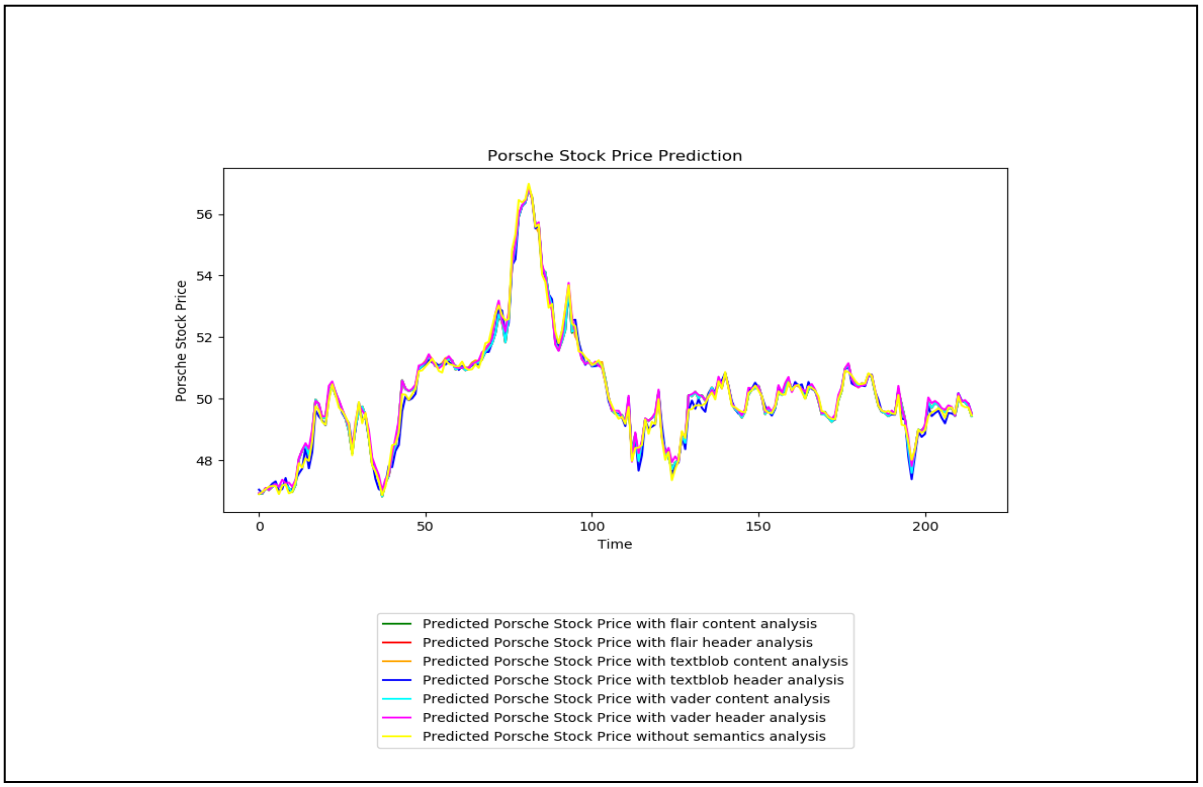


Figure 103: Stock Price Prediction Of Porsche With Hourly Stock Price Data Using RandomForest Base Model

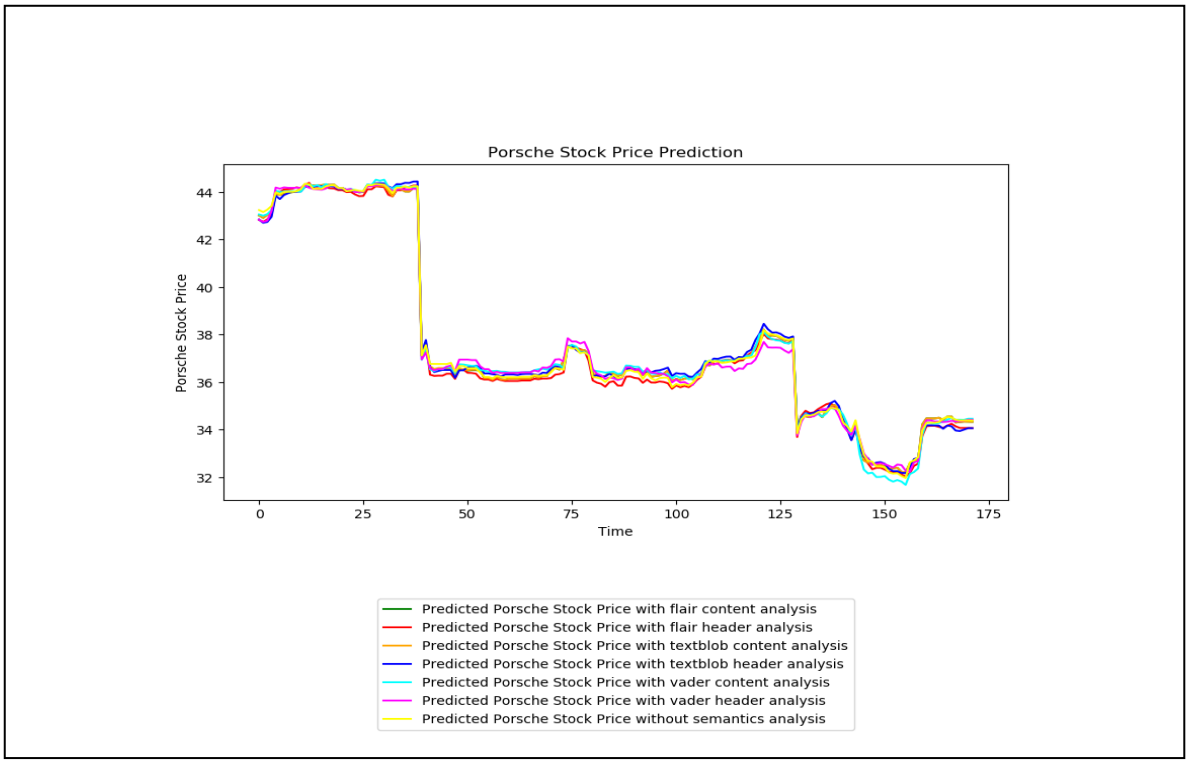


Figure 104: Stock Price Prediction Of Porsche With Hourly Stock Price Data Using RandomForest Feature Model

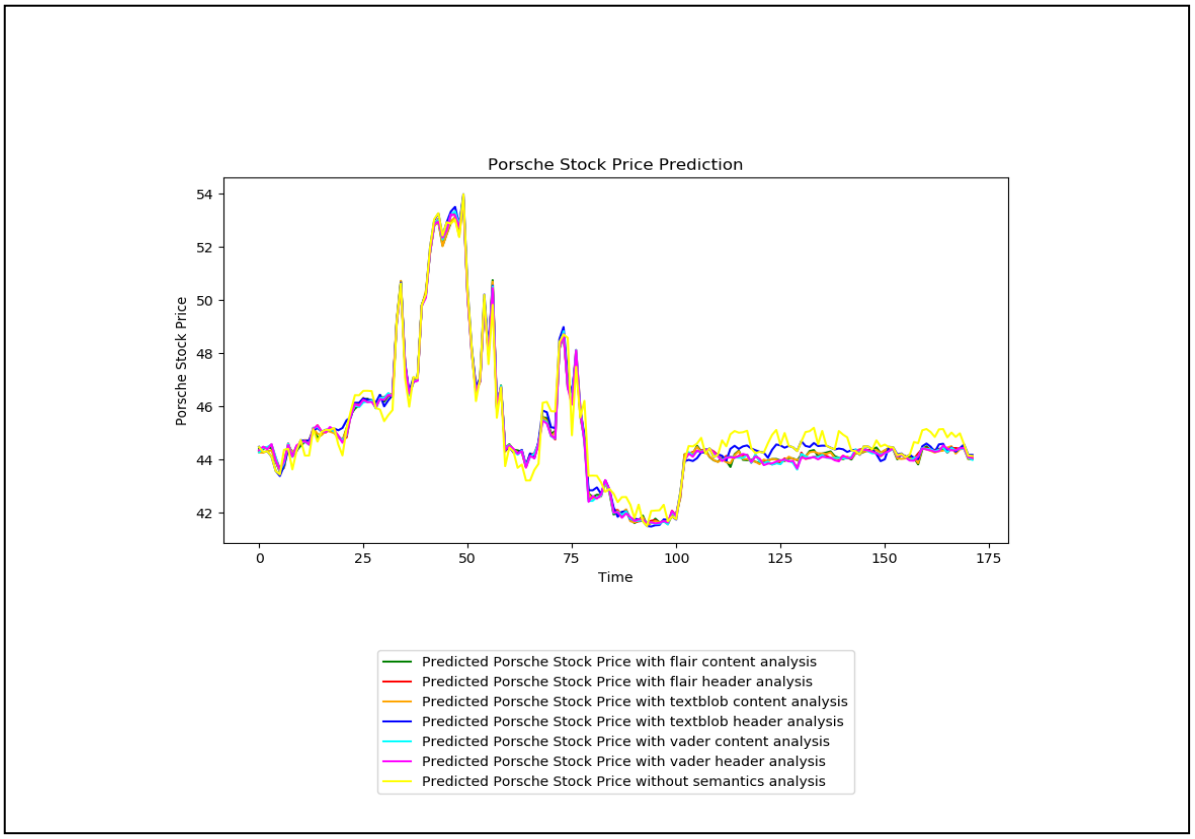


Figure 105: Stock Price Prediction Of Porsche With Hourly Stock Price Data Using XGBoost

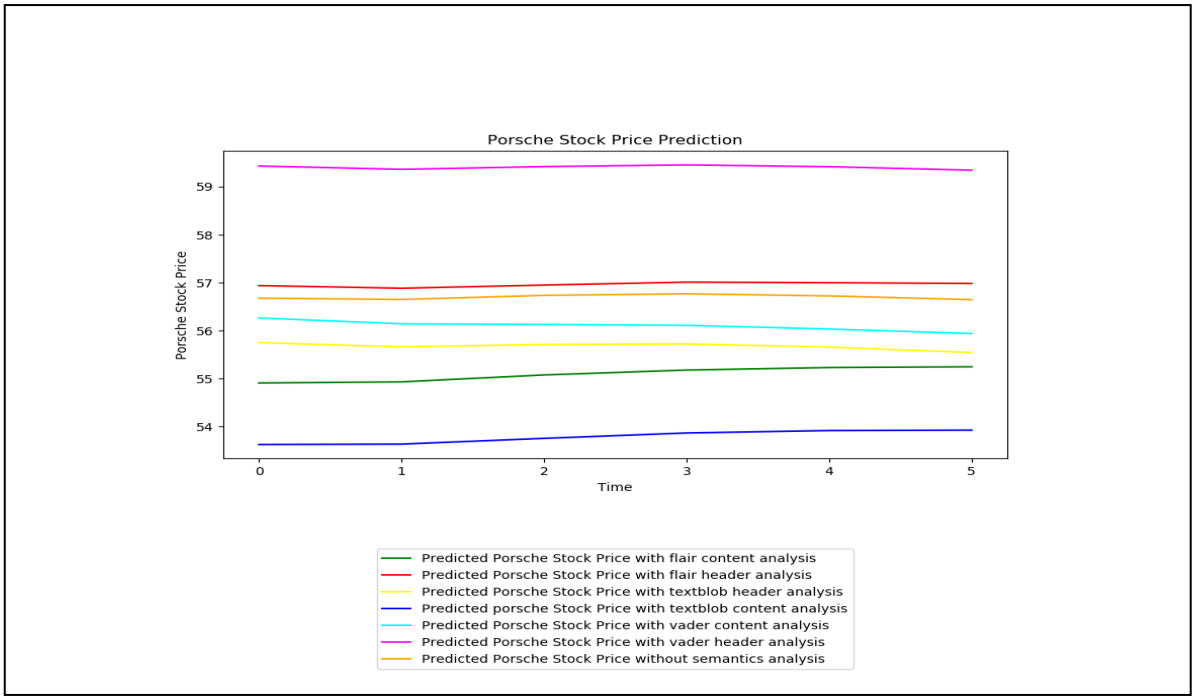


Figure 106: Stock Price Prediction Of Porsche With Daily Stock Price Data Using LSTM

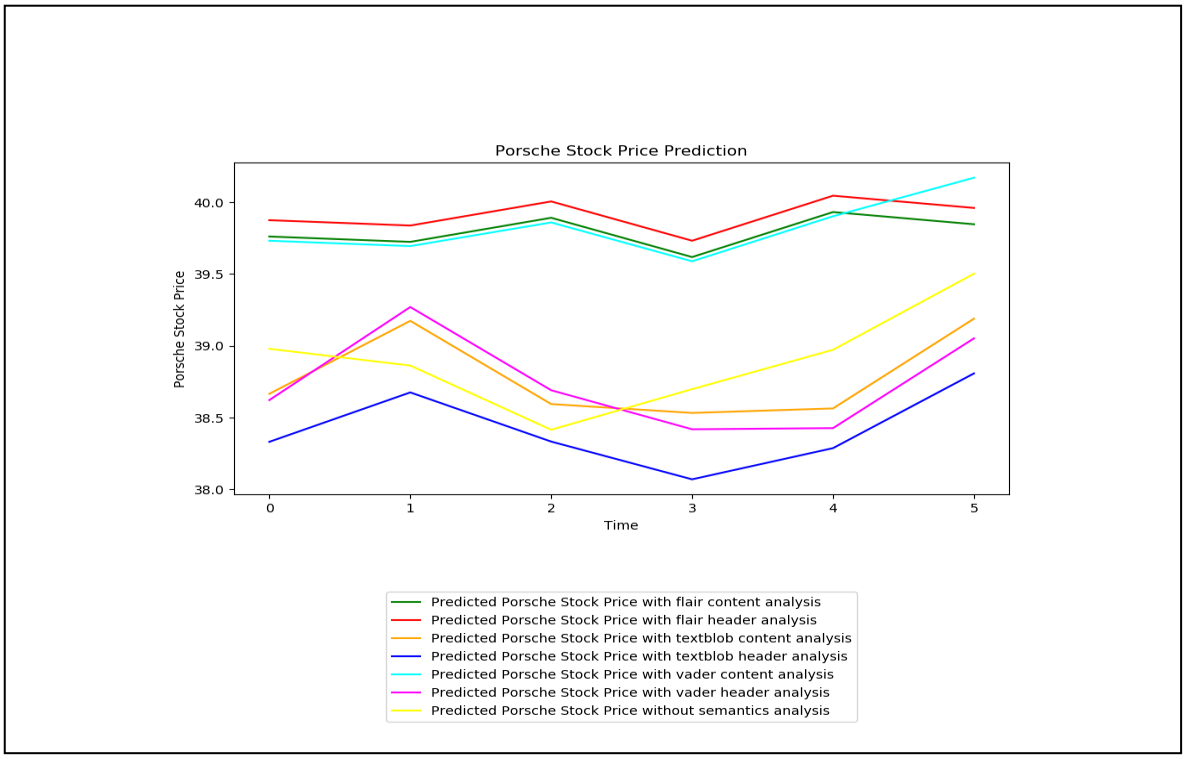


Figure 107: Stock Price Prediction Of Porsche With Daily Stock Price Data Using RandomForest Base Model

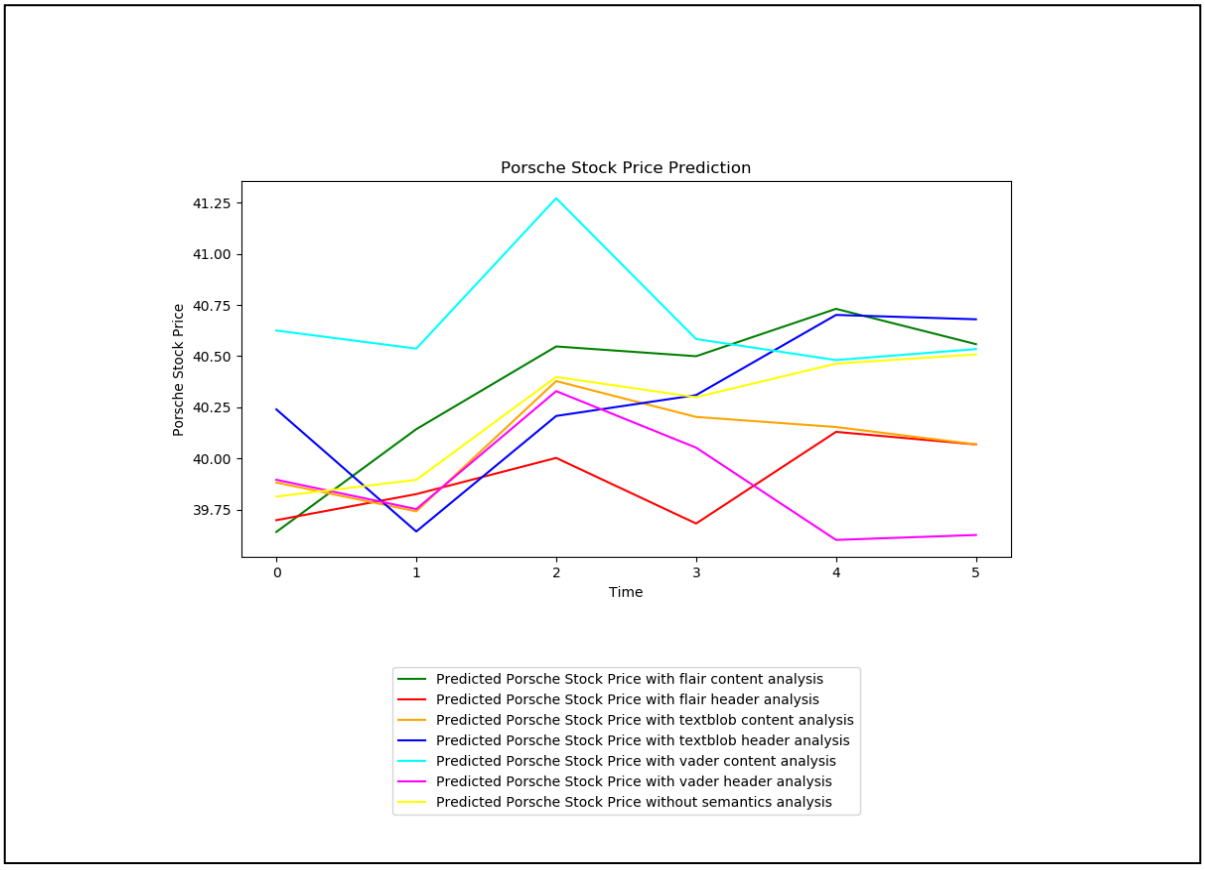


Figure 108: Stock Price Prediction Of Porsche With Daily Stock Price Data Using RandomForest Feature Model

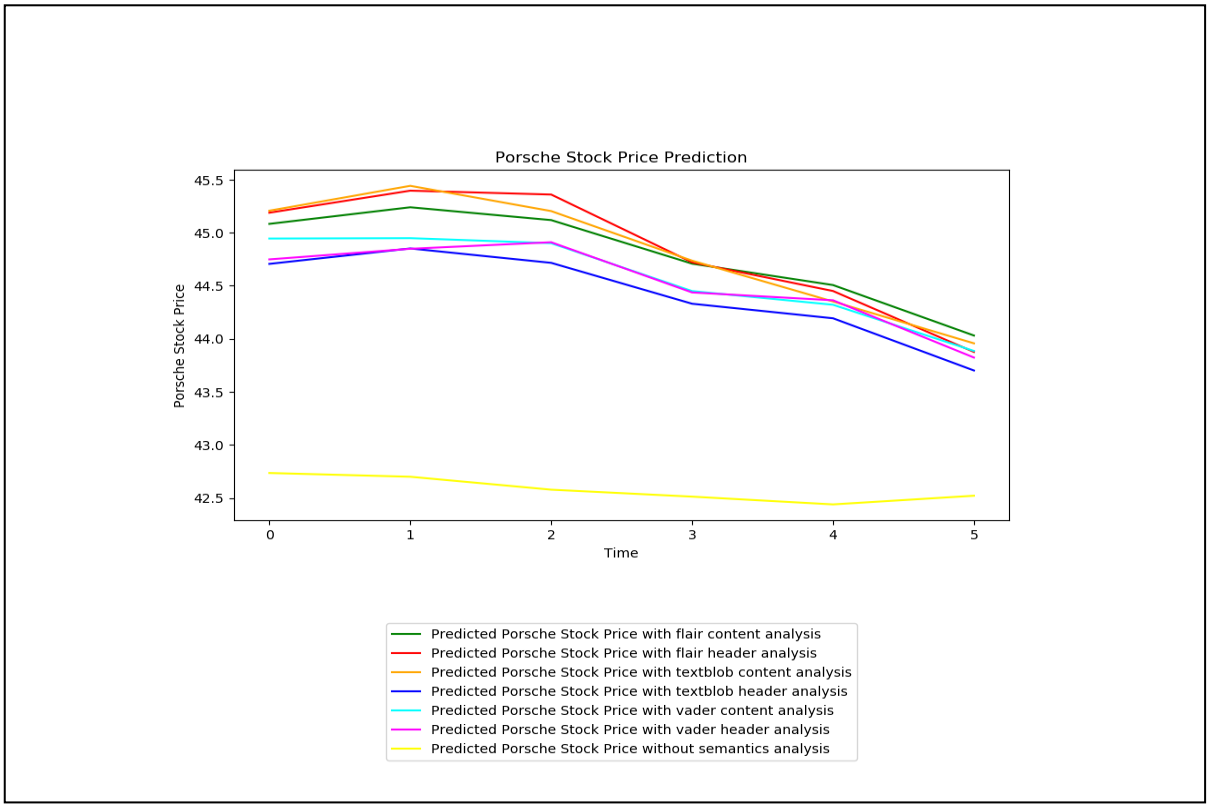


Figure 109: Stock Price Prediction Of Porsche With Daily Stock Price Data Using XGBoost

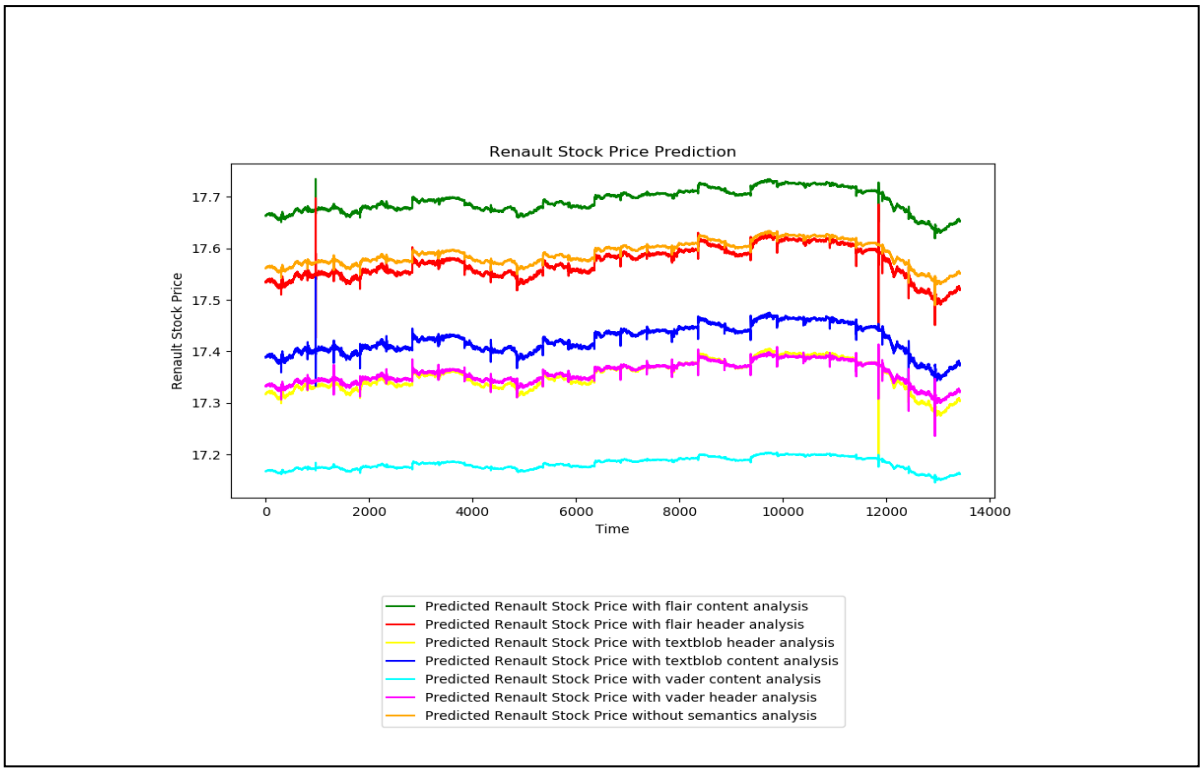


Figure 110: Stock Price Prediction Of Renault With Minutely Stock Price Data Using LSTM

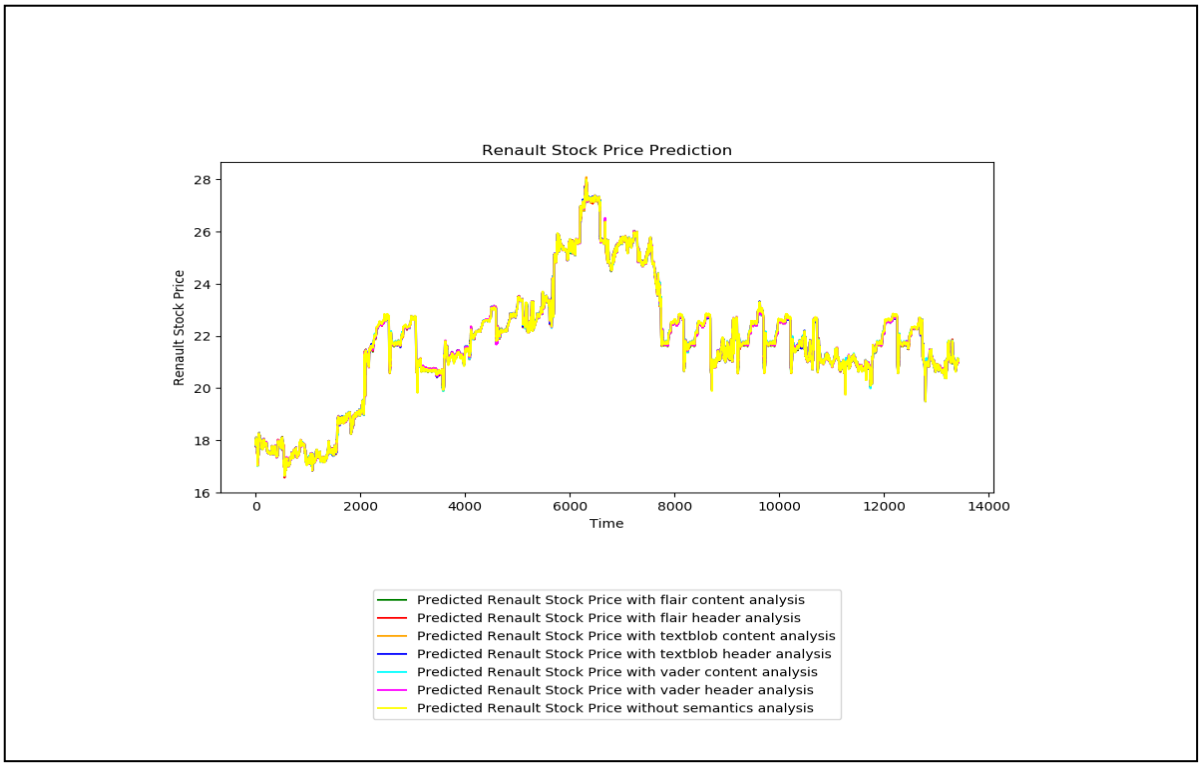


Figure 111: Stock Price Prediction Of Renault With Minutely Stock Price Data Using Random-Forest Base Model

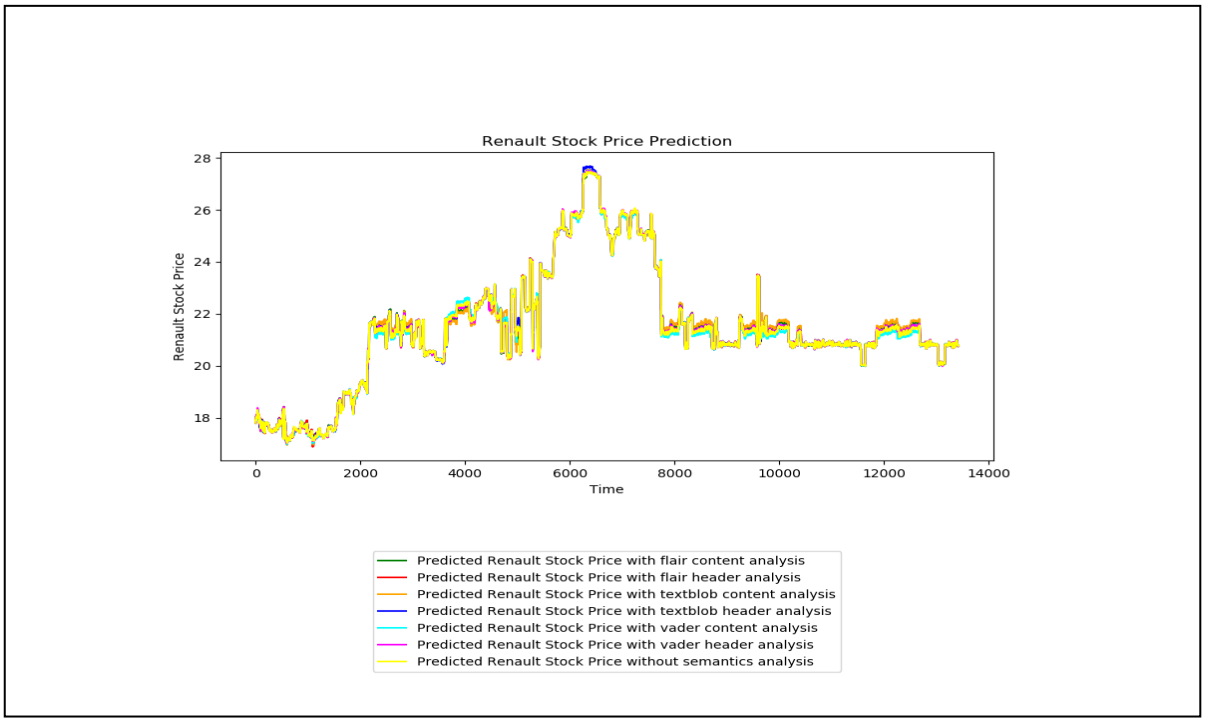


Figure 112: Stock Price Prediction Of Renault With Minutely Stock Price Data Using Random-Forest Feature Model

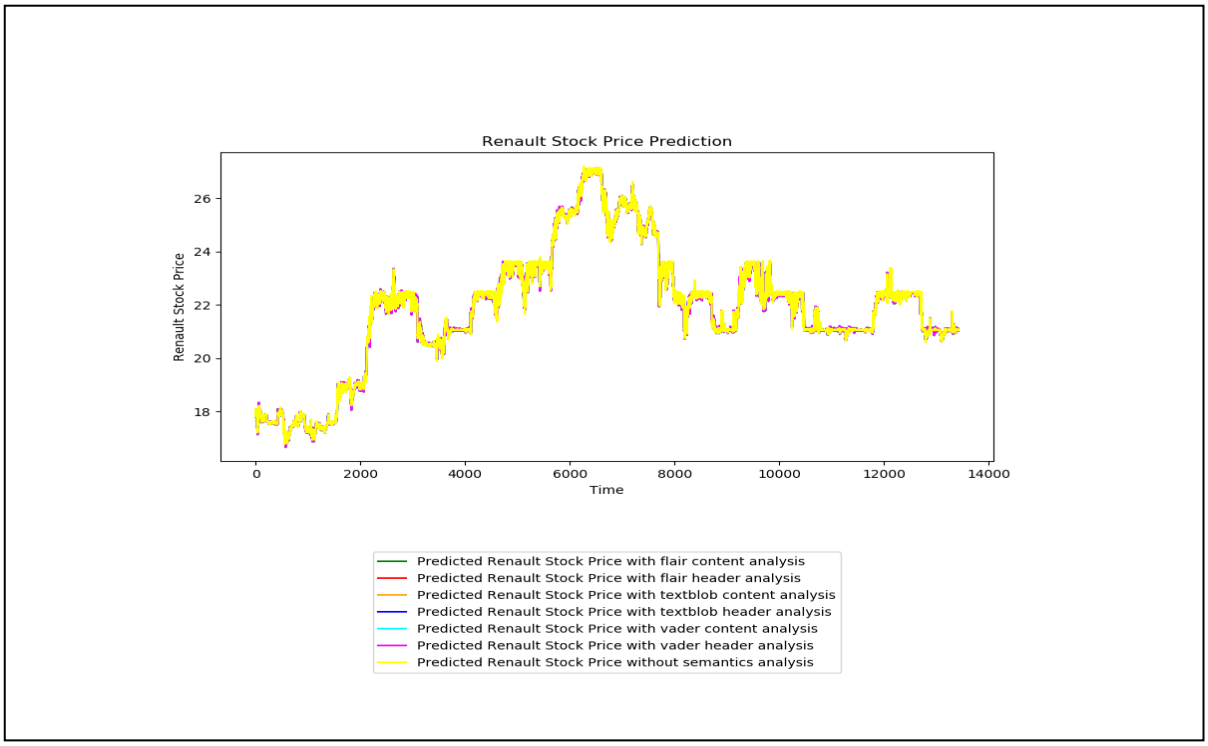


Figure 113: Stock Price Prediction Of Renault With Minutely Stock Price Data Using XGBoost

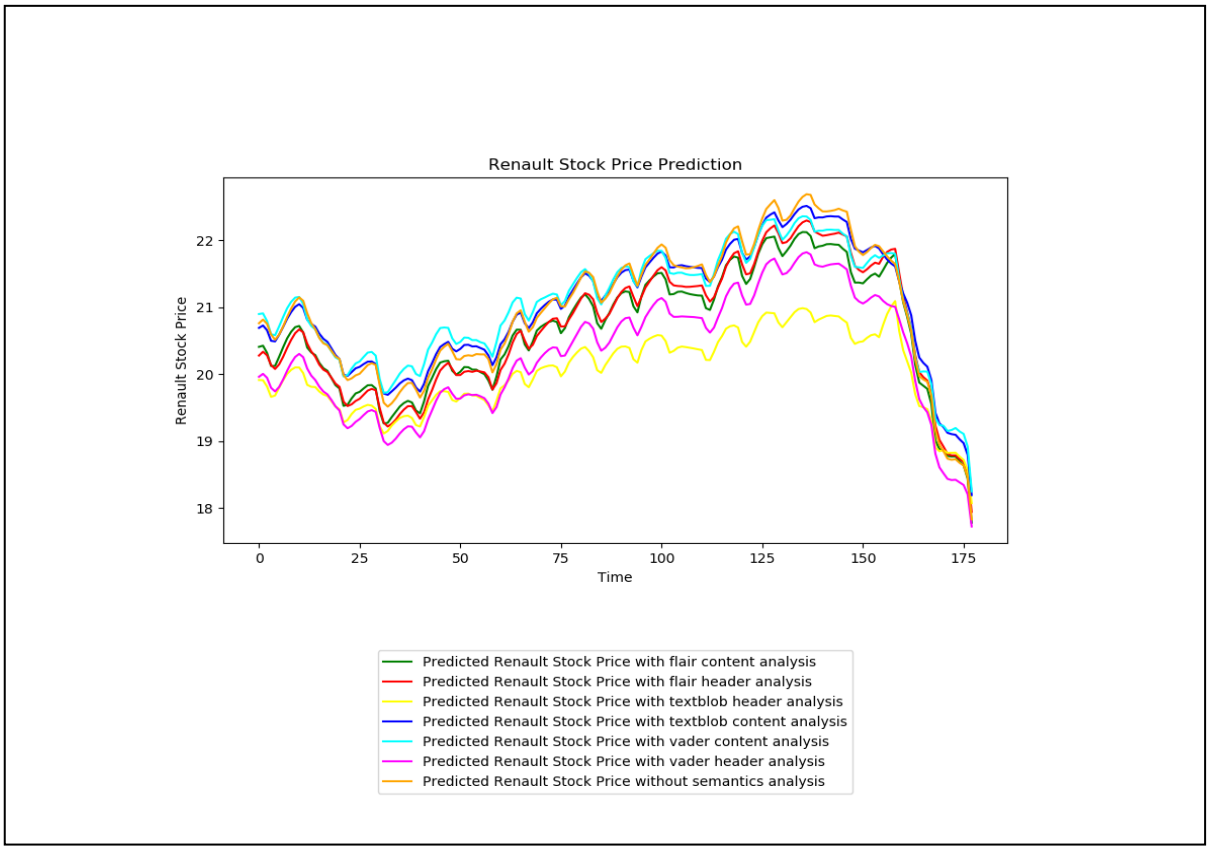


Figure 114: Stock Price Prediction Of Renault With Hourly Stock Price Data Using LSTM

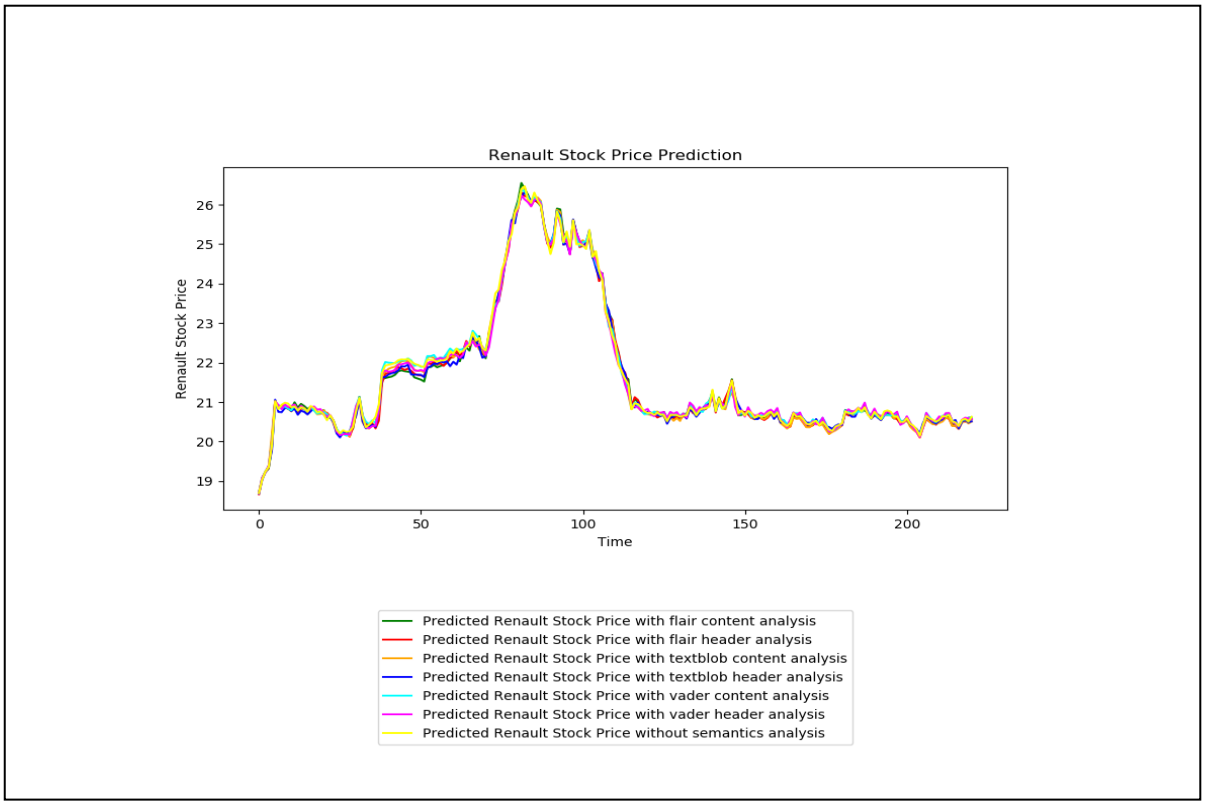


Figure 115: Stock Price Prediction Of Renault With Hourly Stock Price Data Using RandomForest Base Model

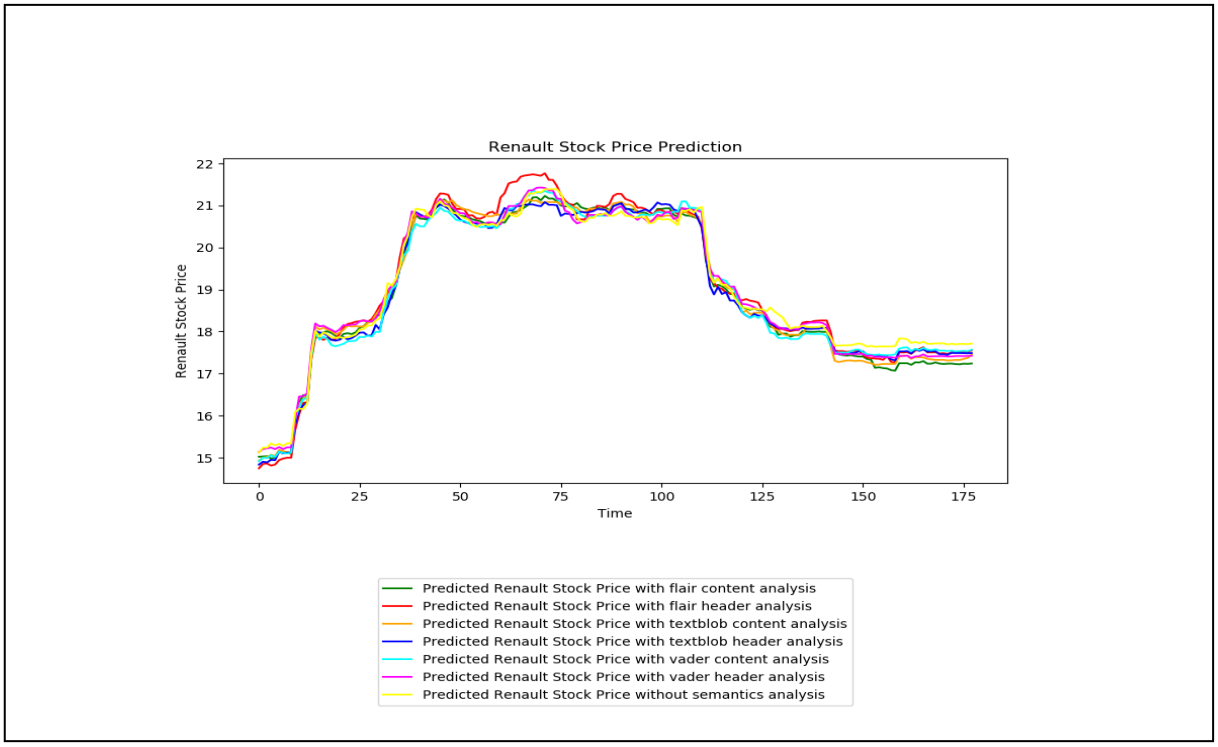


Figure 116: Stock Price Prediction Of Renault With Hourly Stock Price Data Using RandomForest Feature Model

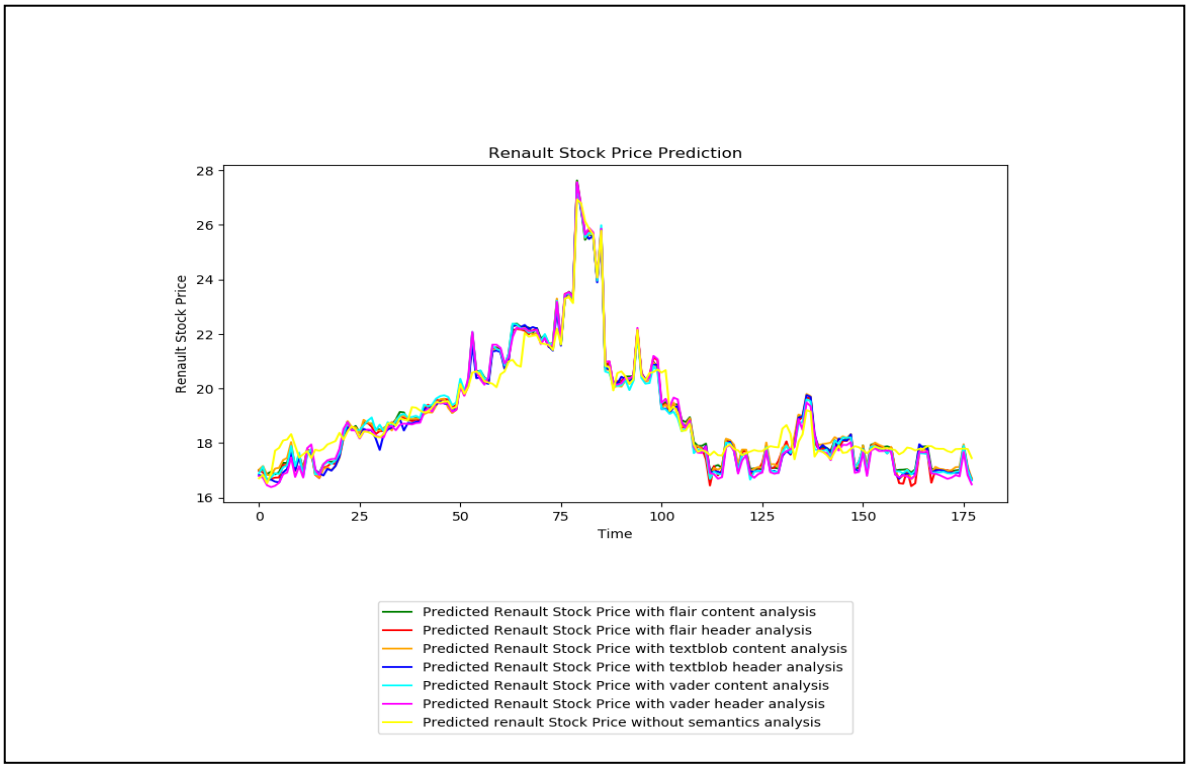


Figure 117: Stock Price Prediction Of Renault With Hourly Stock Price Data Using XGBoost

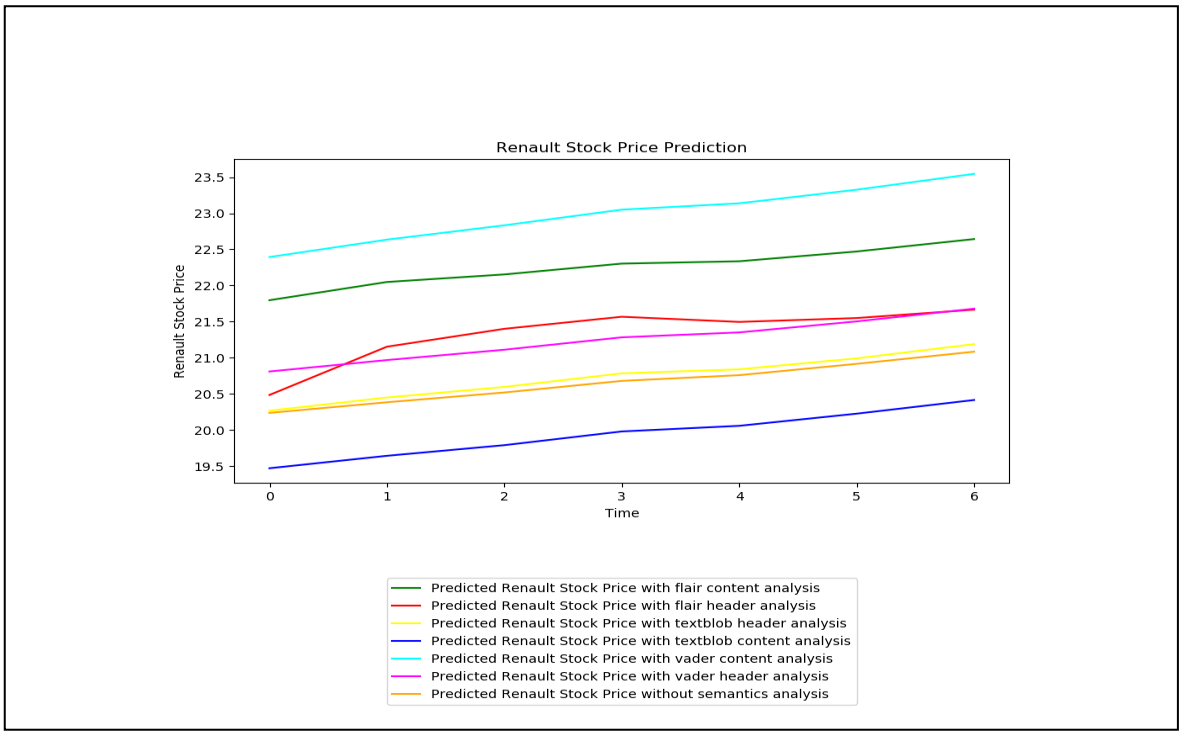


Figure 118: Stock Price Prediction Of Renault With Daily Stock Price Data Using LSTM

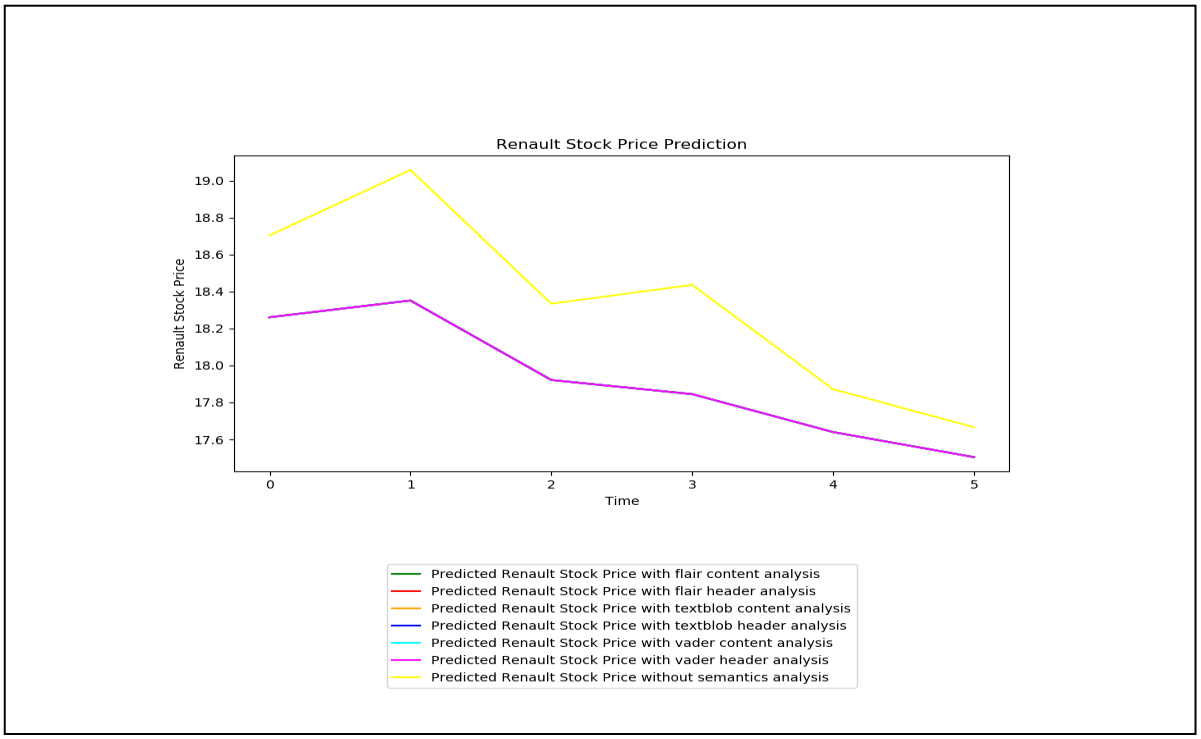


Figure 119: Stock Price Prediction Of Renault With Daily Stock Price Data Using RandomForest Base Model

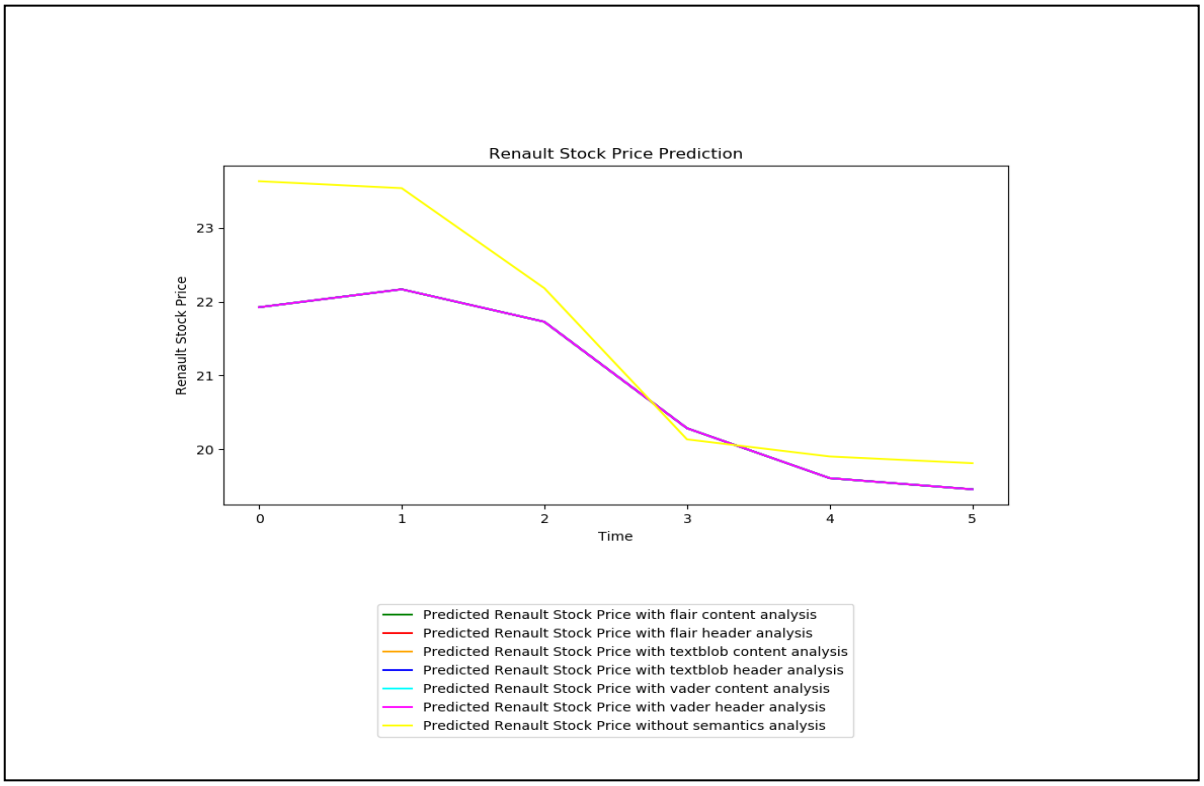


Figure 120: Stock Price Prediction Of Renault With Daily Stock Price Data Using RandomForest Feature Model

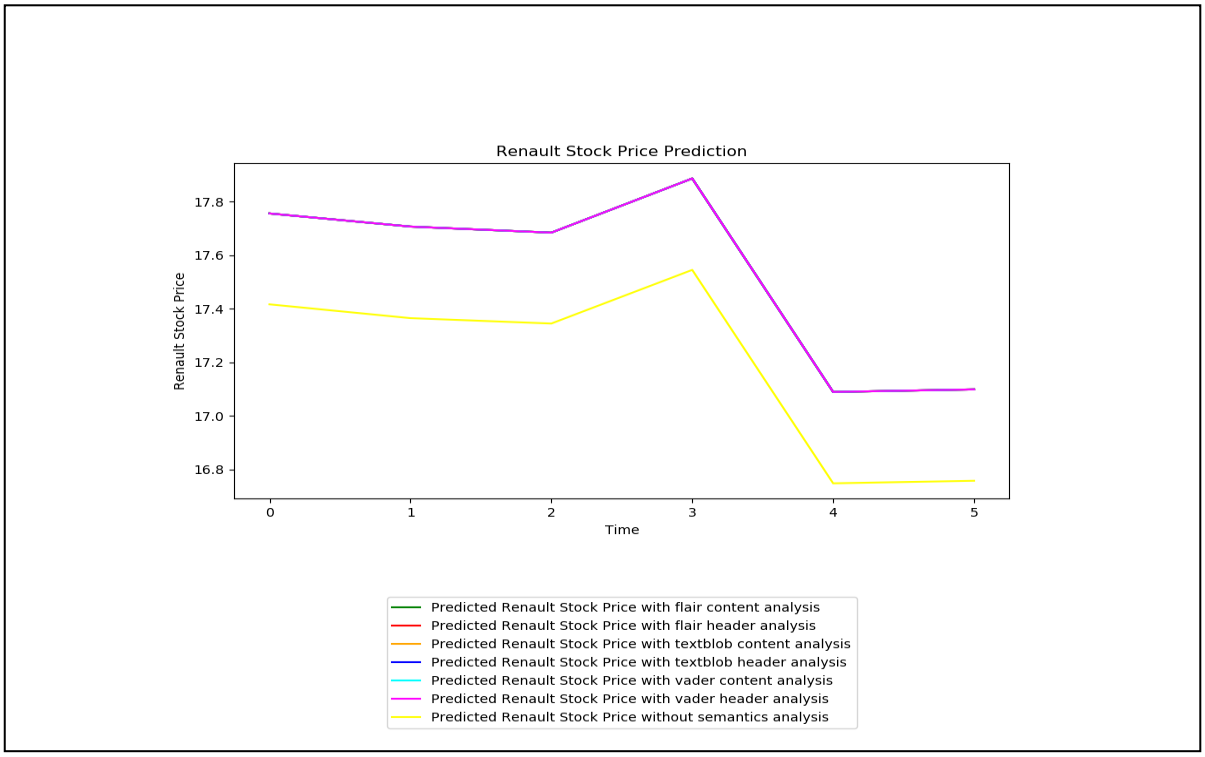


Figure 121: Stock Price Prediction Of Renault With Daily Stock Price Data Using XGBoost

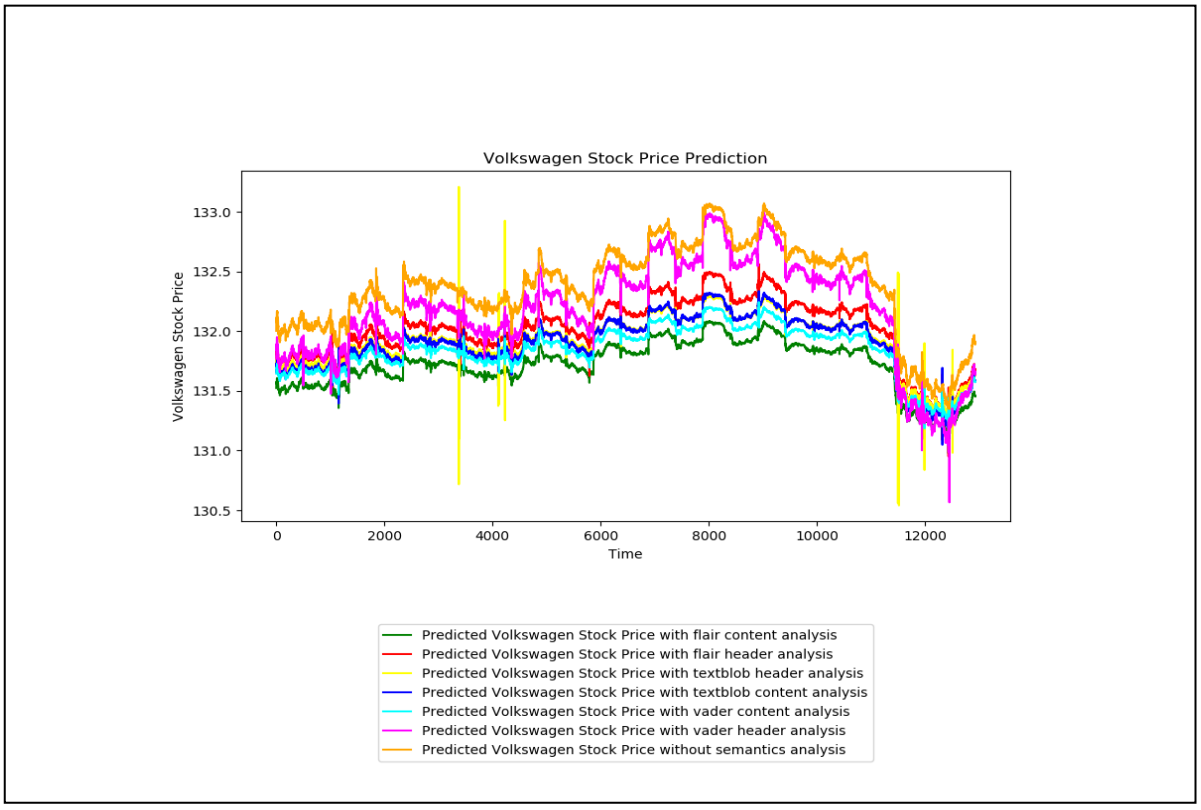


Figure 122: Stock Price Prediction Of Volkswagen With Minutely Stock Price Data Using LSTM

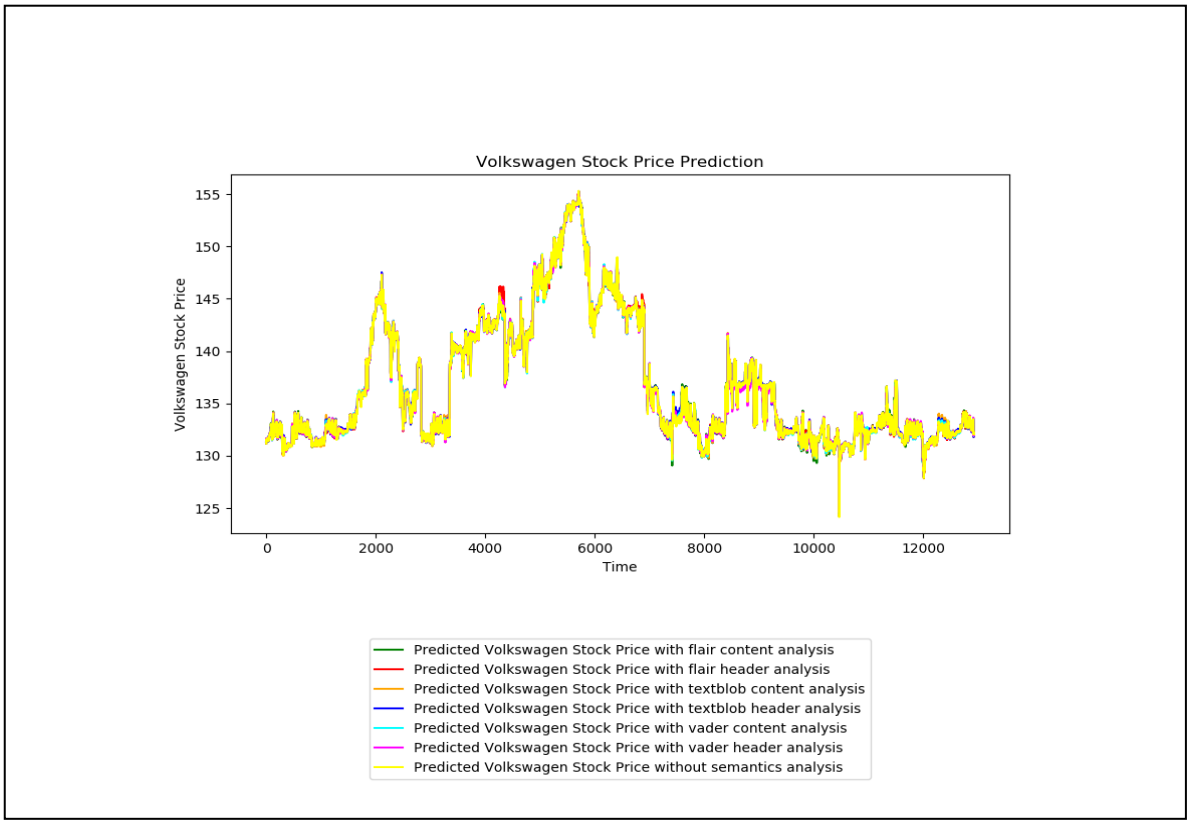


Figure 123: Stock Price Prediction Of Volkswagen With Minutely Stock Price Data Using Random-Forest Base Model

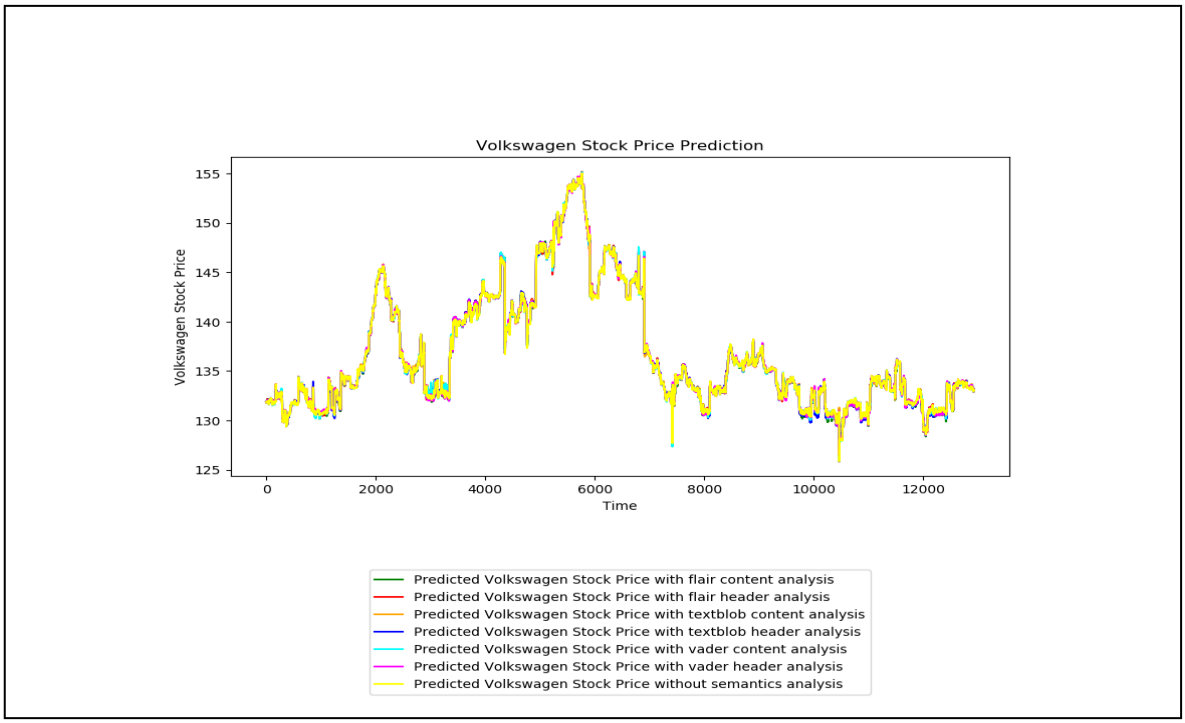


Figure 124: Stock Price Prediction Of Volkswagen With Minutely Stock Price Data Using Random-Forest Feature Model

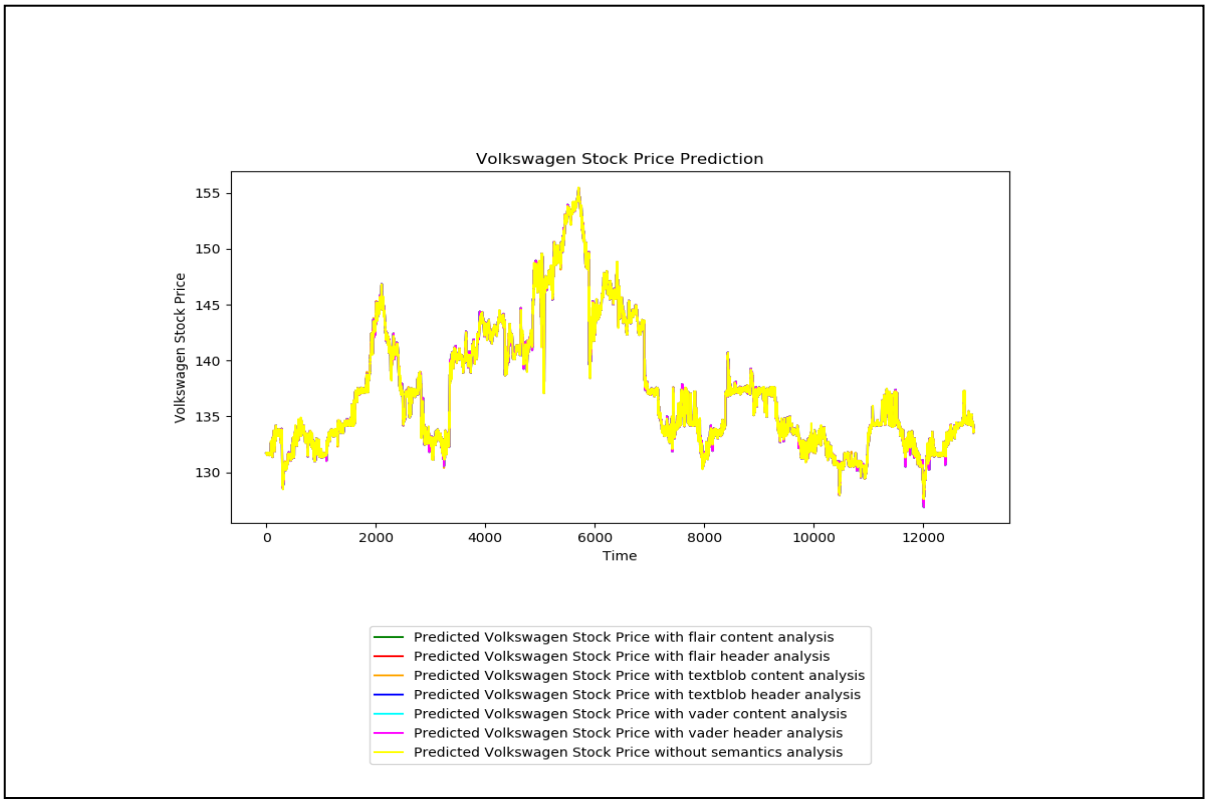


Figure 125: Stock Price Prediction Of Volkswagen With Minutely Stock Price Data Using XGBoost

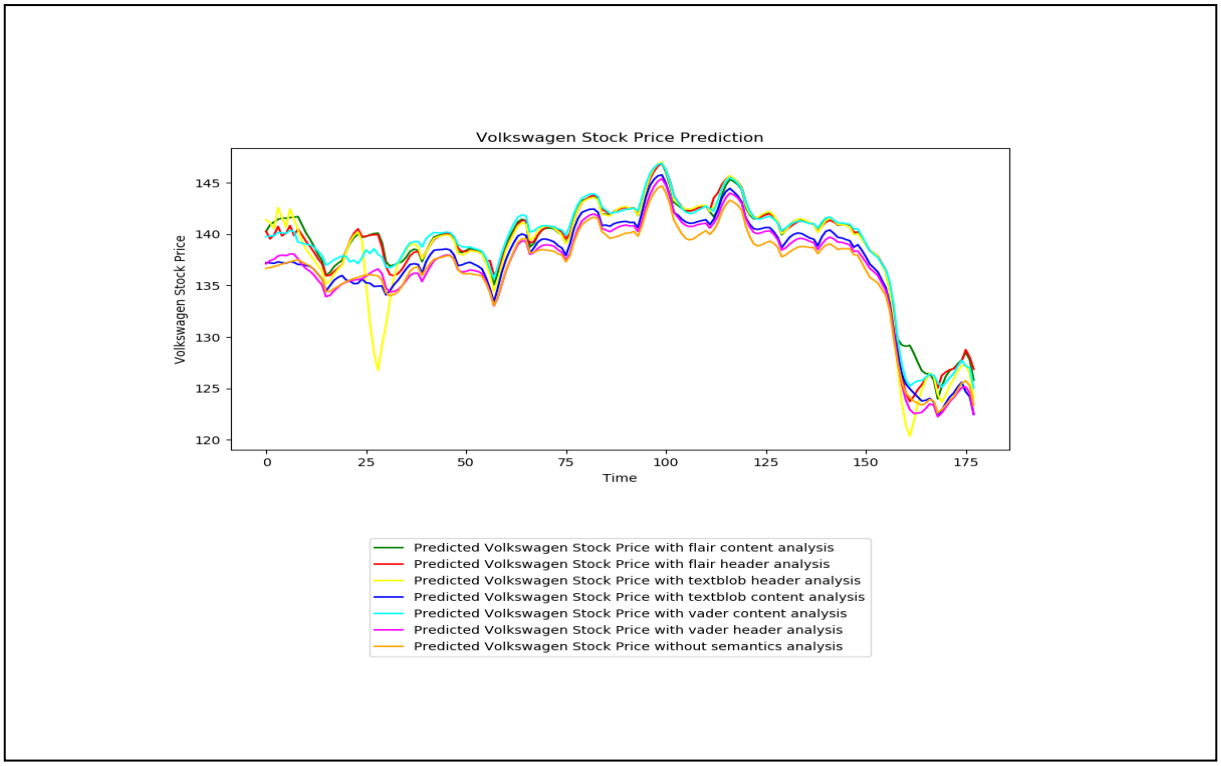


Figure 126: Stock Price Prediction Of Volkswagen With Hourly Stock Price Data Using LSTM

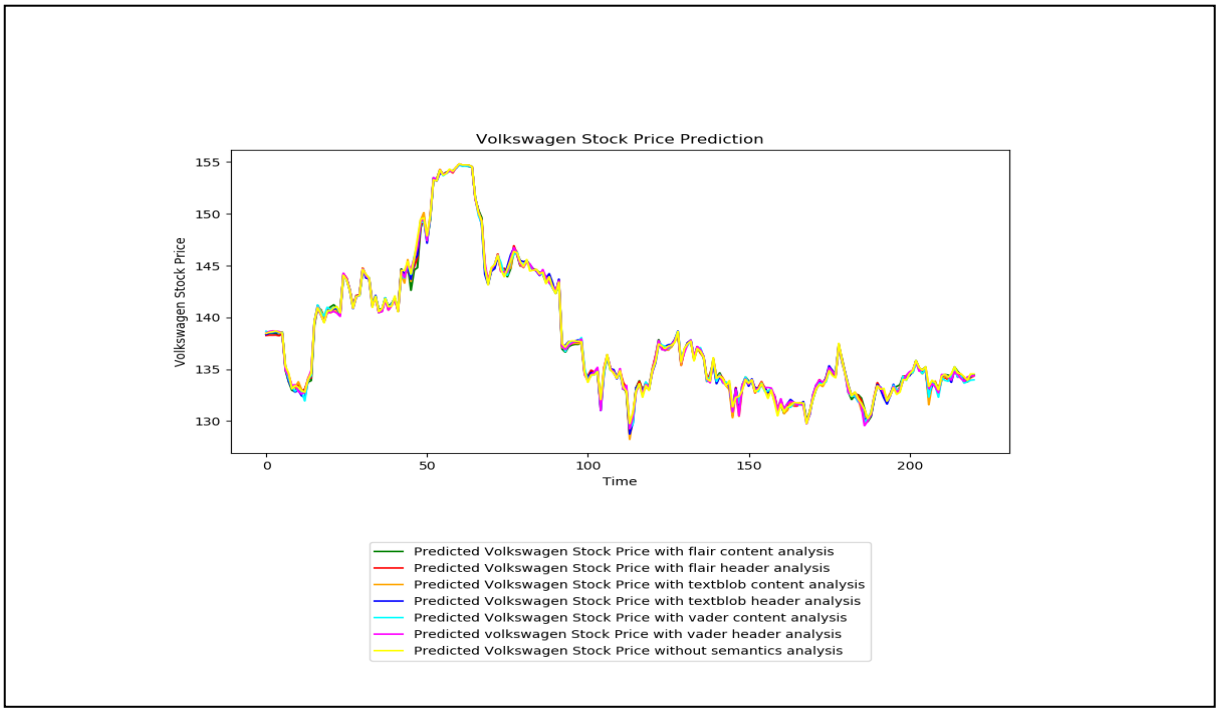


Figure 127: Stock Price Prediction Of Volkswagen With Hourly Stock Price Data Using Random-Forest Base Model

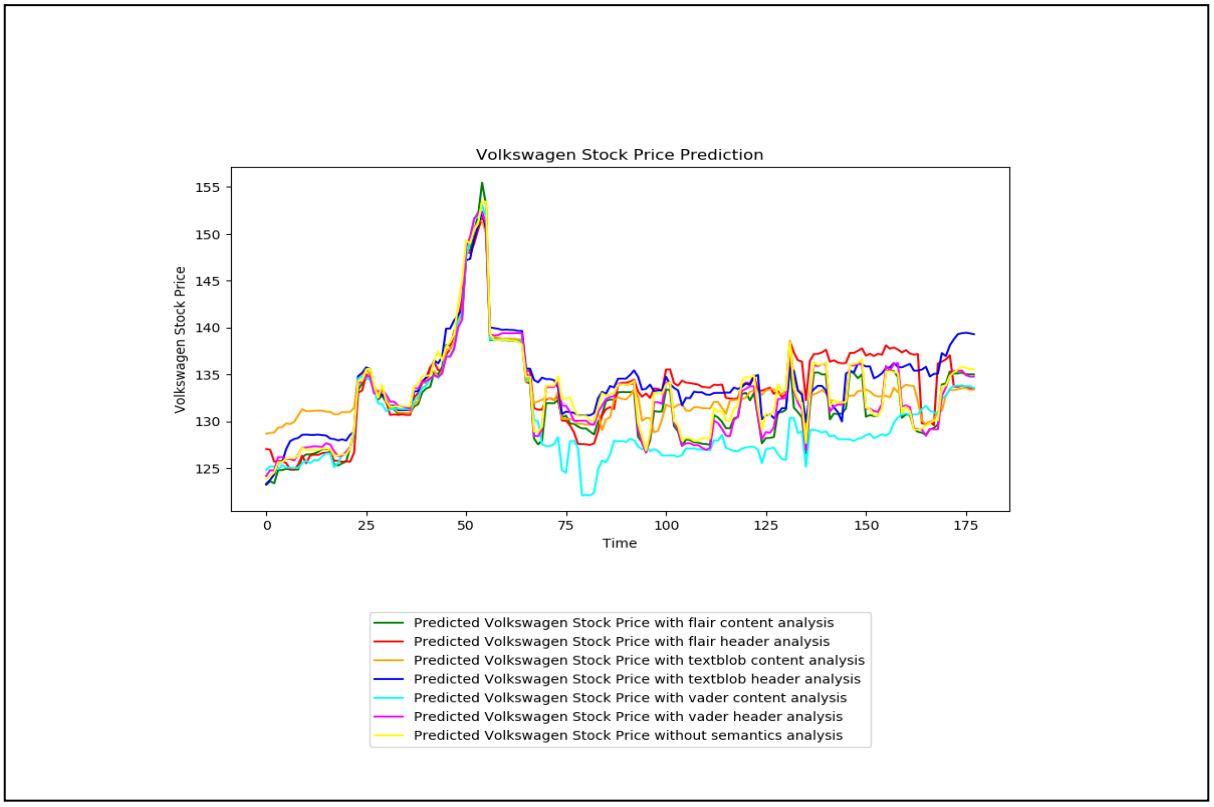


Figure 128: Stock Price Prediction Of Volkswagen With Hourly Stock Price Data Using Random-Forest Feature Model

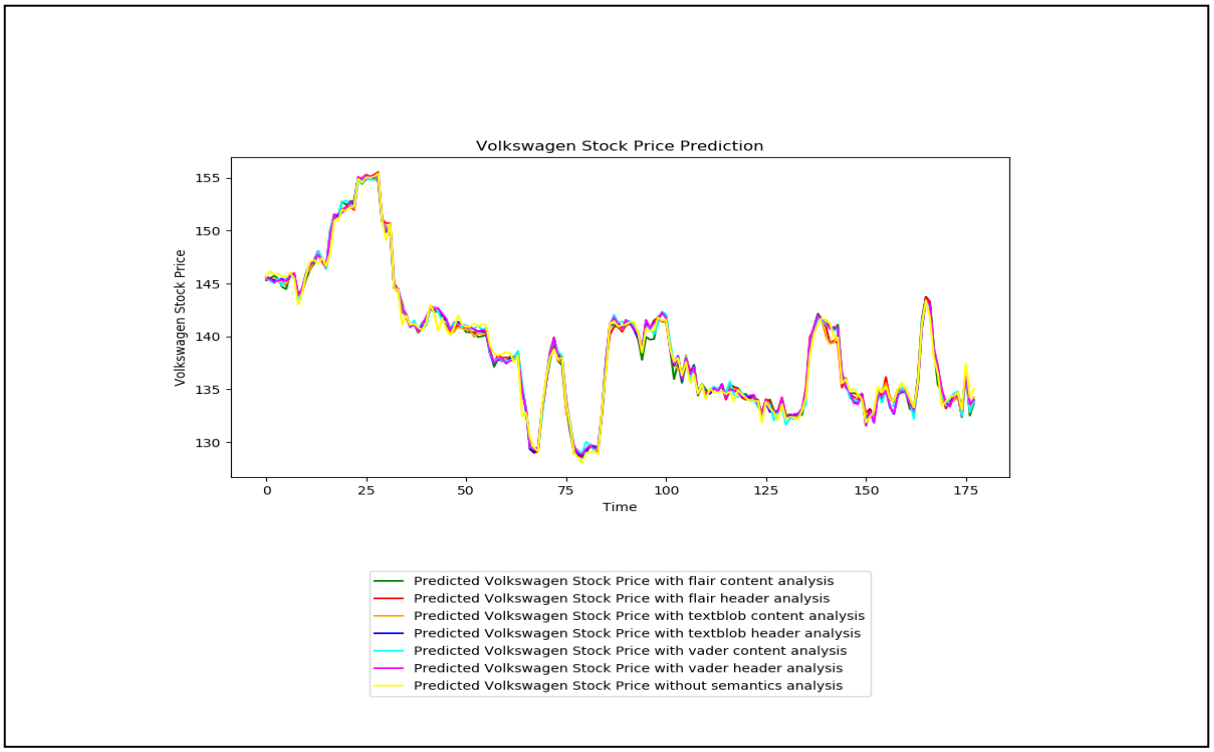


Figure 129: Stock Price Prediction Of Volkswagen With Hourly Stock Price Data Using XGBoost

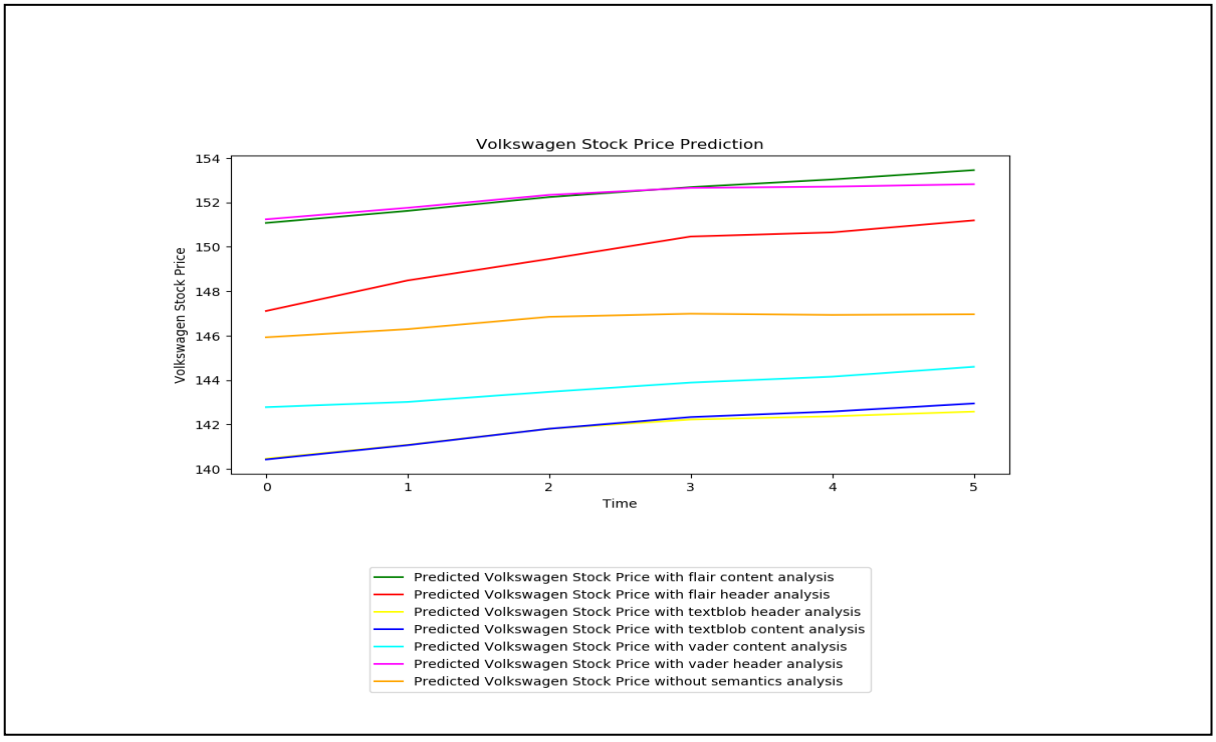


Figure 130: Stock Price Prediction Of Volkswagen With Daily Stock Price Data Using LSTM

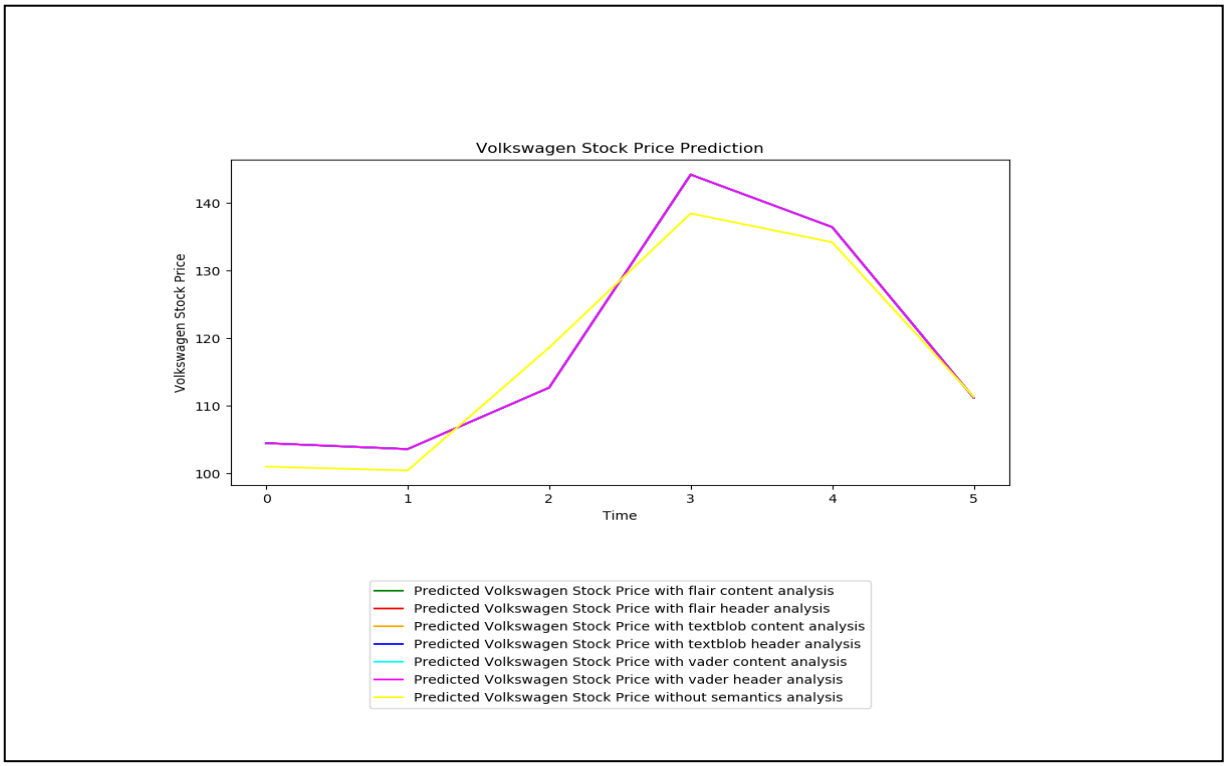


Figure 131: Stock Price Prediction Of Volkswagen With Daily Stock Price Data Using Random-Forest Base Model

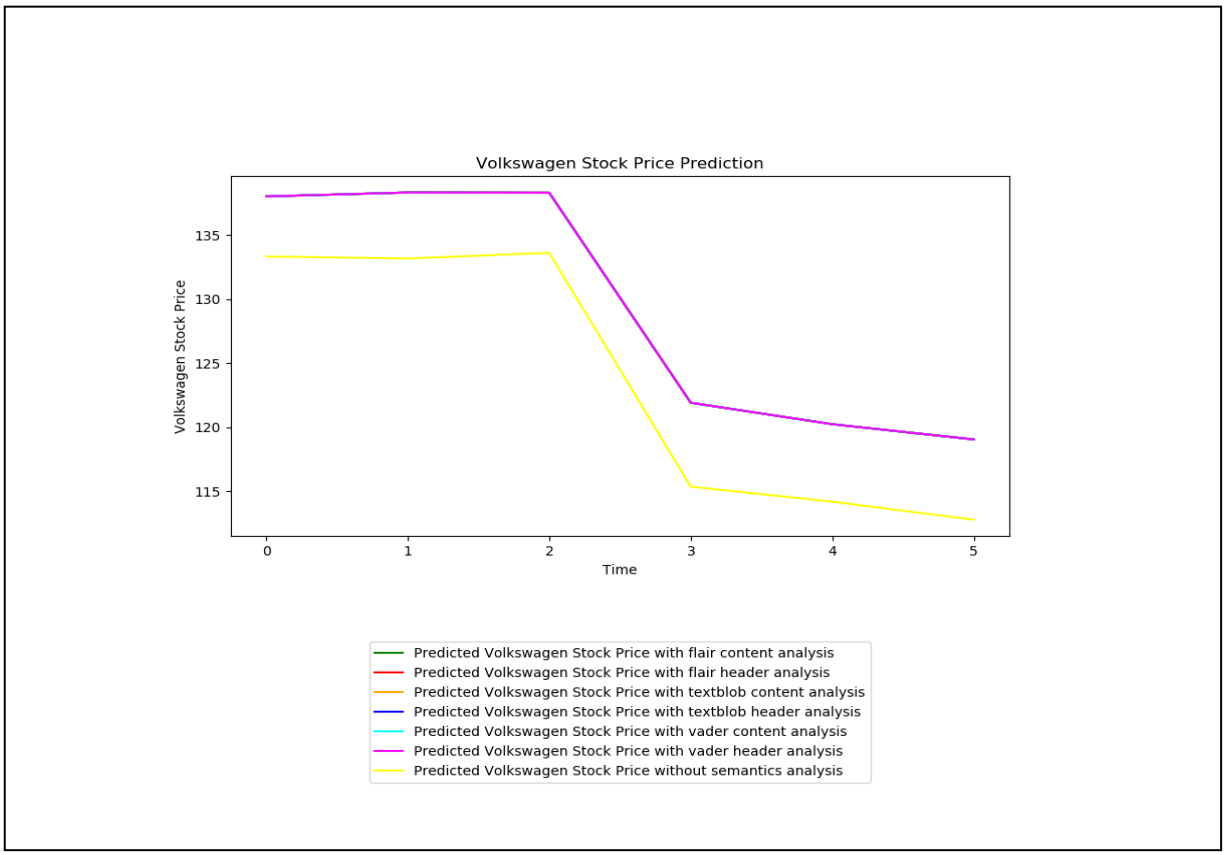


Figure 132: Stock Price Prediction Of Volkswagen With Daily Stock Price Data Using Random-Forest Feature Model

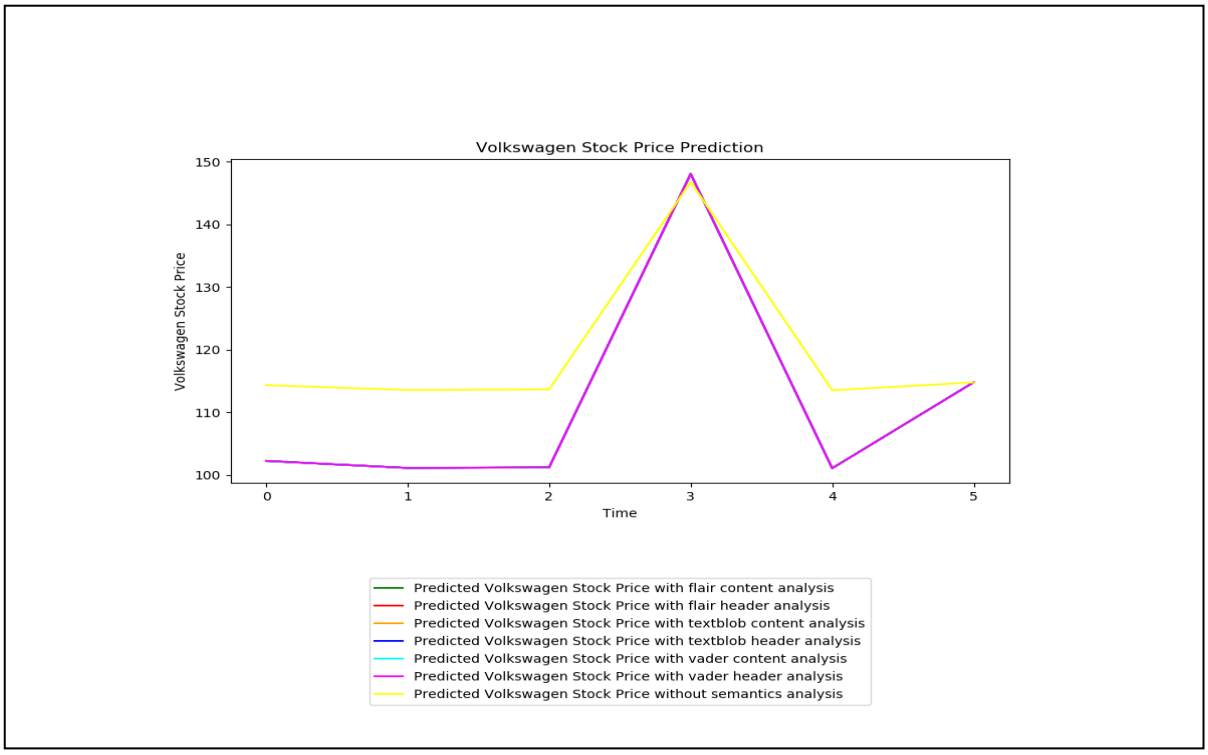


Figure 133: Stock Price Prediction Of Volkswagen With Daily Stock Price Data Using XGBoost

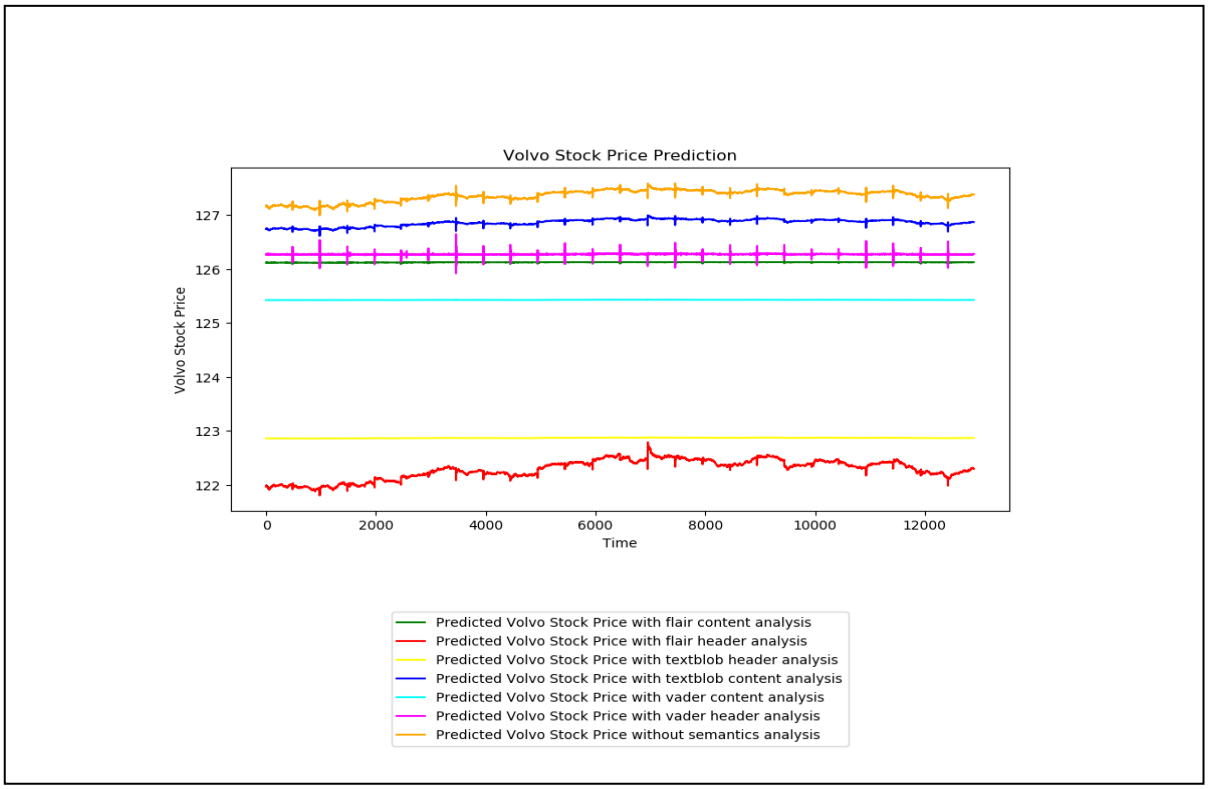


Figure 134: Stock Price Prediction Of Volvo With Minutely Stock Price Data Using LSTM

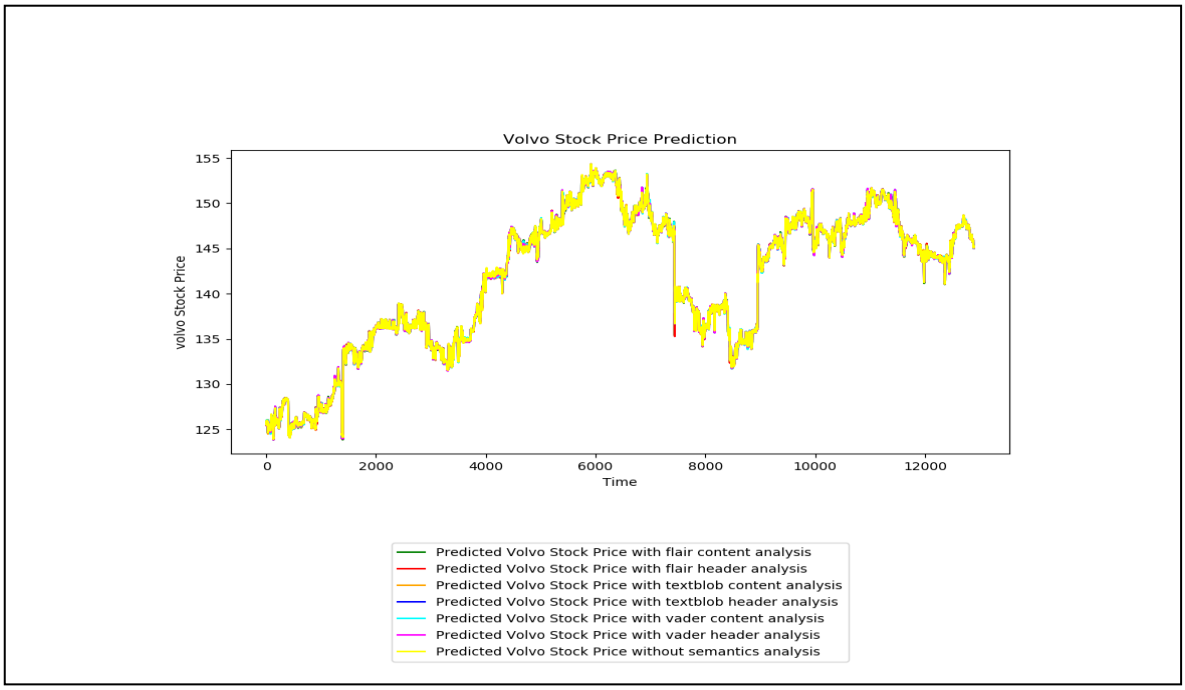


Figure 135: Stock Price Prediction Of Volvo With Minutely Stock Price Data Using RandomForest Base Model

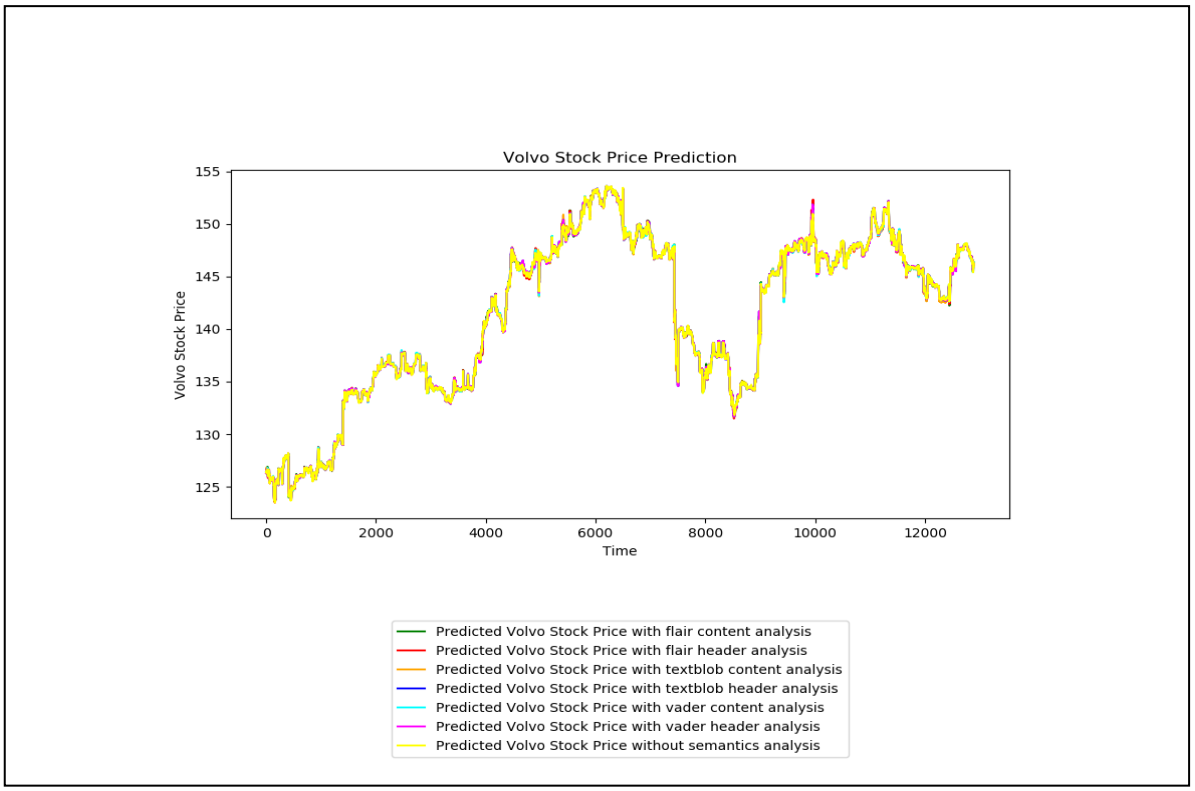


Figure 136: Stock Price Prediction Of Volvo With Minutely Stock Price Data Using RandomForest Feature Model

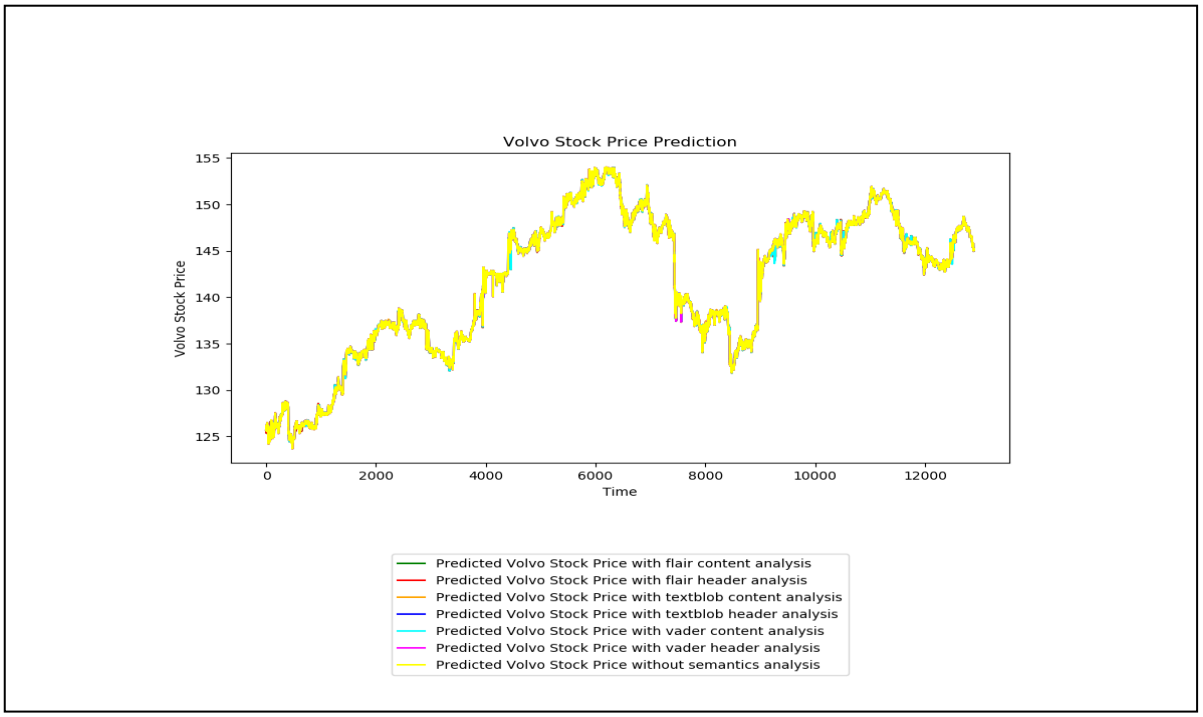


Figure 137: Stock Price Prediction Of Volvo With Minutely Stock Price Data Using XGBoost

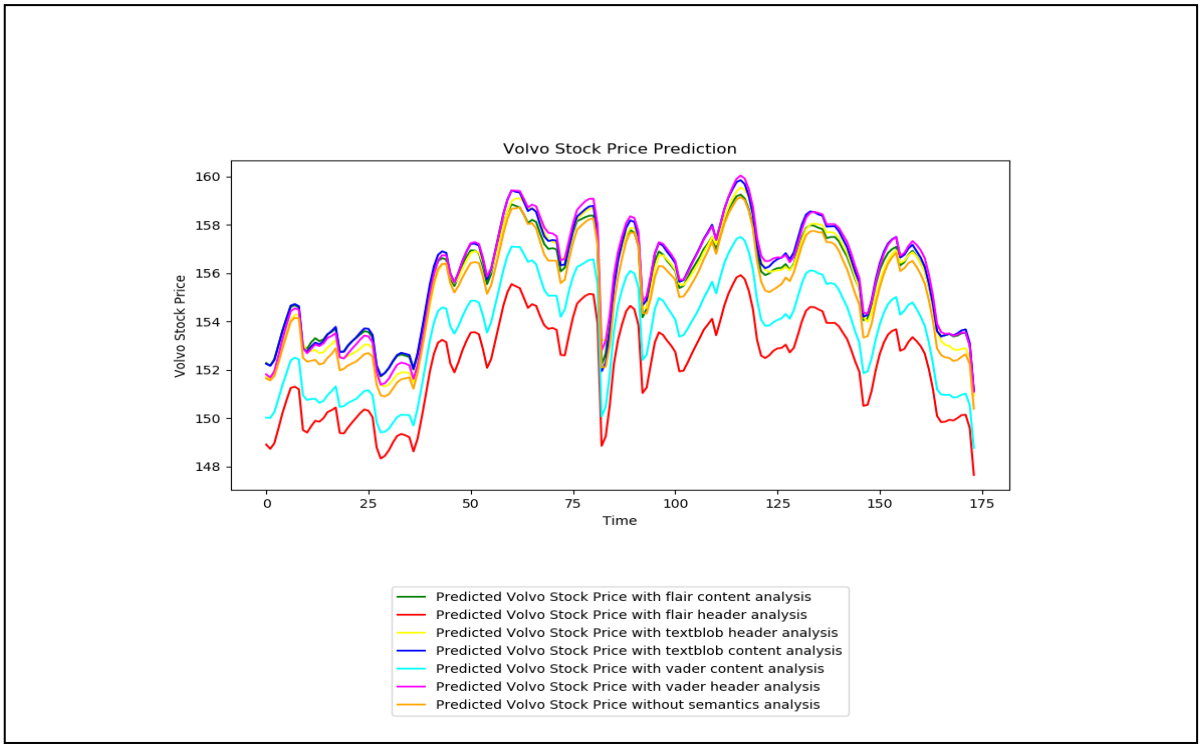


Figure 138: Stock Price Prediction Of Volvo With Hourly Stock Price Data Using LSTM

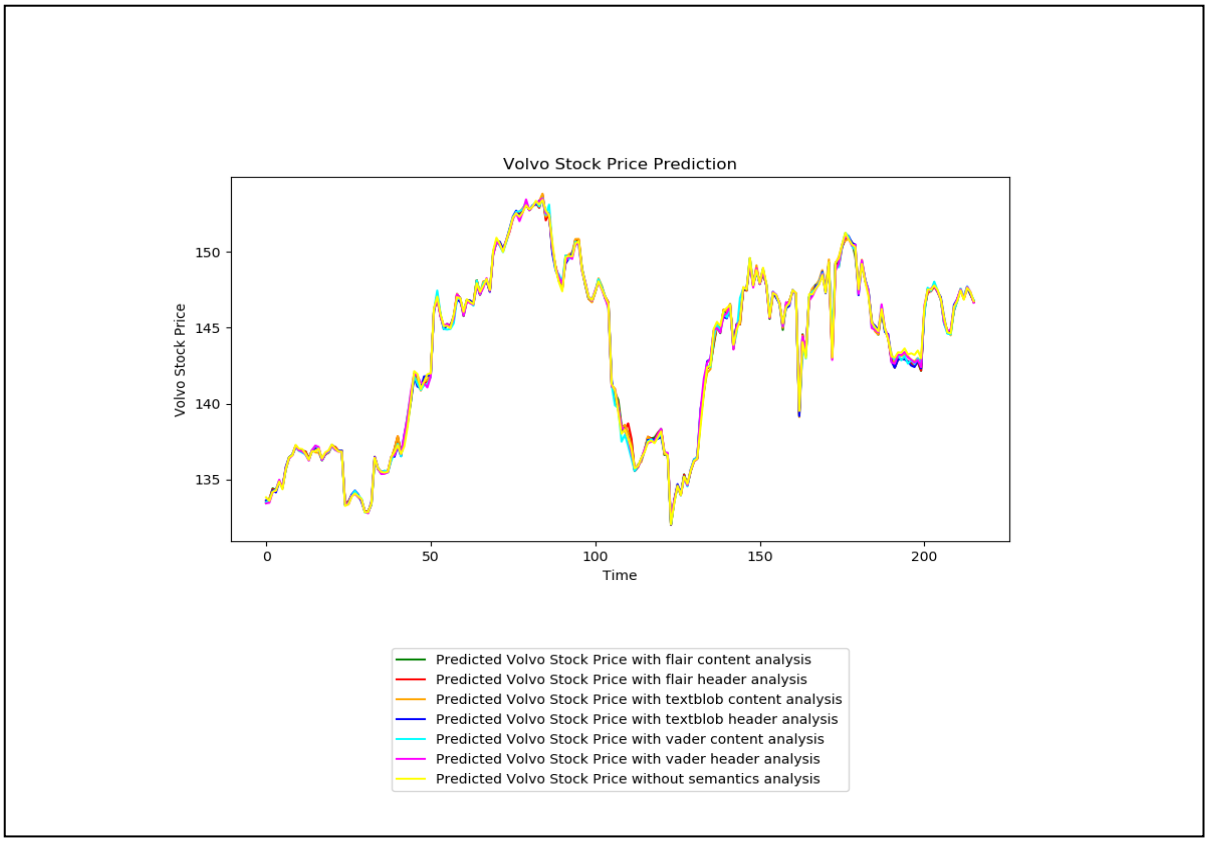


Figure 139: Stock Price Prediction Of Volvo With Hourly Stock Price Data Using RandomForest Base Model

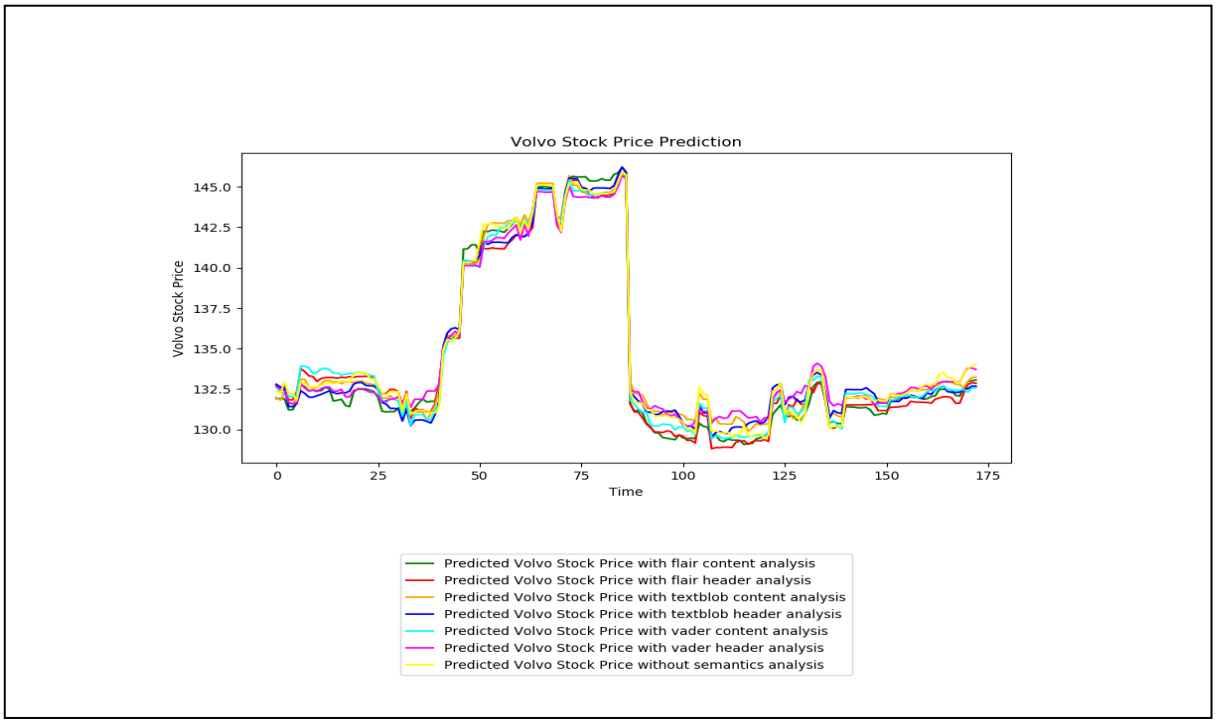


Figure 140: Stock Price Prediction Of Volvo With Hourly Stock Price Data Using RandomForest Feature Model

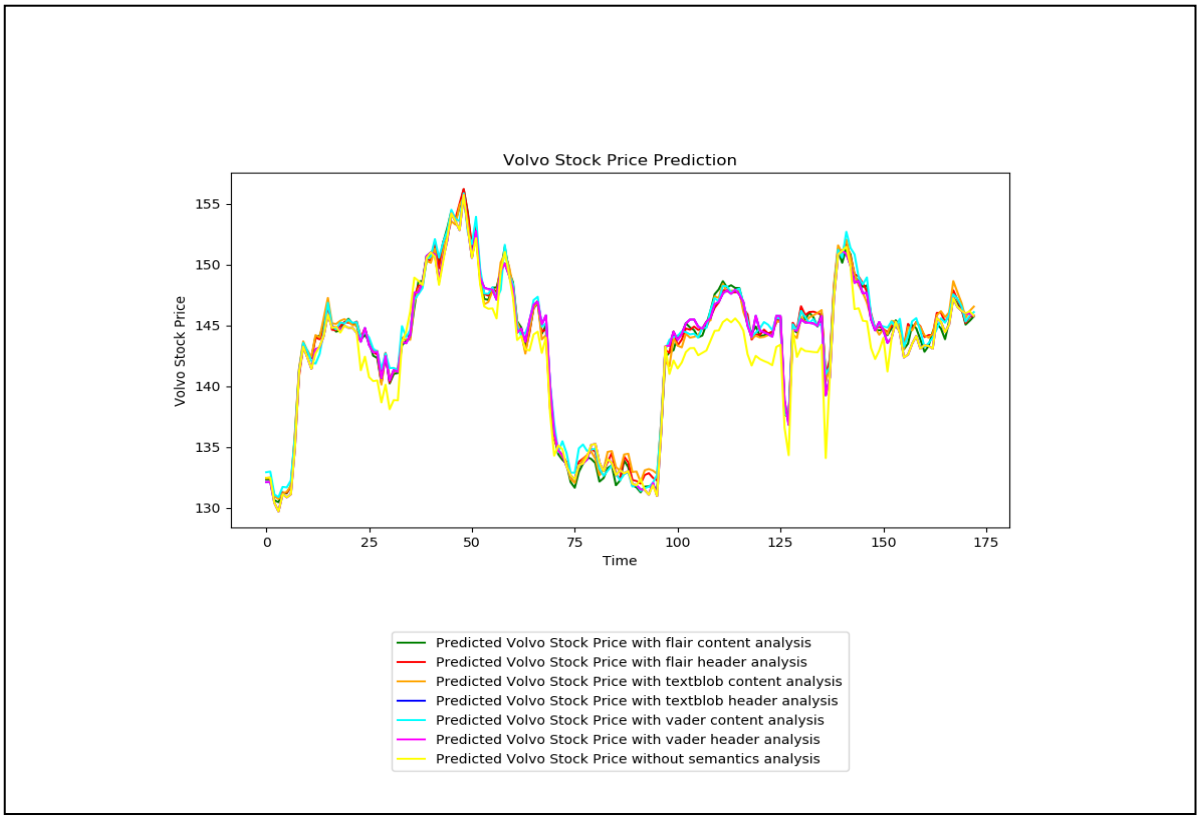


Figure 141: Stock Price Prediction Of Volvo With Hourly Stock Price Data Using XGBoost

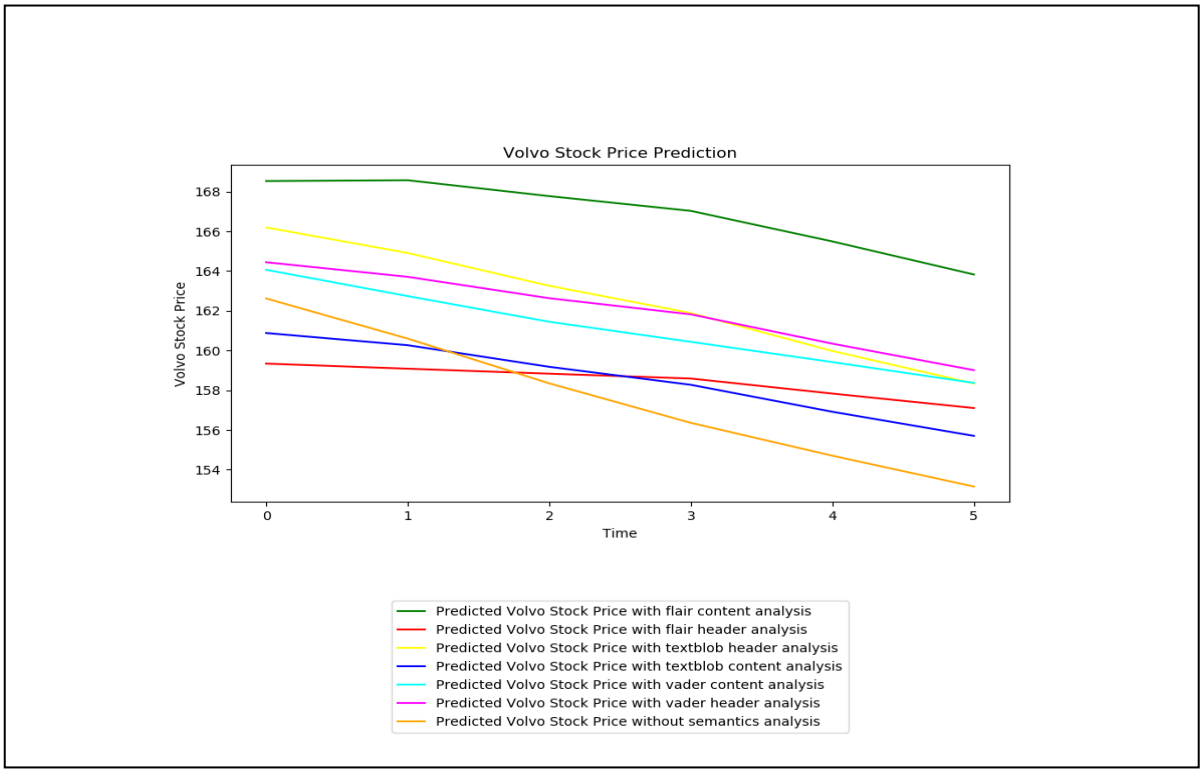


Figure 142: Stock Price Prediction Of Volvo With Daily Stock Price Data Using LSTM

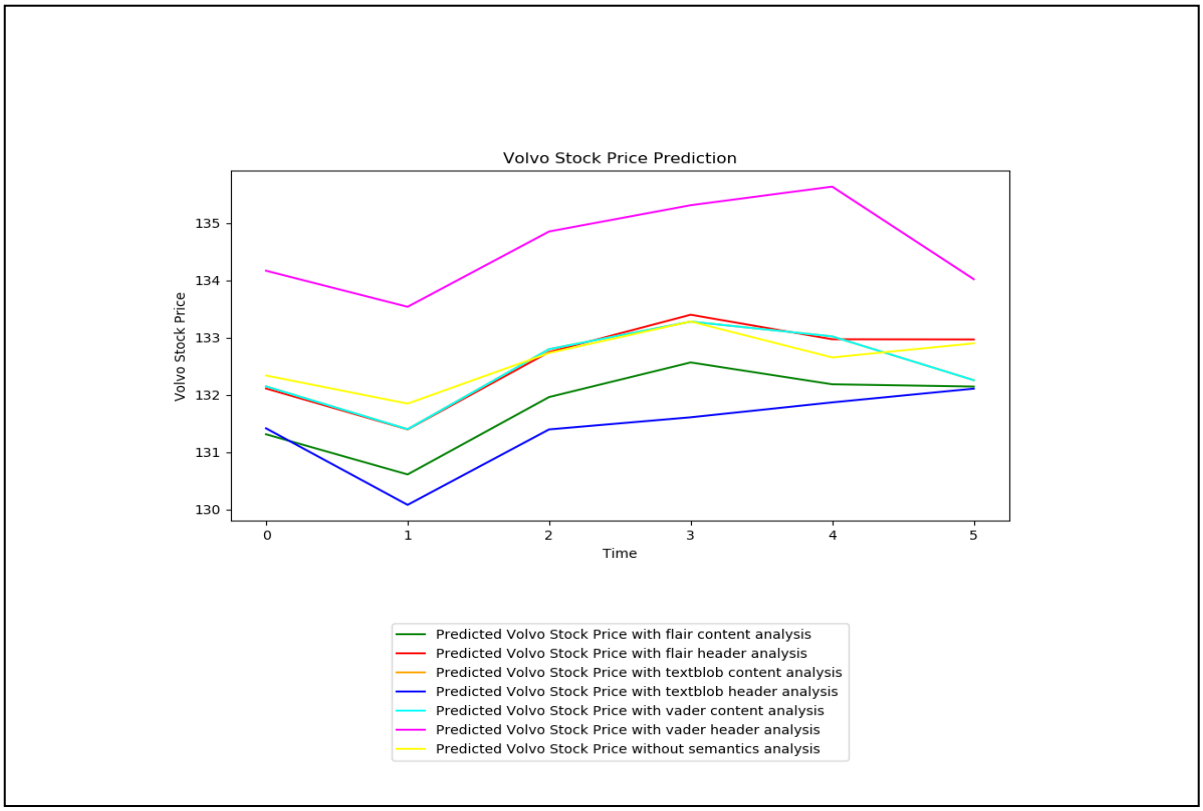


Figure 143: Stock Price Prediction Of Volvo With Daily Stock Price Data Using RandomForest Base Model

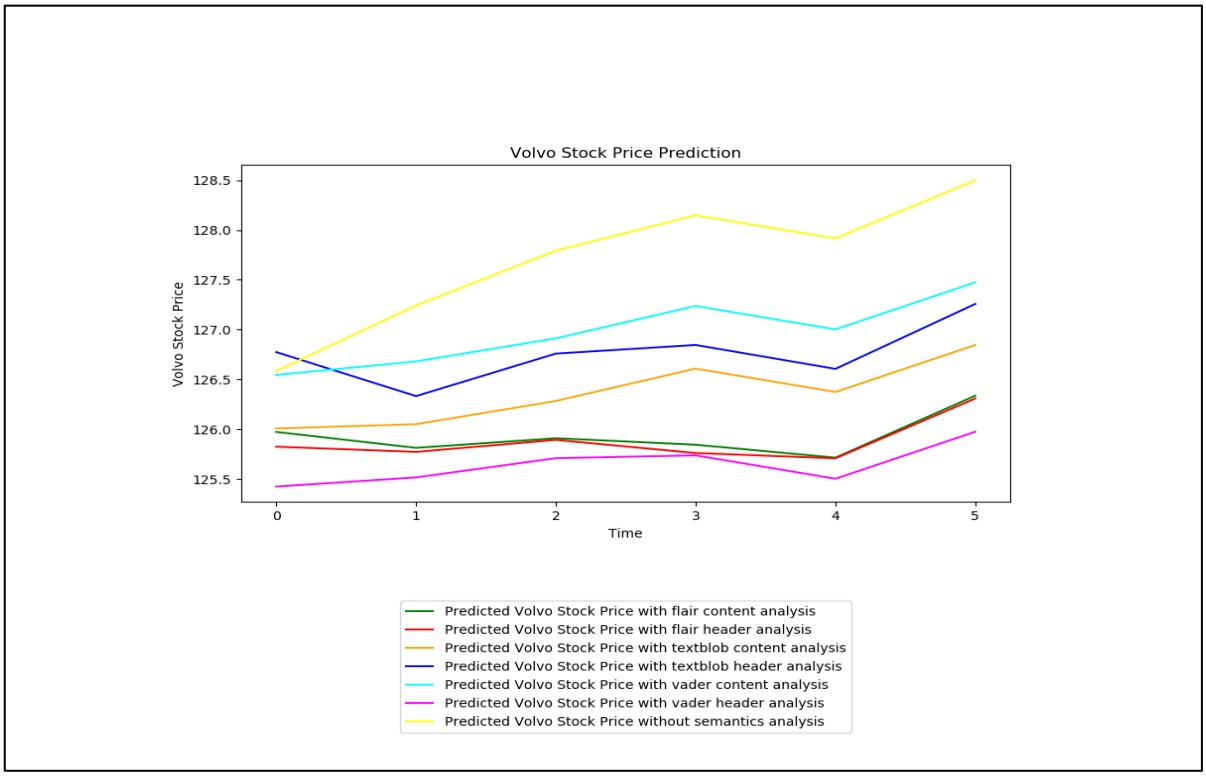


Figure 144: Stock Price Prediction Of Volvo With Daily Stock Price Data Using RandomForest Feature Model

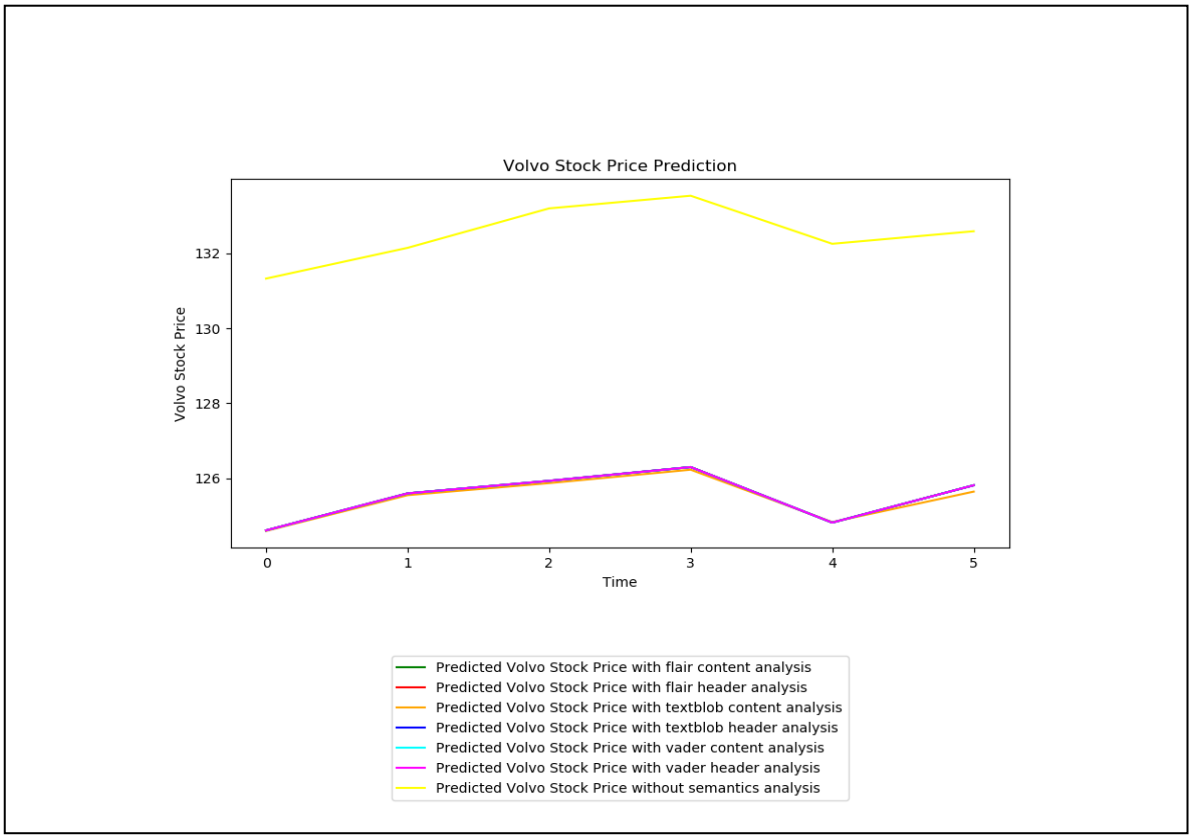


Figure 145: Stock Price Prediction Of Volvo With Daily Stock Price Data Using Daily