MEASURING DOMAIN-SPECIFIC SENTIMENT TO PREDICT STOCK PRICES

THE WALLSTREETBETS 'MEME-STOCK' SAGA

STEFAN WINTER

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2067606

COMMITTEE

Dr. Peter Hendrix
Dr. Raquel Garrido Alhama

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

January 10, 2022

WORD COUNT

8700 words

ACKNOWLEDGMENTS

I want to thank Dr. Peter Hendrix for his invaluable guidance during this thesis.

MEASURING DOMAIN-SPECIFIC SENTIMENT TO PREDICT STOCK PRICES

THE WALLSTREETBETS 'MEME-STOCK' SAGA

STEFAN WINTER

Abstract

Until the GameStop short-squeeze in early 2021, the impact of changes in investor-sentiment of the Reddit discussion board *WallStreetBets* on the financial market was vastly underappreciated. Due to the novelty of the WallStreetBets phenomenon, there is also almost no research available on that topic. This thesis will explore methodologies on how to measure sentiment of the discussion board and use the sentiment to predict changes of the GameStop stock price. One of the challenges when measuring the sentiment of WallStreetBets is the usage of novel domain-specific words and terminologies, which are shown to have a big impact on the results of sentiment analysis models. Hence, this thesis proposes a method to create a labeled dataset that covers the sentiment of text data, including the terminology of a given domain. It will be shown that supervised sentiment analysis machine learning models that use the domain-specific text corpus as input outperform general purpose lexicons, which are currently commonly used in both academia and industry to measure the sentiment of WallStreetBets. Furthermore, this thesis will demonstrate that stock prices can be predicted more accurately if the aforementioned sentiment is included as an input feature.

1 DATA, CODE AND ETHICS STATEMENTS

The code for this thesis is publicly available in the following github repository: `https://github.com/StefanWinterToo/Master-Thesis`
The Readme file in the repository lists all sources that were used in the thesis. The data required to complete this thesis was obtained from Reddit and Yahoo Finance. All the data are publicly available and accessible for free. The author of this thesis acknowledges that he does not have any legal claim to this data. All graphics, images and visualizations used in this thesis were created by the author of this thesis. To the best the author's knowledge, the literature used was referenced appropriately.

2 INTRODUCTION

Modern society has been able to access vast amounts of information, communicate ideas, and become part of communities with the advent of the internet. Online discussion boards are playing a critical role by providing a platform where people can do so. Those discussion boards are also used by a variety of people to talk about the stock market and discuss trading strategies. Recently, the Reddit forum WallStreetBets has become one of the most well-known and influential investing online-forums.

Even though the Reddit subforum was created in 2012 already, it received the majority of its media exposure in 2021 as a result of a short-squeeze of the GameStop (GME) stock, which drove the stock price up hundreds of percentage points (Diangson & Jung, 2021). Over the ensuing months, however, the stock price experienced extraordinary volatility. Prices fluctuated by double-digit percentage points which not only lead to gains, but also to large losses for market participants. Research shows that discussion board activity can be one cause of increased volatility (Das & Chen, 2007). Interestingly, finance scholars did not consider Reddit as a platform capable of having such a significant impact on the financial markets. As a result, the site has been neglected in their research (Long, Lucey, & Yarovaya, 2021).

However, it was neither the volatility nor the rapid price appreciation in the beginning of the short-squeeze of the Gamestop stock that astounded market observers. Instead, it was the unprecedented decentralized and coordinated buying of Gamestop shares by members of the WallStreetBets community that attracted attention (Anand & Pathak, 2021). Organizing the mass-coordinated buying of stock, however, requires that enough participants share the same sentiment. Furthermore, social media sentiment has a particularly strong impact on uninformed traders, of which WallStreetBets has plenty (Danbolt, Siganos, & Vagenas-Nanos, 2015). It is argued, that coordinated investments will also occur in the future, mainly due to the influence of social media and other online platforms on our society today (Semenova & Winkler, 2021). Hence, it is of the utmost importance to study and understand the impact of WallStreetBets. This thesis attempts to answer one of the numerous questions that have arisen with the growing popularity of Wallstreetbets, by answering the following research question:

> *Can sentiment analysis of the WallStreetBets Reddit-forum be used to predict daily changes in the stock price of Gamestop?*

To begin, it must be determined how the discussions about the Gamestop stock on WallStreetBets should be handled to serve as good input features

for sentiment analysis. One of the challenges is the heavy use of peculiar terminology and domain-specific phrases on the WallStreetBets forum, as well as many novel words (Anand & Pathak, 2021). According to recent research, sentiment lexicons and text-corpora with a focus on a certain domain produce superior sentiment analysis results compared to a general-purpose sentiment lexicon or text-corpus (Park, Lee, & Moon, 2015). Furthermore, the text data needs to be cleaned and pre-processed in order to be accurately processed by a machine learning algorithm (Jemai, Hayouni, & Baccar, 2021). As a result, the following sub-research question was formed:

RQ1 *How can the domain-specific language of the Reddit forum WallStreetBets be incorporated into sentiment analysis?*

Subsequently, machine learning models can be trained to perform sentiment analysis. However, each machine learning algorithm has its own idiosyncrasies and assumptions, and no single classifier works optimally in all possible scenarios. Hence, it is a good idea to evaluate the results and performance of different machine learning algorithms. As a result, the best model with a given set of hyperparameters can be selected to solve a particular problem (Raschka & Mirjalili, 2019, p. 53).

This thesis will explore traditional machine learning methods such as Naive Bayes (NB), as well as deep learning methods like a Long Short Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT). Due to the high dimensionality of textual data, deep learning methods have shown to outperform traditional machine learning techniques. That can be explained by the ability of deep learning methods to automatically learn the most important features, whereas traditional methods may suffer from the curse of dimensionality (Xianghua, Jingying, Jainqiang, Min, & Huihui, 2018). As was mentioned earlier, however, no classifier works best in all scenarios which is why the next sub-research question needs to be answered:

RQ2 *How do different sentiment analysis approaches perform based on the predefined evaluation metrics Accuracy, Precision, Recall and F_1-Score?*

An accurate measure of sentiment can represent an infromative feature, which is why the impact of sentiment on changes in stock prices has gained attention by researchers in recent years. However, predicting stock prices is a complex undertaking due to the nonlinearity and nonstationarity in the time series data of stock prices (Nikou, Mansourfar, & Bagherzadeh, 2019). Generally, this means that certain attributes of the data change over time, making it hard to forecast stock prices (Shetty & Ismail, 2021). One

established stock-price forecasting method is Auto Regressive Integrated Moving Average (ARIMA), which captures temporal structures in time-series data. It has shown strong predictive results (Caginalp & Constantine, 1995). However, it is not designed to include other features, such as sentiment. This is why this thesis will also implement an LSTM, which has demonstrated promising predicitive capabilities with regards to time-series data, especially when sentiment is included. (Chen, Zhang, Mehlawat, & Jia, 2021; Jin, Yang, & Liu, 2020). This leads to the following sub-research question:

RQ3 *Do machine learning models show stronger predictive capabilities in forecasting the stock price of Gamestop if sentiment obtained from WallStreetBets is included as a feature, based on the predefined evaluation metrics MAE, MSE and RMSE?*

By answering the three sub-research questions, a scientific and thought-out answer for the main research question can be found.

3 RELATED WORK

Gauging sentiment of online forums to predict movements in stock prices has been a research subject for many years now. Das and Chen (2007) conducted a study on the *Yahoo!* message board, which was amongst the first ones on the internet for investors to exchange ideas. Others did similar studies on platforms such as *Twitter, Reddit and StockTwits* and (Anand & Pathak, 2021; Gu & Kurov, 2020; Piñeiro-Chousa, Vizcaino-Gonzalez, & Perez-Pico, 2017). However, since the WallStreetBets *'meme-stock movement'* is a relatively recent phenomenon, there is very little research on that topic and none that accounts for the novel terminology used on the forum to create sentiment models which can be used to predict changes in stock prices. As a result, incorrect conclusions may be drawn.

3.1 *Domain Specific Terminology*

Even though the literature suggests many innovative ways to enhance model performance by a few percentage points, the biggest benefits seem to come from high quality input data in the form of a domain-specific knowledge base. It has been demonstrated that adapting data to a certain domain results in more accurate sentiment analysis results (Park et al., 2015). Furthermore, it is argued that there is no general-purpose sentiment lexicon that can be optimally applied on all domains. This is due to different meaning of terms, depending on the domain. A good example is

the word *unpredictable*, which would be associated with negative sentiment for a car but can be a positive label for movies (Pang & Lee, 2008).

One proposed approach is to adapt sentiment lexicons to a specific domain (Yue, Malu, Umeshwar, & ChengXiang, 2011). This adapted lexicon can then be searched to find and score the sentiment of a specific word (Muhammad, 2014). While lexicon-based methods have found widespread adoption, mainly due to their simplicity, more advanced machine learning methods have shown strong performance with regards to domain specificity (Yanyan, Fulian, Jianbo, & Marco, 2020).

Machine learning methods can be used to automatically detect and identify domain-specific words in sentences. By doing so it is assumed that the algorithm can not only detect whether domain-specific words are used, but also identify the position of the term in the sentence. Hence, it is possible to detect new meanings of words in an already existing text corpus. In addition, this approach also allows to classify novel words, that do not yet exist in a lexicon (Zhengqi, Zhewei, & Yang, 2019). This can be achieved by having models that formulate domain-specific word detection as a sequence-labelling task. Furthermore, novel domain-specific words can be learned by understanding the contextual structure of a sentence (Zhengqi et al., 2019). For example, out-of-vocabulary tokens can be learned in the hidden layers of LSTMs (Hochreiter & Schmidhuber, 1997).

To train machine learning models, however, a labeled corpus is needed. Obtaining one is not without its challenges. For example, working with multiple human annotators can lead to discrepancies in the annotation results (Jin-Dong, Tomoko, & Junichi, 2008; Salah & Gayar, 2019). Additionally, it is hard to estimate the total annotation cost which can depend of various factors. One example would be whether the annotator is capable of fluently understanding the language for the given task (Arora, Nyberg, & Rose, 2009). Additionally, labelling an entire dataset incurs extremely high costs, which can be avoided. With the support of an Active Learner, a complete domain-specific corpus with its respective labels can be created using only partial annotations (Park et al., 2015).

One of the key concepts of Active Learners is that if a machine learning algorithm is allowed to choose the data from which it learns, it will achieve higher accuracy with less training data. If a considerable amount of the data is unlabeled, this is especially desirable. As a result, the total cost of annotation can be reduced drastically. Research shows that the total number of manual annotations can be reduced by 80% when using an Active Learner instead of randomly selecting data to label (Baldridge & Osborne, 2004).

In comparison, if data are manually annotated at random (passive learning), the annotator will invest a lot of time into labeling irrelevant

instances. That is especially true if the class distribution of the data is imbalanced or if there are many very similar documents. For example, if a specific feature set appears on only 1% of instances, the annotator would have to label 1000 documents to cover the feature set on 10 relevant documents. When it comes to document similarity, large clusters of very similar documents might be identifiable. Because features may be barely distinctable, the annotator might spend a lot of effort labeling uninformative instances when selecting them randomly. An Active Learner, on the other hand, suggests which instances the annotator should label. Those instances can be determined on various quantitative metrics (Miller, Linder, & Mebane, 2020).

Based on the reviewed literature, an Active Learner seems to result in a highly accurate labeled dataset that contains domain-specific terminology which can then be learned by machine learning models.

3.2 *Sentiment Analysis*

Once an annotated text corpus is obtained and pre-processed it can be used by machine learning models to perform sentiment analysis.

To further optimize performance, models can be improved, by applying a character-based convolutional neural network to encode the spelling of words (Zhengqi et al., 2019). Other research shows that the accuracy of an LSTM can be improved by introducing Word2Vec to the LSTM. As a result, the one-hot encoded input to the LSTM is converted into a low dimensional vector that covers the semantic similarity of the words in it. Due to the lower dimensionality, over-fitting can be prevented and the network may also need less parameters (Xiao, Wang, & Zuo, 2018). However, Gennaro, Buonanno, and Palmieri (2021) argue that there is almost no research on specific choices of hyperparameters for the Word2Vec model. Hence, this thesis will add to the literature by considering varying hyperparameters of the Word2Vec embedding.

Other research demonstrates the importance of large pre-trained models using transfer learning (Deng et al., 2009). Devlin, Chang, Lee, and Toutanova (2019) introduce BERT, a pre-trained model that uses the English Wikipedia and the BooksCorpus, which shows promising results. One of the advantages is that only one output layer needs to be added to the model to achieve state-of-the-art sentiment analysis performance. However, it is also shown that BERT lacks domain awareness. Hence, it cannot differentiate between properties of source and target domains. However, it is also argued that a vanilla implementation can still outperform other machine learning models (Du, Sun, Wang, Qi, & Liao, 2020). Other research that compares BERT to a lexicon approach shows that on average BERT

achieves better performance. However, the better performance cannot be observed on all analyzed test sets which keeps the authors "optimistic about the lexicon-based approach in general" (Kotelnikova, Paschenko, Bochenina, & Kotelnikov, 2021).

However, deep-learning models typically require much more computing power compared to traditional machine learning methods. The Naive Bayes method, in contrast, is very easy to implement and fast to train. As a result, the classifier is oftentimes used as a baseline for text classification. Multinomial Naive Bayes (MNB), one type of the Naive Bayes classifier, has established itself as a standard for text classification (Abbas et al., 2019). MNB is a frequency based method that calculates the conditional probability of a word belonging to a specific class (Susanti, Djatna, & Kusuma, 2017). It is argued that MNB performs better than many rule-based lexicas, which are oftentimes used as baseline models (S. Wang & Manning, 2012). Furthermore, MNB seems to perform especially well if the words in the document are shown to be significant for the classification problem (Sharif, Hoque, & Hossain, 2019).

3.3 *Stock Price Prediction*

Using machine learning to predict stock prices has been a research topic for many years now. It was shown that high accuracy is attainable within a short prediction time span (Schöneburg, 1990). However, De Gooijer and Hyndman (2006) found that most published papers mainly focus on time-series forecasting while excluding other features that may also lead to an improvement in performance. It is shown that social media sentiment can have a direct effect of how market participants perceive a company, which can lead to changes in the stock price of companies. This is especially true for smaller firms with hardly any analyst coverage (Feng & Johansson, 2019). Other researchers show that sentiment obtained from Twitter can be used to predict returns of a broader stock market index (Gu & Kurov, 2020). Additionally, Antweiler and Frank (2004) uncover that positive sentiment has a very significant relationship with returns. In other research the emotions of discussions on WallStreetBets are studied by performing sentiment analysis.

Long et al. (2021) tried to uncover the impact of specific emotions such as *"Angry, Fear, Happy, Sad and Surprise"* from the comments on WallStreetBets discussions on intraday changes of the stock price of the affected stock. While they conclude that the tone as well as the number of comments have an impact on the stock price, they show that the number of comments is not directly related to sentiment. They also argue that any asset that is targeted by a large crowd from WallStreetBets can become a

subject of excessive volatility, without being driven by any fundamental reasons. Lyócsa, Baumöhl, and Vyrost (2021) also showcased that as the discussion volume on WallStreetBets increased, the volatility of certain stocks got amplified. Additionally, the research of Zaghum, Mariya, Imran, and Shoaib (2021) also found that sentiment of investors on WallStreetBets affected the returns of the Gamestop stock. However, they also demonstrate that other features such as the put-call ratio and the short-sale volume had a strong impact on the stock price. As a result, this thesis will also include other features, besides the obtained sentiment, as input to the machine learning models.

Most of the aforementioned research on WallStreetBets, however, uses general purpose lexicas to perform sentiment analysis. Hence, this thesis will add to the literature by juxtaposing a lexicon based approach with machine learning algorithms. However, not all algorithms and models can implement sentiment analysis into their stock price prediction.

One of those models is ARIMA, which is widely used in the financial industry. ARIMA works well on time-series data due to its ability to catch time-series specific features (Vuong, Dat, Mai, Uyen, & Bao, 2021). It does so by gauging the strength of one dependent variable based on other independent ones that may change. One of the limitations of ARIMA is that the time-series data needs to be stationary, meaning there are no trends in the data. This can be achieved by differencing the data. By applying the *Augmented Dickey-Fuller* test, the condition for stationarity can be tested (Ivanovic, Bogdan, & Baresa, 2013).

Other models that can include sentiment are LSTMs, which have shown strong results for time series prediction (Rammurthy & Patil, 2021). That is because of their strength in analyzing connections among time-series data by using the LSTM's memory function. Other feed-forwards neural networks, as a comparison, cannot handle the complex time correlation between information. Even though the results seem promising when using an LSTM to predict the stock prices, the performance of the LSTM can be enhanced by accounting for sentiment as well (Jin et al., 2020; H. Wang, Guo, & Chen, 2019).

4 METHODOLOGY

4.1 *General Description*

For better understanding of the methodologies used this paragraph will provide a brief summary of the steps taken, before explaining each step in more detail.

To begin with, stock prices, and posts from the WallStreetBets subreddit were obtained for GameStop. Since the posts were completely unlabeled, ten percent of the data were manually labeled at the document level. This also allowed to explore the data and helped in determining how to best create a fully annotated dataset. The labeled targets are either *bearish, neutral or bullish*. Generally, bearish is associated with negative sentiment where investors assume a decline in stock prices. Bullishness, on the other hand, relates to positive sentiment where investors hope for rising stock prices. Subsequently, it was determined that an Active Learner is a good solution to label the rest of the data. As a result, *Ground Truth* data were created which can be used for supervised machine learning sentiment models. Afterwards, the model with the best evaluation metrics was used to predict the sentiment of the entire dataset. The sentiment was then used as an additional input feature for the stock price prediction task.

4.1.1 *The Case for a Semi-Supervised Method over an Unsupervised Method to Label Data*

Since the data obtained from Reddit are unlabeled, they cannot be fed into supervised machine learning algorithms. That is because supervised sentiment analysis methods rely on labeled data (Sazzed & Jayarathna, 2021). One approach to label data is using unsupervised machine learning models. Unsupervised models are commonly applied in Natural Language Processing and text classification (Namcheol & Ghang, 2019). However, unsupervised models are a better choice for uncovering hidden patterns in a dataset, especially without any a priori knowledge of the structure of the data. As a result, unsupervised models excel at summarizing or exploring a large text corpus. For the case at hand, a *t-Distributed Stochastic Neighbor embedding (t-sne)* algorithm was applied on the data to extract similarity features and project them onto a lower dimension (Binu & Sony, 2020). As can be seen in Figure 1, admittedly at a low dimension, the majority of the data do not belong to any particular cluster. The data used for the t-sne algorithm were the manually labeled text data that were transformed into a *Term Frequency-Inverse Document Frequency (tf-idf)* representation. By using tf-idf, the relevancy of a word in a set of documents is determined. If a word occurs often, but in many documents it will rank low, as such a word might not mean much to a particular document. tf-idf is calculated by multiplying the tf part with the idf part. The tf represents how often a word appears while the idf part accounts for the importance of the word, depending on the number of documents the word can be found in (Guia, Silva, & Bernardino, 2019; Sugitomo, Kevin, Jannatri, & Suhartono, 2021).

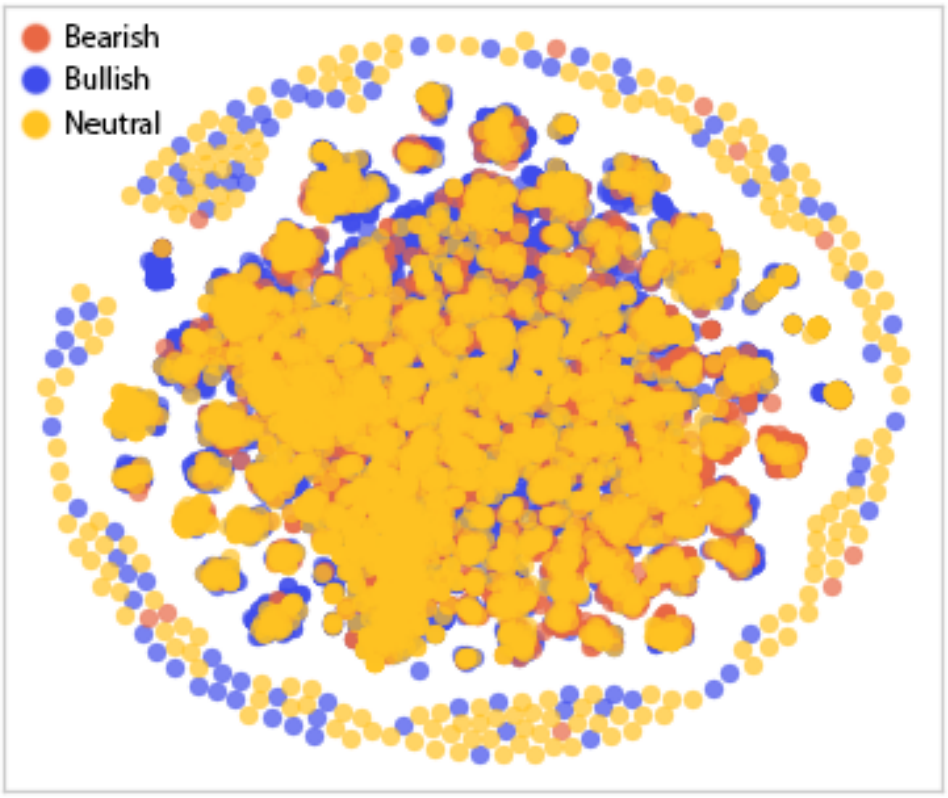

Figure 1: t-sne Visualization of Seed Data

Even though there are some approaches to clustering high dimensional data, it generally is difficult to do so accurately. One of the explanations is the increased sparsity and the difficulty to distinguish between the distances of specific instances (Tomasev, Radovanovic, Mladenic, & Ivanovic, 2014). For these reasons, an unsupervised approach was not chosen.

However, labeled data are still needed to train supervised machine learning models for sentiment analysis. While manually labelling all data might be the most accurate solution, it is associated with high costs (Miller et al., 2020). Hence, this thesis proposes the implementation of an Active Learner. With its support, a complete domain-specific corpus can be labeled while only relying on partial annotations (Park et al., 2015). As a result, a domain-specific labeled dataset is created that can be used as input to different supervised machine learning algorithms.

4.1.2 *Active Learner Workflow*

The illustrated workflow in Figure 2 provides an overview of how an Active Learner works. To begin with, cleaned and pre-processed data needs to be available that can be used by the Active Learner. Furthermore, the Active Learner can also be trained with some initial training data, which is also referred to as the seed. All the unlabeled instances will become the pool data, which need to be labeled. The seed data is fed into the Active Learner and trains an estimator, which needs to be defined when creating the Active Learner.

In addition, a query strategy needs to be defined, based on which the Active Learner queries new instances from the aforementioned pool. A query strategy evaluates the informativeness of unlabeled samples. Common strategies are *uncertainty sampling, query-by-committee, expected model change, expected error reduction and variance reduction*. While each strategy has its own intricacies, all essentially try to find instances that are hard for the model to classify and hence might benefit from manual annotation. After the query function selected instances from the pool, an oracle needs to label those. An oracle normally is at least one human with knowledge on how to annotate the data at hand (Settles, 2009). Once the new instances are labeled, those instances need to be removed from the pool, since they are now part of the labeled data. The Active Learner then needs to be taught the new instances, which he can use to adjust the model. After each iteration, the results can be evaluated. A common performance measure for Active Learners is *accuracy*.

If a predefined stopping criterion is not yet met, the query strategy selects more instances from the pool and repeats the process. If the stopping criterion is met, the process ends (Lu, Henchion, & Namee, 2019).

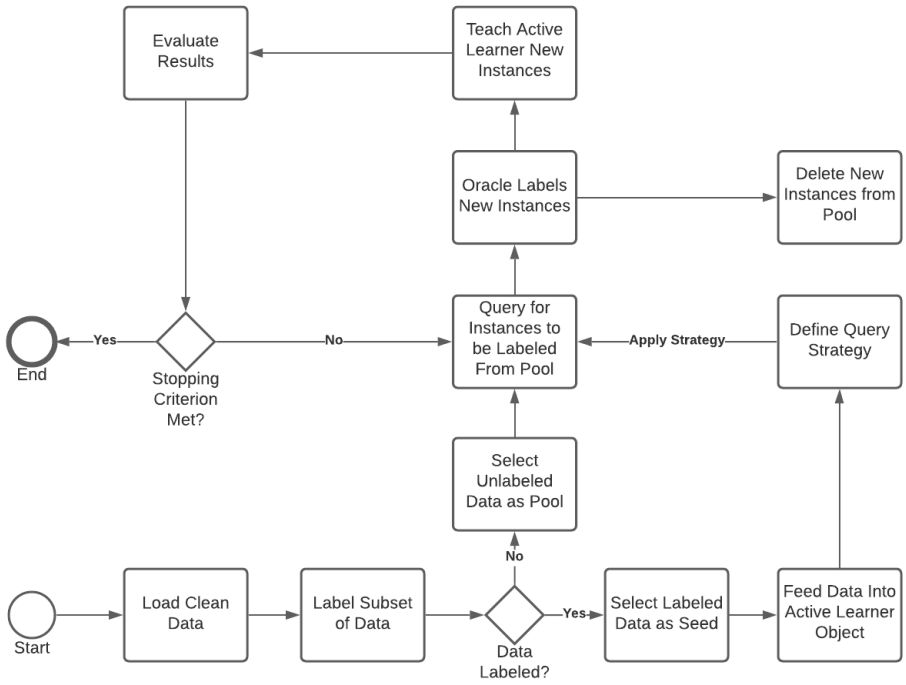

Figure 2: Visualized Workflow of an Active Learner. Created with lucid.app

4.1.3 *Sentiment Analysis Models*

The next sub-section explores the machine learning models that will be used to perform sentiment analysis on the domain-specific corpus created by the Active Learner.

Multinomial Naïve Bayes (MNB)
NB is a probabilistic supervised machine learning model. By working probabilistically, the classifier assigns the probability of belonging to a given class based on certain features (Jemai et al., 2021). Because of the high dimensionality of text data, which can be handled very well by NB, this algorithm has established itself as one of the standards for sentiment analysis. This thesis will use Multinomial Naïve Bayes to classify the sentiment of the text. This is due to the model's ability to handle larger vocabulary sizes (Abbas et al., 2019). In addition, the algorithm is simple to implement, suitable for real-time applications, and highly scalable. However, the algorithm's prediction accuracy is frequently lower than that of other sentiment analysis techniques (Song, Kim, Lee, Kim, & Youn, 2017). Due to the easy implementation and fast training of the algorithm, MNB will serve as the baseline classifier.

Long Short Term Memory (LSTM)
LSTMs are built on a recurrent neural network architecture (RNN). In an RNN the neurons are connected to themselves through time. As a result, the input from a time instance t_i will also be used as an input for the next time instance t_{i+1}. That leads to the problem of vanishing gradients, which means that it is hard for the model to learn long-term dependencies. This occurs because in a long sequence like a sentence, as the loss gradients are backpropagated through the RNN, they may shrink to zero (Ribeiro, Tiels, Aguirre, & Schön, 2020). LSTMs are designed to overcome that problem. The LSTM architecture does so via its four constituents: A memory cell which can remember a lot of information from previous states, an input gate which controls the inputs into the neurons, an output gate with an activation function and lastly a forget gate which resets the neuron (Priyantina & Sarno, 2019). When training an LSTM, it is shown that using a Word2Vec embedding can help solving the curse of dimensionalty that might occur when using a one-hot encoded input (Xiao et al., 2018).

Bidirectional Encoder Representations from Transformers (BERT)
BERT is a relatively new machine learning algorithm developed by Google in 2018 and mainly designed for natural language processing. BERT is pretrained on the English Wikipedia and BooksCorpus. Because of the pretraining users won't need as much computing power to achieve good results, even if the dataset is relatively small (Devlin et al., 2019). The BERT github page even states that "most NLP researchers will never need to pre-train their own model from scratch" (Google Research, 2020).

Valence Aware Dictionary for sEntiment Reasoning (VADER)
Due to widespread usage of lexicons, the VADER sentiment lexicon was also included. However, the results of VADER are intended for illustrative purposes only, as a lexicon based approach is not within the research scope of this thesis. VADER is specifically designed to classify social media sentiment and also includes emoticons, acronyms and *slang* words (Hutto & Gilbert, 2015). Vader looks up the associated compound score of words in the lexicon. Based on the score, the sentiment is determined. The VADER lexicon does not have any hyperparameters to tune. It is only possible to optimize results by changing classification thresholds.

4.1.4 *Evaluation Metrics for Sentiment Classification*

Typically, *accuracy, precision, recall and the F-score* are used as evaluation metrics to assess the performance of a sentiment analysis model.
Accuracy is the percentage of correctly predicted observations over all instances. Accuracy should only be used if the classes in the data are

balanced. Otherwise, a model that only predicts the majority class may be able to achieve quite high accuracy.
Precision expresses the proportion of how many classes were classified as positive, that actually are positive.
Recall refers to the percentage of total relevant results that were correctly classified.
F-score is a metric that combines precision and recall and presents the harmonic mean of the two (Hossin & Sulaiman, 2015).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + TN}$$

$$F_1 = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

In the formulas above, TP and TN are positive and negative classes that were correctly classified. Contrary, FP and FN are incorrectly classified instances.

Even though the class distribution of the data used in this thesis is skewed towards to label *bullish*, the remaining two classes still represent almost 50% of the data. As a result, accuracy can still lead to representative results, which is why this metric is used for hyperparameter tuning and model selection.

4.1.5 *Stock Price Prediction Models*

The next section explores the machine learning models that will be used to predict the stock price of GameStop.

Auto-Regressive Integrated Moving Average (ARIMA)
One of the standard methods for time series forecasting is ARIMA. Therefore, it is commonly used in forecasting stock prices. However, it has some limitations, especially with regards to nonlinearity. As a result, some prerequisites are required to create good results (Siami-Namini, Tavakoli, & Siami-Namin, 2018). First of all, ARIMA does not work well with seasonal data. Seasonality can be identified by plotting the stock prices, Autocorrelation and Partial Autocorrelation. Furthermore, ARIMA only works on stationary time-series. That can be achieved by differencing the time-series

data. By using the *augmented-dickey-fuller* test (adf), the data can be checked for stationarity (Jain & Mallick, 2017). If the null hypothesis of the test can be rejected, the time series is assumed to be stationary.

Long Short Term Memory (LSTM)
As stock prices are generally volatile, non-stationary and can have changes in their statistical properties it is beneficial to use models that can learn those attributes. Since LSTMs are able to capture such contextual information, they are shown to outperform many other methods (Preeti, Bala, & Singh, 2019). This thesis implements a Vanilla LSTM, which is a model with only a single LSTM layer. This implementation was chosen because others, such as a stacked-LSTM, do not necessarily outperform a Vanilla LSTM for stock price prediction tasks (Hai et al., 2020). Comparing other implementations would be beyond the research-scope of the thesis.

When using an LSTM for time-series tasks, a look-back period needs to be defined which chooses how many previous timesteps, for the case at hand - days, will be used to predict the stock price at the next timestep (Lim & Zohren, 2020). One advantage of LSTMs is that they can also include other features into their time-series prediction. When including other input features in addtition to the stock price, the time series is considered to be multivariate (Liu & Songcan, 2019). To answer the research question outlined in the Introduction, the obtained sentiment was added as an input feature to the model. However, since the literature also identified other features, besides sentiment, as strong predictors this thesis will also explore the performance of a model that includes trading volume and a combination of sentiment and trading volume in addition to the stock price.

4.1.6 *Evaluation Metrics for Stock Price Prediction*

Standard evaluation metrics for time-series data include Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) (Rezaei, Faaljou, & Mansourfar, 2021).
MAE computes the absolute difference between the actual and predicted price. As a result, the MAE is on the same unit scale as the output value.
MSE represents the squared distance betwen the actual and predicted prices.
RMSE simply takes the square-root of MSE. Hence, the output value is the same unit as the required output and can still represent the dispersion of results. Therefore, RMSE will be used as the evaluation metric to select the best hyperparameters and model. In the formulas y_i represents the actual value and $\hat{y}_i$ the predicted value (Chen et al., 2021). Since the ARIMA and LSTM models are on different scales, they need to be rescaled

before calculating the evaluation metrics to ensure comparable results. The aforementioned metrics are defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

4.2 *Experimental Setup*

The following subsection will explore how the methods outlined in the General Description were implemented. All the code was written in Python version 3.9.5 on a Windows machine and version 3.9.7 on a Mac in the Visual Studio Code IDE. All the packages that were installed in the virtual development environment of this thesis can be found in Appendix A.

4.2.1 *Data*

Reddit Posts
While Reddit does offer an official API, the API is most useful for streaming data. There are some strict limitations on accessing large amounts of historical data. As a result, the official API is not the best choice for this thesis. However, *pushshift.io* provides a solution for the strict limits. Essentially, Pushshift copies data from Reddit at the time it is posted. Since Pushshift uses the document-based database Elastic, it is extremely fast to query data (Brasetvik, 2015).

For this thesis all Gamestop (GME) related posts between January 1st, 2020 and October 26th, 2021 were requested for the subreddit WallStreetBets. The query returns 89 columns. Most of which, however, can be dropped since they either are not useful for this thesis or contain no data. The most important columns are the title and the content of the post. Emoticons are also included in the content text. In total 179,544 posts were obtained. Of all obtained posts, 10% or 17,955 were manually labeled as either bearish, neutral or bullish. This was done by using a graphical user interface that displayed the title and text of every tenth Reddit post. Of all manually labeled data 3479 (19.4%) are bearish, 5119 (28.5%) are neutral and 9357 (52.1%) are bullish.

GameStop Stock Price
By using the python library yfinance, stock prices for GameStop were obtained from Yahoo Finance. The start and end dates are the same as the ones used to load data from Reddit. The query returns the columns Date, Open, High, Low, Close, Adjusted Close and Volume. This thesis only uses the Date, Volume and Adjusted Close, which is the closing price of the stock after adjusting for factors such as dividend payouts. Volume refers to the number of stocks traded on a given day.

4.2.2 *Active Learner Implementation*

To implement an Active Learner the *modAL* package was used. modAL was designed with modularity, flexibility and extensibility as high priorities (Danka & Horvath, 2018). The estimator defined in the Active Learner object is a *Support Vector Machine (SVM)*. A SVM was chosen because of its strong generalization performance (Firmino, Baptista, Firmino, Oliveira, & Paiva, 2014). For the case at hand, the algorithm needs to solve a classification problem by optimally separating the data between bearish, neutral and bullish instances. Classification is done by fitting a hyper-plane with the biggest margin, meaning it looks for the greatest distance to the nearest sample points (Jemai et al., 2021). SVMs use spatial transformations, commonly known as kernel functions, to fit the hyperplane. By doing so the data is projected into a higher dimensional space, which makes them easier to separate. Kernels can be linear, RBF or others. The radial basis function (RBF) kernel is best used for non-linear problems and is a general-purpose kernel that is often used in pattern recognition problems. The linear kernel, on the other hand, is typically used when there are only two classes present. A good example for that might be positive and negative sentiment (Firmino et al., 2014).

The initial seed data to train the SVM-estimator in the Active Learner was the data that was annotated manually. Since an SVM cannot handle text data, the data had to be preprocessed and converted to a tf-idf representation. Furthermore, the implementation of the Active Learner in this thesis deviates from the literature a little bit: The literature that was reviewed does not set aside a test set from the initial seed data and the accuracy of the Active Learner is evaluated on the entire seed data after every iteration. While the literature does not explain why this approach was taken, I hypothesize that is due to the cost associated with labeling the data. This thesis will not deviate from well established machine learning practices and therefore set aside 20% of the seed data as test data, which will be used to evaluate the performance of the Active Learner after every iteration (Raschka & Mirjalili, 2019, p. 196).

Uncertainty sampling was chosen as the query strategy because it has been demonstrated to be a strong baseline strategy. This query strategy assumes, that instances that are far from the decision boundary are adequately explainable and instances close to the decision boundary are uncertain. Naturally, this complements the SVM-estimator very well. As a result, the Active Learner queries the samples about which it is most uncertain about (Osborne & Baldridge, 2004). Two human oracles labeled an additional 10% of the data, which were chosen by the Active Learner. Those 10% were chosen by the Active Learner in ten iterations, meaning the Active Learner loop was repeated every time an additional 1% of the data was labeled. Subsequently, the estimator was retrained on all labeled instances, excluding the test set.

4.2.3 *Data Preprocessing*

The research by Jemai et al. (2021) presents a system for structuring a sentiment analysis project, which was also applied in this thesis. The data collection phase is the first step, where textual data is obtained from a source. The data is then cleaned in the second step, the data pre-processing phase. To do so, several actions need to be performed. One of them is tokenization. This is a natural language processing technique in which a large body of text is broken down into multiple sentences, each of which is then broken down into a list of words. Stop words such as is, the, a and other common words are also removed during the pre-processing phase. If stop words are included, they may play a negative role in sentiment classification and increase the overall vocabulary size while having little predictive power. (Zhao & Gui, 2017). In addition, special characters such as @ and urls should also be removed. It is also suggested that the text is converted to lowercase. As the final step, the research proposes lemmatization. By doing so, the structure of a word is analyzed and converted to its normalized form. The research conducted by Camacho-Collados and Pilehvar (2018) shows that lemmatization improves sentiment analysis results especially when using domain-specific datasets.

Since it is shown that having data with emoticons leads to more accurate results than data without emoticons, this thesis does not remove emoticons from the text corpus (Parveen & Pandey, 2016).

4.2.4 *Sentiment Analysis Models Implementation*

The next section explores how the machine learning models to classify sentiment were implemented and how their optimal hyperparameters were chosen. Before training the models, 20% of the data were set aside as the test set. By setting aside 20% of the training data as a validation set the

optimal hyperparameters were selected. To account for class imbalances, stratification was applied. As a result, all sets have approximately the same class distribution as the full set (Sahu, Mukhopadhyay, Szengel, & Zachow, 2017). Stratification was chosen over other methods to handle class imbalances, such as over- or undersampling, because the imbalance is not too extreme (Ganganwar, 2012).

Multinomial Naïve Bayes (MNB)
To train the classifier, the data was first converted to a tf-idf representation. The classifier uses five-fold gridsearch cross-validation to find the optimal parameters for *fit_prior*, which determines if prior class probabilities shall be learned, and *alpha*, which is a smoothing parameter that solves the problem of zero probability. That problem might occur, if a an unseen word appears in the test set. By setting alpha to a value greater than zero, the model pretends to have seen a word before.

Long Short Term Memory (LSTM)
Before training the LSTM, data is first fed into a Word2Vec model to learn the word embeddings. As explained in the literature, this can enhance the performance of the model by learning the similarity between words. Selecting optimal hyperparameters for the Word2Vec model is oftentimes neglected, even though that may lead to performance differences. The Word2Vec hyperparameters that were analyzed is the *vector_size, min_count* and *window*. The optimal respective hyperparameters are 50, 1 and 1. Those were determined based on a comparative intrinsic evaluation (Schnabel, Labutov, Mimno, & Joachims, 2015). A subset of the most similar words, calculated as the cosine similarity, of given words can be found in Appendix B. The input and output dimensions of the Embedding layer of the LSTM, as well as the weights are taken from the word2vec model. The output dimensions of the embedding layer are also used as the units for the subsequent LSTM layer. Furthermore, the model adds a dropout layer to improve generalization. To find the optimal *dropout* parameter and *optimizer* for the model, a loop runs through a set of hyperparameters when fitting the model. The optimal model is determined by evaluating the performance on the validation set, which is 20% of the training data. The final Dense output layer uses softmax as its activation function, which is typically used for multiclass classification. Furthermore, the model uses categorical crossentropy as its cost function and accuracy as its metric.

Bidirectional Encoder Representations from Transformers (BERT)
The BERT model used in this thesis uses the *bert base uncased* implementation, which has 12 encoders, 12 self-attention heads and 110 million

parameters (Devlin et al., 2019). Even though a maximum of 512 tokens can be used when training BERT, this implementation only uses 64 tokens because of computational reasons. For hyperparameter tuning the *optimizer* and its *learning rate*, which is used to find the minimum of the loss function, are analyzed. The model is trained with a batch size of two over three epochs. Typically, BERT does not show performance increases after more than three epochs.

Valence Aware Dictionary for sEntiment Reasoning (VADER)
VADER classifies data based on a threshold value. If the score is greater than or equal to the threshold, it is associated with positive sentiment. If the score is smaller than or equal to the threshold it is associated with negative sentiment. For scores in between, neutral sentiment is assigned (Hutto & Gilbert, 2015). To obtain optimal evaluation metrics varying thresholds were analyzed.

4.2.5 *Stock Price Prediction Implementation*

When splitting time series data, it needs to be split into windows. In time series data, the sequence of the values is important. Therefore, it cannot be randomly split (LeBaron & Weigend, 1998). The training set consists of the first 80% of the timeseries data and the test set of the remainding 20% of the data. Furthermore, 20% of the training set is set aside as the validation set. To select the optimal hyperparameters, a rolling forecast is implemented. By doing so, the model is first trained on the train set and then predicts the first timestep of the validation set. After that, the predicted value can be compared to the actual value of the validation set at the given timestep. Subsequently, the actual value is added to the train set and the process is repeated. (Siami-Namini et al., 2018). To test the generalization performance, however, no rolling window approach is applied on the test set. Instead, the model will predict the entire test set based on the parameters it has learned before. By doing so, it is tested how well it would perform in a real world scenario.

ARIMA
First the stock price data was analyzed to identify a seasonality pattern and to check for stationarity. By visualizing the stock price, no seasonality was identified. This makes sense, because the data does not contain many years and hence does not contain any repeating patterns. After differencing the time-series once the dickey-fuller test was applied, which showed stationarity at the 1% significance level. The hyperparameters that can be optimized are the number of lag observations included in the model (p), the number of times the data shall be differenced (d) and the size of the

moving average window (q) (Siami-Namini et al., 2018).

LSTM
The data fed into the LSTM, in comparison to the ARIMA model, was not differenced as one of the strengths of an LSTM is that it can also work well with non-stationary data. As the first step, the input data was scaled to be between 0 and 1. This is standard practice and commonly applied in the literature. As a result, the LSTM is expected to be less sensitive to the scale of the input data. The LSTM looks for the optimal hyperparameters of the optimizer, the lookback period, and the dropout on the input which excludes a random subset of the data from the node activation and update of the weights. A lookback period between one and five days was chosen, because Saud and Shakya (2020) found that a suitable lookback period for LSTMs is less than 5. In addition to the price, trading volume and sentiment were also included as scaled input features. For the model that also includes sentiment, however, some additional preprocessing steps were required. That is because sentiment is measured for every post, of which there can be multiple per day, whereas there is only one time-series observation per day. Therefore, sentiment was encoded as -1 for bearish, 0 for neutral and 1 for bullish. Then the sentiment was summed up and added to the relevant date. By doing so, the collective sentiment of the entire WallStreetBets forum on a given day can be included. If, for example, more people are bullish that leads to a higher collective sentiment score. Whereas, if more people are pessimistic about the stock, that leads to a lower score.

5 RESULTS

This section is broken down into three parts. First the results of the Active Learner are looked at, then the sentiment models and lastly the stock price prediction models. If the visualization uses a horizontal bar chart, the best result is the one on top. If the visualization contains multiple line charts, the best result is drawn as a solid line and the rest as dashed lines.

5.1 *Active Learner Results*

The Active Learner is represented as a line chart with the accuracy on the y-axis and the respective query instance on the x-axis. The accuracy at query-instance 0 is the performance after the initial training on the seed data. As can be seen in Figure 3, the performance of the Active Learner hit a plateau relatively quickly. It is hypothesized that this plateau was reached for two reasons: First, by labeling the data at the document

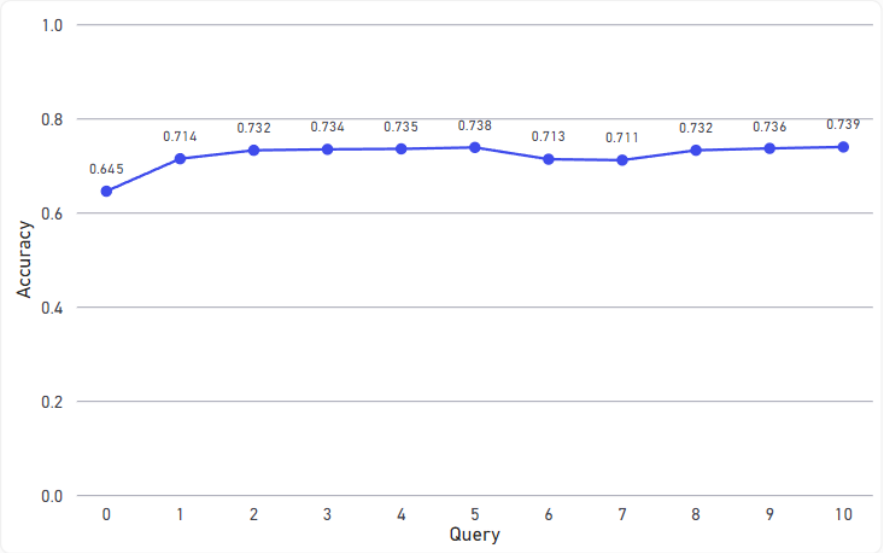

Figure 3: Accuracy of Active Learner over 10 Query Instances

level, a document can contain bearish, neutral and bullish words. Those contradictions may not only make it hard for the model to learn the structure of the input, but also for the human annotator when labeling the data. Other methods, such as part-of-speech tagging may lead to more accurate annotations and hence better results. Additionally, by having two human oracles labeling the data, inconsistencies seem to have occured.

5.2 *Sentiment Models Results*

The best performing model is selected based on the validation set and will subsequently be used to determine the sentiment for the rest of the dataset. The best performing hyperparameter-set of the NB classifier achieved accuracy of 0.80, the LSTM 0.89 and BERT 0.90.

MNB
Since the Naive Bayes model does not train over epochs, its results are visualized on a bar chart. As can be seen in Figure 4 the highest accuracy of 80.4% can be achieved by setting the hyperparameters alpha to one and fit_prior to False. This baseline estimator performs much better, than just randomly guessing the majority class (bullish).

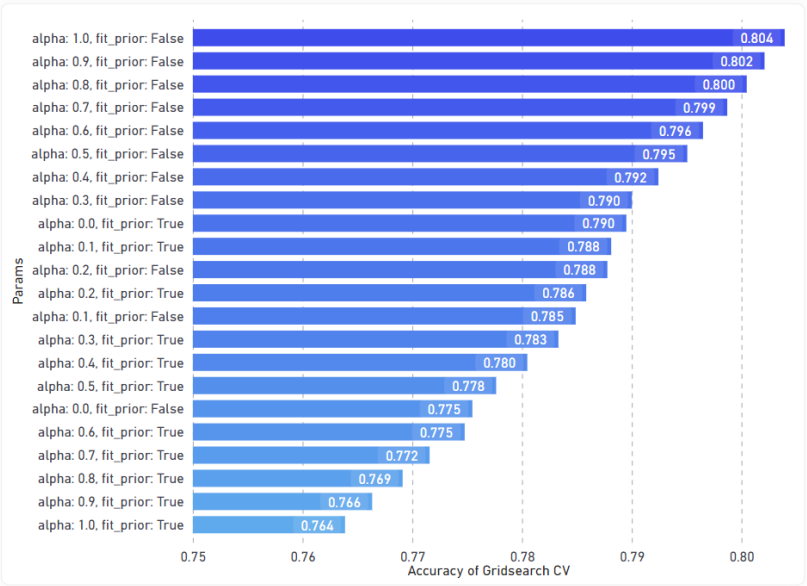

Figure 4: Accuracy of NB With Varying Hyperparameters

LSTM

The LSTM finds its optimal hyperparameters with an accuracy of 89% after three epochs, as can be seen in Figure 5. After three epochs the model seems to start overfitting a little bit. This can be concluded, as the validation set accuracy starts dropping, while the training set accuracy keeps improving. The same can be argued for the loss. The accuracy and loss for the trainining and validation set can be found in Appendix C. The optimal hyperparameters are a dropout rate of zero and the Adam optimizer.

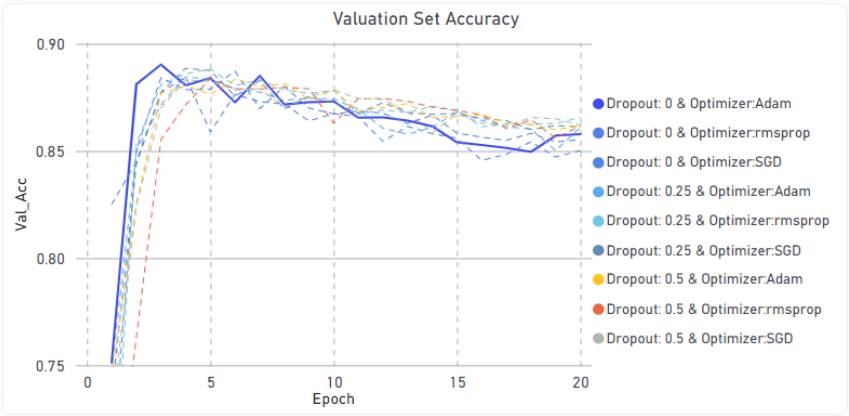

Figure 5: Accuracy of LSTM Validation Set With Varying Hyperparameters

BERT

Since BERT is already pre-trained, it does not need as many epochs to optimize a model. As can be seen in Figure 6, there is almost no change to the accuracy, meaning the model does not benefit from training over more epochs. As the literature suggests, even though BERT lacks domain awareness it managed to show better accuracy than other machine learning models, albeit by a small margin. The accuracy and loss for the trainining and validation set can be found in Appendix D. The optimal hyperparameters that were found are an AdamW optimizer with a learning rate of 0.00001.

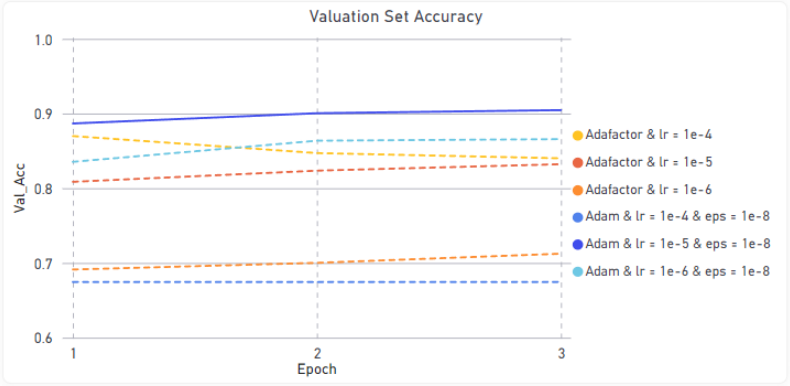

Figure 6: Accuracy of BERT Validation Set With Varying Hyperparameters

Based on the validation accuracy BERT was chosen to label the rest of the data. To compare how well the models generalize, the evaluation metrics of the testsets are outlined in Table 1. All models seem to generalize very well and their accuracy on the test set is only marginally lower than on the validation set. Again, BERT shows better performance than the LSTM and NB, even though their metrics also show satisfactory results. VADER, however, does not perform very well. With regards to predicting the classes, all models achieve high Precision, Recall and F_1-Score when predicting the target bullish. While both BERT and the LSTM also do relatively well on other classes, MNB does not show as strong results. Interestingly, however, the LSTM achieves better metrics for some classes than BERT. How well certain classifiers do on specific classes can be found in Appendix E.

Table 1: Test Set Macro Average Evaluation Metrics Models

	Evaluation Metrics			
Model	Accuracy	Precision	Recall	F_1-Score
MNB	0.80	0.72	0.79	0.75
LSTM	0.86	0.86	0.85	0.85
BERT	**0.89**	**0.87**	**0.85**	**0.86**
VADER	0.39	0.38	0.39	0.38

5.3 *Stock Prediction Evaluation*

The best performing stock price prediction model and its hyperparameters was selected based on the RMSE of the validation set. As can be seen in Table 2, the LSTM that only used Price as its input feature achieved the lowest RMSE for a given set of hyperparameters. Figure 7 shows the RMSE of the validation set for each LSTM model's respective input features with their optimal hyperparameter setting. The RMSE and Loss of the training and validation set can be found in Appendix F.

Table 2: Validation Set RMSE for ARIMA and LSTM Time-Series Models

Model	RMSE
ARIMA	51.96
LSTM including Price, Trading Volume and Sentiment	29.53
LSTM including Price and Sentiment	25.29
LSTM including Price and Trading Volume	33.40
LSTM that only includes Price	**24.11**

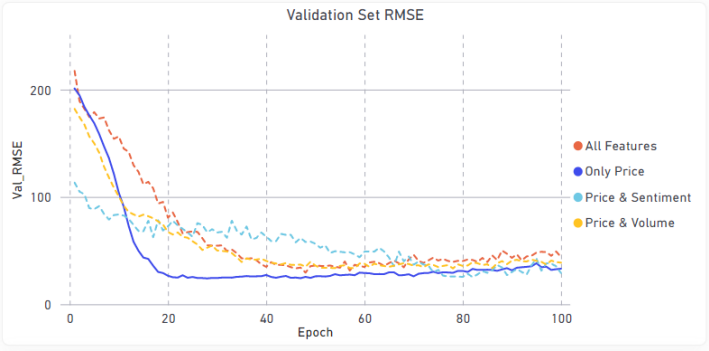

Figure 7: Validation Set RMSE of Models with Various Input Features and Their Optimal Hyperparameters

The RMSE of the best performing model on the validation set are presented in Figure 8. As can be seen, the lowest loss can be achieved at epoch 28 with the hyperparameter setting Dropout equals zero, an Adam optimizer and lookback equals one. Since too many hyperparameters were analyzed, only the five best performing ones were visualized. The RMSE and Loss of the model can be found in Appendix G and the models with other input features can be found in Appendix H.

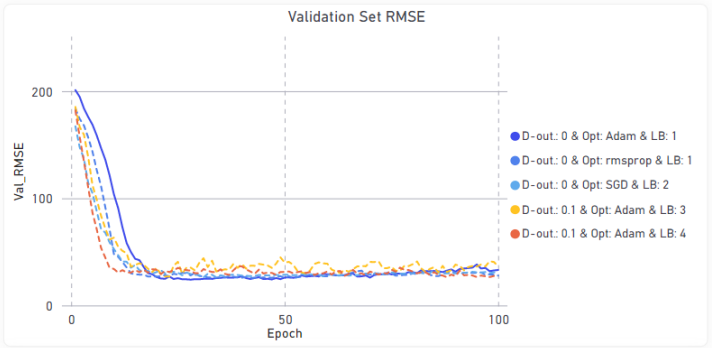

Figure 8: RMSE of Five Optimal Hyperparameters of LSTM Model With Only Price as Input Feature

Because ARIMA is not trained over epochs, it is visualized as a bar-chart in Figure 9. As can be seen, the optimal hyperparameter setting is (0, 0, 1).

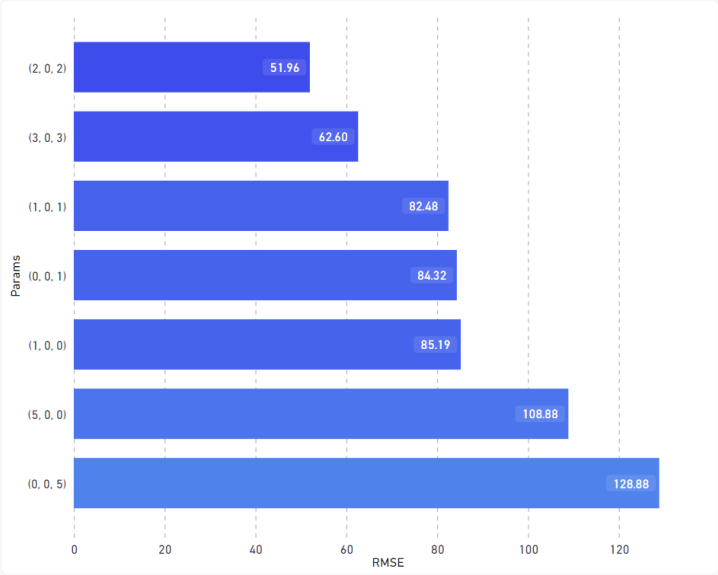

Figure 9: RMSE of ARIMA

To see how well the different stock price prediction models generalize, they were evaluated based on their test set. Because the time series training, validation and test sets are split into windows, they do not necessarily share the same characteristics. As can be seen in Table 3, the RMSE of most models is lower than that of the validation set. This can be explained by the circumstance that the test set does not contain as volatile periods as the validation set.

Table 3: Test Set Evaluation Metrics for ARIMA and LSTM Time-Series Models

	Evaluation Metrics		
Model	RMSE	MSE	MAE
ARIMA	80.00	6,399.72	75.96
LSTM including Price, Trading Volume and Sentiment	9.14	83.60	6.39
LSTM including Price and Sentiment	10.47	109.62	8.54
LSTM including Price and Trading Volume	**8.07**	**65.12**	**5.52**
LSTM that only includes Price	11.29	127.40	9.76

While the LSTM seems to understand the strucut of the time-series and hence achieve good evaluation metrics, the ARIMA model overfit a lot. While the LSTM that only includes price as its input feature was selected as the preferred model based on the validation set, the LSTMs that include other features as well achieve better metrics on the test set. However, the generalization performance of the model that only includes price still seems satisfactory. This can also be concluded when plotting the LSTM that only includes price as its input. As can be seen in 10 the model generally gets the price direction correct, even though it does not overlap with the actual stock price at most timesteps. Additionally, the prediction was able to identify when the direction of the stock price changes to both the up- and downside. The price forecasts of the other models are represented in Appendix I.

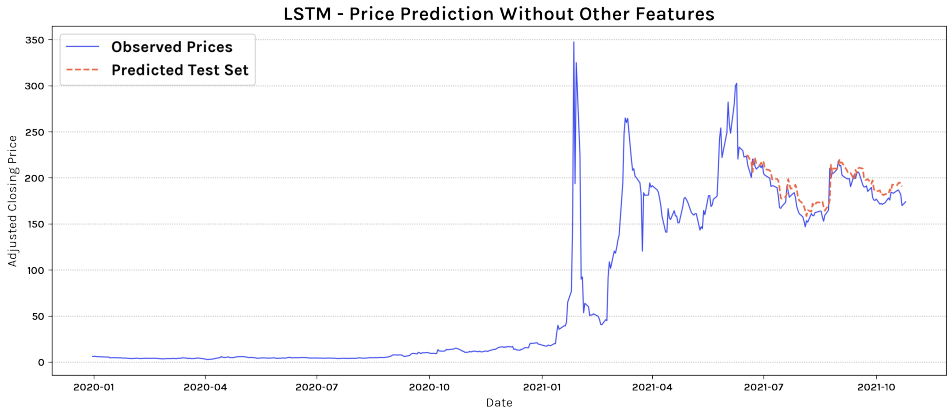

Figure 10: Price Forecast of Selected Model from Validation Set

6 DISCUSSION

As the 'to the moon' WallStreetBets movement had and will have a tremendous impact on the financial markets and the lives of individuals, it is important to accurately measure and monitor the eponymous subreddit. As a result, disruptions to the financial markets may be avoided and stability ensured. The underlying hypothesis of this thesis is that by accurately measuring the sentiment of the forum changes in the Gamestop stock price can be predicted. This is not without its own challenges, however.

First of all, to the best of my knowledge there are no datasets available that contain labeled data of the forum. It is argued that having a labeled dataset that is tailored to a specific domain generally leads to better results *(reference to literature)*. Therefore, this thesis proposes an Active Learner to annotate the data for a fraction of the total cost. However, the accuracy of the Active Learner was not as high as expected. This can be attributed to the ambiguity of forum-posts at the document level. Even when manually labeling the data it was not always entirely clear how to appropriately classify it. While the advantages of an Active Learner still seem promising, a different labeling approach, such as part-of-speech tagging or named entity recognition might be a better solutions than simply labeling the data on the document level *(reference to literature)*.
After the ground truth data is created, it can be used by machine learning models which have demonstrated to perform well on sentiment classification tasks *(reference to literature)*. The literature, in contrast, usually uses a lexicon based approach to classify the sentiment of WallStreetBets. However, to the best of my knowledge the domain-specific terminology used on the forum is currently not covered by any sentiment lexicas, which may lead to incorrect conclusions.

This thesis shows that the general purpose VADER lexicon is not suitable for sentiment classification on WallStreetBets. In contrast, a traditional machine learning model such as Naive Bayes or deep learning approaches such as BERT or LSTMs show much better performance. While BERT produces the best evaluation metrics, the LSTM only performs marginally worse. Further research, however, is needed on why the machine learning models performed that well even though the Active Learner did not deliver highly accurate ground truth data. It could be argued that the Active Learner did in fact deliver good results, but that the test set used contained many complex instances. Additionally, one reason may be that the test set was simply too small and did not reflect the true properties of the dataset. Of course, it can also be argued that the commonly applied approach in the literature of merging the train and test set at every iteration of the Active Learner is better than the approach chosen in this thesis, which is having the test set set aside initially. Answering these questions, however, is subject for future research. Nonetheless, arguing that the models overfit would in my opinion not be correct, because all models show good results on the test set. For all models the accuracy on the test set is only slightly lower than on the validation set, which would make overfitting unlikely.

Once accurate sentiment is obtained it can be tested whether or not it adds to predictive capabilities with regards to time-series forecasting. This thesis shows, that while the results seem promising when including sentiment as an input feature, it does not necessarily outperform a model that only includes price to forecast stock prices. One possible explanation for that is the Efficient Market Hypothesis, which states that all publicly available information and expectations are already priced into the stock price (Fama, 1970). However, it needs to be noted that this thesis only analyzed the impact on the subsequent day which gives the market enough time to incorporate the sentiment of the forum. The immediate influence of sentiment on changes in the stock price may be more pronounced.

7 CONCLUSION

Yet to be written...

REFERENCES

Abbas, M., Ali, K., Memon, S., Jamali, A., Memon, S., & Ahmed, A. (2019, 03). *Multinomial naive bayes classification model for sentiment analysis.* doi: 10.13140/RG.2.2.30021.40169

Anand, A., & Pathak, J. (2021). Wallstreetbets against wall street: The role of reddit in the gamestop short squeeze. *Indian Institute of Management Bangalore Research Paper Series*.

Antweiler, W., & Frank, M. (2004, 02). Is all that talk just noise? the information content of internet stock message boards. *Journal of Finance*, *59*, 1259-1294. doi: 10.2139/ssrn.282320

Arora, S., Nyberg, E., & Rose, C. (2009, 01). Estimating annotation cost for active learning in a multi-annotator environment. *HLT-NAACL*. doi: 10.3115/1564131.1564136

Baldridge, J., & Osborne, M. (2004, jul). Active learning and the total cost of annotation. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (p. 9-16). Barcelona, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W04-3202

Binu, M., & Sony, G. (2020). Dimensionality reduction and visualisation of hyperspectral ink data using t-sne. *Forensic Science International*, *311*, 110194. Retrieved from https://www.sciencedirect.com/science/article/pii/S0379073820300566 doi: https://doi.org/10.1016/j.forsciint.2020.110194

Brasetvik, A. (2015, February 15). *Uses of elasticsearch, and things to learn.* Web. Retrieved from https://www.elastic.co/blog/found-uses-of-elasticsearch

Caginalp, G., & Constantine, G. M. (1995). Statistical inference and modelling of momentum in stock prices. *Applied Mathematical Finance* 2, 225-242.

Camacho-Collados, J., & Pilehvar, M. T. (2018). *On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis.*

Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean-variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, *100*, 106943. Retrieved from https://www.sciencedirect.com/science/article/pii/S1568494620308814 doi: https://doi.org/10.1016/j.asoc.2020.106943

Danbolt, J., Siganos, A., & Vagenas-Nanos, E. (2015). Investor sentiment and bidder announcement abnormal returns. *Journal of Corporate Finance*, 164-179.

Danka, T., & Horvath, P. (2018). *modal: A modular active learning framework for python.*

Das, S. R., & Chen, M. Y. (2007). Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 1375-1388.

De Gooijer, J. G., & Hyndman, R. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, *22*, 443-473. doi: 10.1016/j.ijforecast.2006.01.001

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai, L., & Li, F.-F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 ieee conference on computer vision and pattern recognition* (p. 248-255). doi: 10.1109/CVPR.2009.5206848

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding.*

Diangson, B., & Jung, N. (2021). *Bet it on reddit: The effects of reddit chatter on highly shorted stocks.*

Du, C., Sun, H., Wang, J., Qi, Q., & Liao, J. (2020). Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Acl.*

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383-417. Retrieved from http://www.jstor.org/stable/2325486

Feng, X., & Johansson, A. C. (2019). Top executives on social media and information in the capital market: Evidence from china. *Journal of Corporate Finance*, *58*, 824-857. Retrieved from https://www.sciencedirect.com/science/article/pii/S0929119918303225 doi: https://doi.org/10.1016/j.jcorpfin.2019.04.009

Firmino, A., Baptista, C., Firmino, A., Oliveira, M., & Paiva, A. (2014). A comparison of svm versus naive-bayes techniques for sentiment analysis in tweets: A case study with the 2013 fifa confederations cup. In *Proceedings of the 20th brazilian symposium on multimedia and the web* (p. 123–130). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi-org.tilburguniversity.idm.oclc.org/10.1145/2664551.2664561 doi: 10.1145/2664551.2664561

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, *2*(4), 42-47.

Gennaro, G., Buonanno, A., & Palmieri, F. (2021, 11). Considerations about learning word2vec. *The Journal of Supercomputing*, *77*. doi: 10.1007/s11227-021-03743-2

Google Research. (2020, March 11). *bert.* Retrieved from https://github.com/google-research/bert

Gu, C., & Kurov, A. (2020). Informational role of social media: Evidence

from twitter sentiment. *Journal of Banking and Finance*, *121*, 105969. Retrieved from https://www.sciencedirect.com/science/article/pii/S0378426620302314 doi: https://doi.org/10.1016/j.jbankfin.2020.105969

Guia, M., Silva, R. R., & Bernardino, J. (2019). Comparison of naive bayes, support vector machine, decision trees and random forest on sentiment analysis. In *Kdir.*

Hai, P. N., Tien, N. M., Hieu, H. T., Chung, P. Q., Son, N. T., Ha, P. N., & Son, N. T. (2020). An empirical research on the effectiveness of different lstm architectures on vietnamese stock market. In *2020 international conference on control, robotics and intelligent system* (p. 144-149). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3437802.3437827 doi: 10.1145/3437802.3437827

Hochreiter, S., & Schmidhuber, J. (1997, 12). Long short-term memory. *Neural computation*, *9*, 1735-80. doi: 10.1162/neco.1997.9.8.1735

Hossin, M., & Sulaiman, M. N. (2015, 03). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining and Knowledge Management Process*, *5*, 1-11.

Hutto, C., & Gilbert, E. (2015, 01). Vader: A parsimonious rule-based model for sentiment analysis of social media text..

Ivanovic, Z., Bogdan, S., & Baresa, S. (2013, 06). Forecasting croatian stock market index: Crobex. *UTMS Journal of Economics*, *4*(2), 79-91. Retrieved from https://www.proquest.com/scholarly-journals/forecasting-croatian-stock-market-index-crobex/docview/1399281625/se-2

Jain, G., & Mallick, B. (2017). A study of time series models arima and ets.

Jemai, F., Hayouni, M., & Baccar, S. (2021). Sentiment analysis using machine learning algorithms. *International Wireless Communications and Mobile Computing*, 775-779.

Jin, Z., Yang, Y., & Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and lstm. *Neural Computing and Applications*, *32*. doi: 10.1007/s00521-019-04504-2

Jin-Dong, K., Tomoko, O., & Junichi, T. (2008, 02). Corpus annotation for mining biomedical events from lterature. *BMC bioinformatics*, *9*, 10. doi: 10.1186/1471-2105-9-10

Kotelnikova, A., Paschenko, D., Bochenina, K., & Kotelnikov, E. (2021). *Lexicon-based methods vs. bert for text sentiment analysis.*

LeBaron, B., & Weigend, A. (1998). A bootstrap evaluation of the effect of data splitting on financial time series. *IEEE Transactions on Neural Networks*, *9*(1), 213-220. doi: 10.1109/72.655043

Lim, B., & Zohren, S. (2020, Feb). Time-series forecasting with deep

learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *379*(2194). Retrieved from http://dx.doi.org/10.1098/rsta.2020.0209 doi: 10.1098/rsta.2020.0209

Liu, J., & Songcan, C. (2019). Non-stationary multivariate time series prediction with selective recurrent neural networks. In A. C. Nayak & A. Sharma (Eds.), *Pricai 2019: Trends in artificial intelligence* (p. 636-649). Cham: Springer International Publishing.

Long, C., Lucey, B. M., & Yarovaya, L. (2021). 'i just like the stock' versus 'fear and loathing on main street': The role of reddit sentiment in the gamestop short squeeze. *SSRN Electronic Journal*.

Lu, J., Henchion, M., & Namee, B. M. (2019). *Investigating the effectiveness of representations based on word-embeddings in active learning for labelling text datasets.*

Lyócsa, S., Baumöhl, E., & Vyrost, T. (2021). Yolo trading: Riding with the herd during the gamestop episode. *Finance Research Letters*.

Miller, B., Linder, F., & Mebane, W. R. (2020). Active learning approaches for labeling text: Review and assessment of the performance of active learning approaches. *Political Analysis*, *28*(4), 532–551. doi: 10.1017/pan.2020.4

Muhammad, A. (2014, 05). Detection and scoring of internet slangs for sentiment analysis using sentiwordnet. *Life Science Journal*, *11*, 66-72. doi: 10.6084/M9.FIGSHARE.1609621

Namcheol, J., & Ghang, L. (2019, 04). Automated classification of building information modeling (bim) case studies by bim use based on natural language processing (nlp) and unsupervised learning. *Advanced Engineering Informatics*, *41*. doi: 10.1016/j.aei.2019.04.007

Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019, 12). Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, *26*. doi: 10.1002/isaf.1459

Osborne, M., & Baldridge, J. (2004, 01). Ensemblebased active learning for parse selection. In (p. 89-96).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, *2*, 1-135. Retrieved from https://doi.org/10.1561/1500000011 doi: 10.1561/1500000011

Park, S., Lee, W., & Moon, I.-C. (2015). Efficient extraction of domain specific sentiment lexicon with active learning. *Pattern Recognition Letters*, *56*, 38-44.

Parveen, H., & Pandey, S. (2016, 01). Sentiment analysis on twitter dataset using naive bayes algorithm. In (p. 416-419). doi: 10.1109/ICATCCT.2016.7912034

Piñeiro-Chousa, J., Vizcaino-Gonzalez, M., & Perez-Pico, A. M. (2017). Influence of social media over the stock market. *Psychology and Marketing*, *34*, 101-108. doi: 10.1002/mar.20976

Preeti, Bala, R., & Singh, R. P. (2019). Financial and non-stationary time series forecasting using lstm recurrent neural network for short and long horizon. In *2019 10th international conference on computing, communication and networking technologies (icccnt)* (p. 1-7). doi: 10.1109/ICCCNT45670.2019.8944624

Priyantina, R., & Sarno, R. (2019, 06). Sentiment analysis of hotel reviews using latent dirichlet allocation, semantic similarity and lstm. *International Journal of Intelligent Engineering and Systems*, *12*, 142-155. doi: 10.22266/ijies2019.0831.14

Rammurthy, S. K., & Patil, S. B. (2021). An lstm-based approach to predict stock price movement for it sector companies. *International Journal of Cognitive Informatics and Natural Intelligence*, *15*, 1-12.

Raschka, S., & Mirjalili, V. (2019). *Python machine learning*. Packt Publishing.

Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications*, *169*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417420310228 doi: https://doi.org/10.1016/j.eswa.2020.114332

Ribeiro, A. H., Tiels, K., Aguirre, L. A., & Schön, T. (2020, 26–28 Aug). Beyond exploding and vanishing gradients: analysing rnn training using attractors and smoothness. In S. Chiappa & R. Calandra (Eds.), *Proceedings of the twenty third international conference on artificial intelligence and statistics* (Vol. 108, pp. 2370–2380). PMLR. Retrieved from https://proceedings.mlr.press/v108/ribeiro20a.html

Sahu, M., Mukhopadhyay, A., Szengel, A., & Zachow, S. (2017, 03). Addressing multi-label imbalance problem of surgical tool detection using cnn. *International journal of computer assisted radiology and surgery*, *12*. doi: 10.1007/s11548-017-1565-x

Salah, R., & Gayar, N. E. (2019). *Sentiment analysis using unlabeled email data.* EasyChair Preprint no. 2080.

Saud, A. S., & Shakya, S. (2020). Analysis of look back period for stock price prediction with rnn variants: A case study on banking sector of nepse. *Procedia Computer Science*, *167*, 788-798. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050920308851 (International Conference on Computational Intelligence and Data Science) doi: https://doi.org/10.1016/j.procs.2020.03.419

Sazzed, S., & Jayarathna, S. (2021). Ssentia: A self-supervised sentiment analyzer for classification from unlabeled data. *Machine Learning with*

Applications, *4*, 100026. Retrieved from https://www.sciencedirect.com/science/article/pii/S2666827021000074 doi: https://doi.org/10.1016/j.mlwa.2021.100026

Schnabel, T., Labutov, I., Mimno, D., & Joachims, T. (2015, 01). Evaluation methods for unsupervised word embeddings. In (p. 298-307). doi: 10.18653/v1/D15-1036

Schöneburg, E. (1990). Stock price prediction using neural networks: A project report. *Neurocomputing*, *2*(1), 17-27. Retrieved from https://www.sciencedirect.com/science/article/pii/092523129090013H doi: https://doi.org/10.1016/0925-2312(90)90013-H

Semenova, V., & Winkler, J. (2021). *Reddit's self-organised bull runs: Social contagion and asset prices.*

Settles, B. (2009). Active learning literature survey..

Sharif, O., Hoque, M. M., & Hossain, E. (2019). Sentiment analysis of bengali texts on online restaurant reviews using multinomial naïve bayes. In *2019 1st international conference on advances in science, engineering and robotics technology (icasert)* (p. 1-6). doi: 10.1109/ICASERT.2019.8934655

Shetty, D. K., & Ismail, B. (2021). Forecasting stock prices using hybrid non-stationary time series model with ernn. *Communications in Statistics - Simulation and Computation*, *0*(0), 1-13. Retrieved from https://doi.org/10.1080/03610918.2021.1872631 doi: 10.1080/03610918.2021.1872631

Siami-Namini, S., Tavakoli, N., & Siami-Namin, A. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th ieee international conference on machine learning and applications (icmla)* (p. 1394-1401). doi: 10.1109/ICMLA.2018.00227

Song, J., Kim, K., Lee, B., Kim, S., & Youn, H. Y. (2017). A novel classification approach based on naïve bayes for twitter sentiment analysis. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, 2996-3011.

Sugitomo, J. C., Kevin, N., Jannatri, N., & Suhartono, D. (2021). Sentiment analysis using svm and naïve bayes classifiers on restaurant review dataset. In *2021 1st international conference on computer science and artificial intelligence (iccsai)* (Vol. 1, p. 100-108). doi: 10.1109/ICCSAI53272.2021.9609776

Susanti, A., Djatna, T., & Kusuma, W. (2017, 09). Twitter's sentiment analysis on gsm services using multinomial naïve bayes. *Telkomnika (Telecommunication Computing Electronics and Control)*, *15*, 1354-1361. doi: 10.12928/TELKOMNIKA.v15i3.4284

Tomasev, N., Radovanovic, M., Mladenic, D., & Ivanovic, M. (2014). The

role of hubness in clustering high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, *26*(3), 739-751. doi: 10.1109/TKDE.2013.25

Vuong, P. H., Dat, T. T., Mai, T. K., Uyen, P. H., & Bao, P. T. (2021). Stock-price forecasting based on xgboost and lstm. *Computer Systems Science and Engineering*, *40*, 237-246. Retrieved from http://www.techscience.com/csse/v40n1/44219 doi: 10.32604/csse.2022.017685

Wang, H., Guo, Z., & Chen, L. (2019). Financial forecasting based on lstm and text emotional features. In *2019 ieee 8th joint international information technology and artificial intelligence conference (itaic)* (p. 1427-1430). doi: 10.1109/ITAIC.2019.8785505

Wang, S., & Manning, C. (2012, 07). Baselines and bigrams: Simple, good sentiment and topic classification. In (p. 90-94).

Xianghua, F., Jingying, Y., Jainqiang, L., Min, F., & Huihui, W. (2018). Lexicon enhanced lstm with attention for general sentiment analysis. *IEEE Access*, 71884-71891.

Xiao, L., Wang, G., & Zuo, Y. (2018). Research on patent text classification based on word2vec and lstm. In *2018 11th international symposium on computational intelligence and design (iscid)* (Vol. 01, p. 71-74). doi: 10.1109/ISCID.2018.00023

Yanyan, W., Fulian, Y., Jianbo, L., & Marco, T. (2020, 08). Automatic construction of domain sentiment lexicon for semantic disambiguation. *Multimedia Tools and Applications*, *79*. doi: 10.1007/s11042-020-09030-1

Yue, L., Malu, C., Umeshwar, D., & ChengXiang, Z. (2011). Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th international conference on world wide web* (p. 347-356). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi-org.tilburguniversity.idm.oclc.org/10.1145/1963405.1963456 doi: 10.1145/1963405.1963456

Zaghum, U., Mariya, G., Imran, Y., & Shoaib, A. (2021). A tale of company fundamentals vs sentiment driven pricing: The case of gamestop. *Journal of Behavioral and Experimental Finance*.

Zhao, J., & Gui, X. (2017, 02). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, *PP*, 1-1. doi: 10.1109/ACCESS.2017.2672677

Zhengqi, P., Zhewei, S., & Yang, X. (2019, November). Slang detection and identification. In *Proceedings of the 23rd conference on computational natural language learning (conll)* (pp. 881–889). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://

aclanthology.org/K19-1082 doi: 10.18653/v1/K19-1082

8 APPENDICES

8.1 *Appendix A*

List of packages used.

8.2 *Appendix B*

Example of the similarity (denoted in the parentheses) of the three most similar words of an input word:
robinhood -> rh (0.98), etrade (0.87), webull (0.86)
andromeda -> jupiter (0.95), mars (0.94), uranus (0.93)
ape -> autist (0.94), monkey (0.91), retard (0.90)
hedgefund -> hfs (0.93), hf (0.89), shorter (0.88)

8.3 *Appendix C*

This is appendix C.

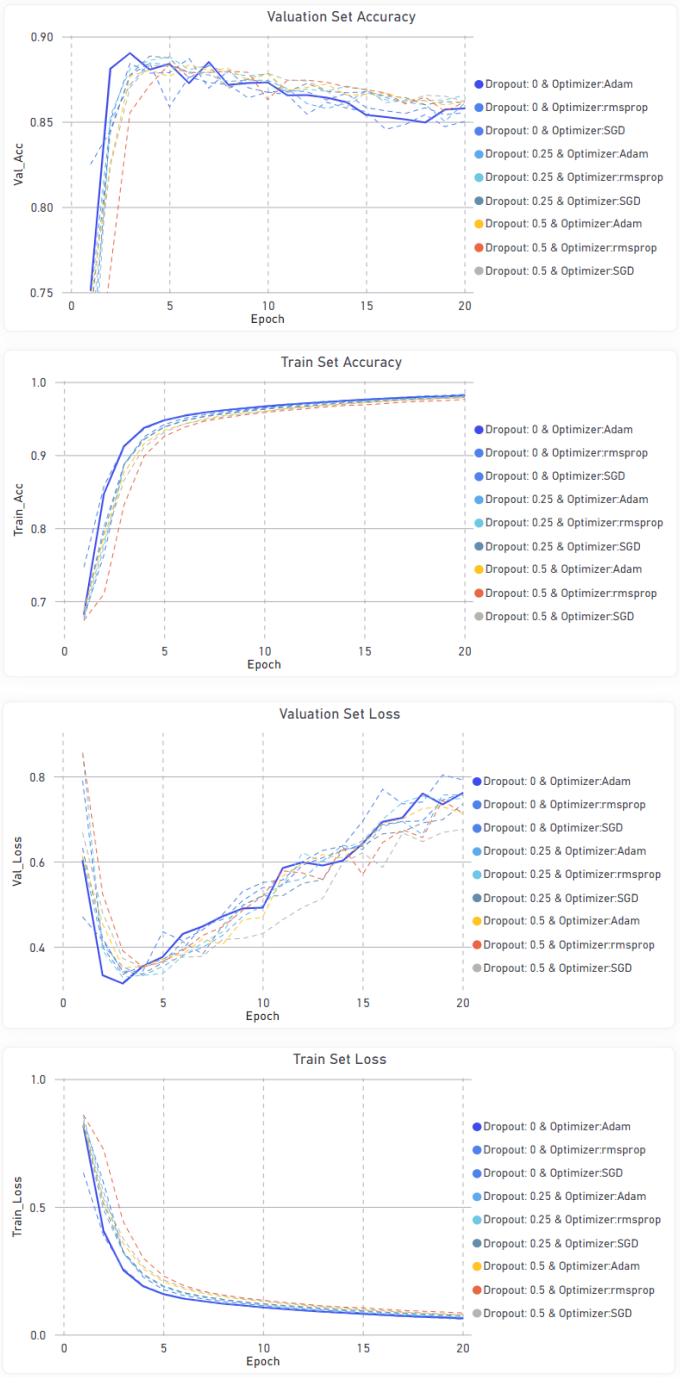

Figure 11: Accuracy and Loss of LSTM-Classifier with Varying Hyperparameters

8.4 *Appendix D*

This is appendix D.

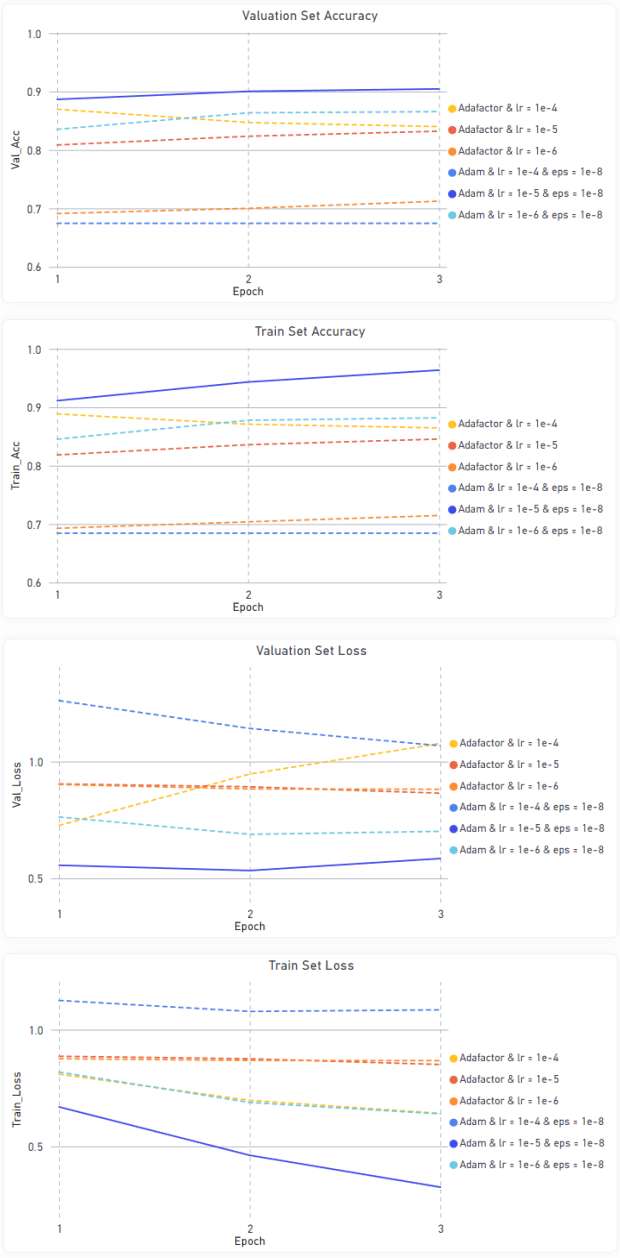

Figure 12: Accuracy and Loss of BERT-Classifier with Varying Hyperparameters

8.5 *Appendix E*

Table 4: Class Performance of Sentiment Classifiers

	MNB		
Class	Precision	Recall	F_1-Score
Bearish	0.57	0.79	0.66
Neutral	0.68	0.75	0.71
Bullish	0.91	0.81	0.86
	LSTM		
Class	Precision	Recall	F_1-Score
Bearish	0.82	0.80	0.81
Neutral	0.82	**0.80**	**0.81**
Bullish	**0.93**	0.94	**0.93**
	BERT		
Class	Precision	Recall	F_1-Score
Bearish	**0.83**	**0.83**	**0.83**
Neutral	**0.84**	0.77	0.80
Bullish	0.92	**0.94**	0.93

8.6 *Appendix F*

This is appendix F.

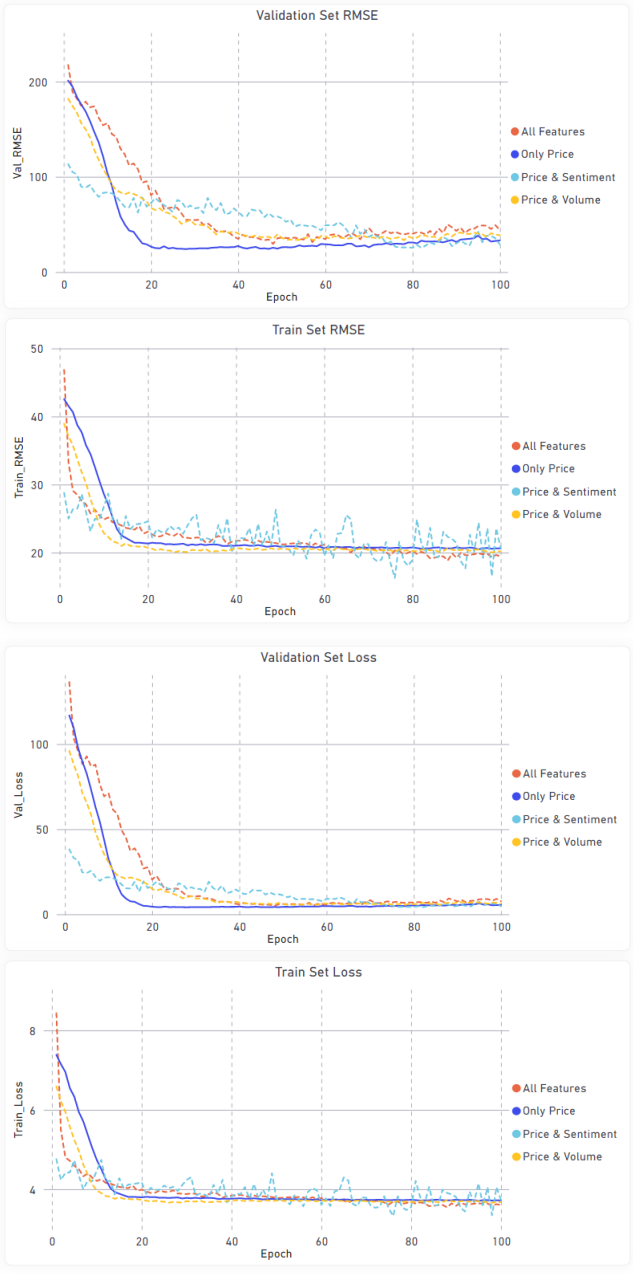

Figure 13: RMSE and Loss of Models with Various Input Features and Their Optimal Hyperparameters

8.7 *Appendix G*

This is appendix G.

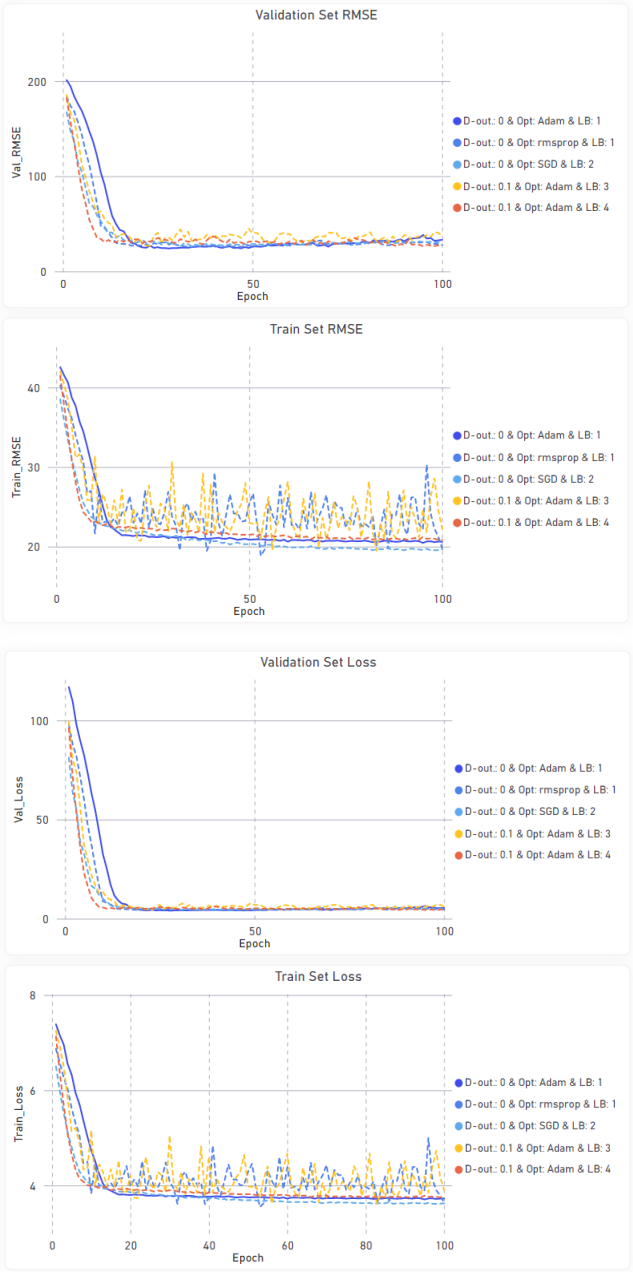

Figure 14: Five Optimal Hyperparameters of LSTM Model With Only Price as Input Feature

8.8 *Appendix H*

This is appendix H.

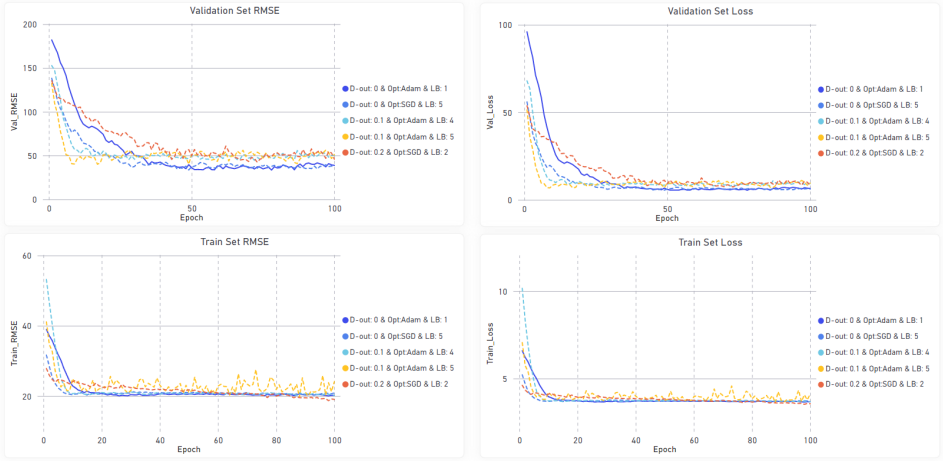

Figure 15: 5 Best Hyperparameters of LSTM Time-Series Model That Uses Price and Trading Volume

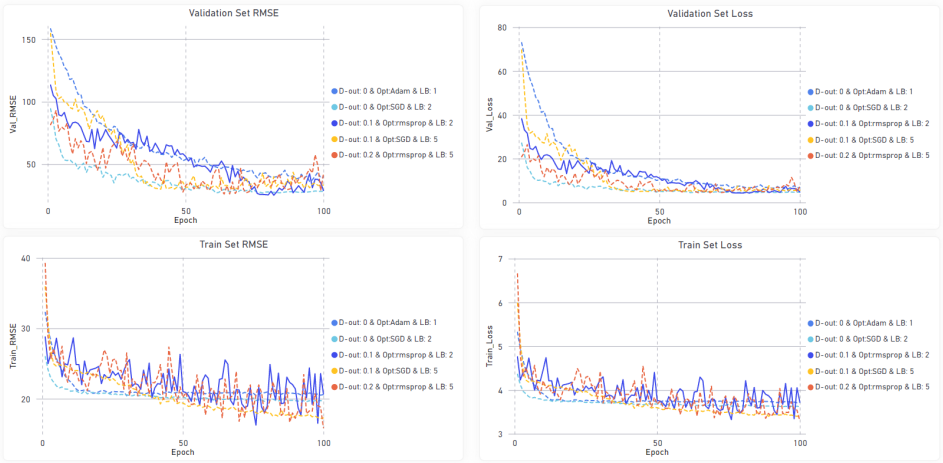

Figure 16: 5 Best Hyperparameters of LSTM Time-Series Model That Uses Price and Sentiment

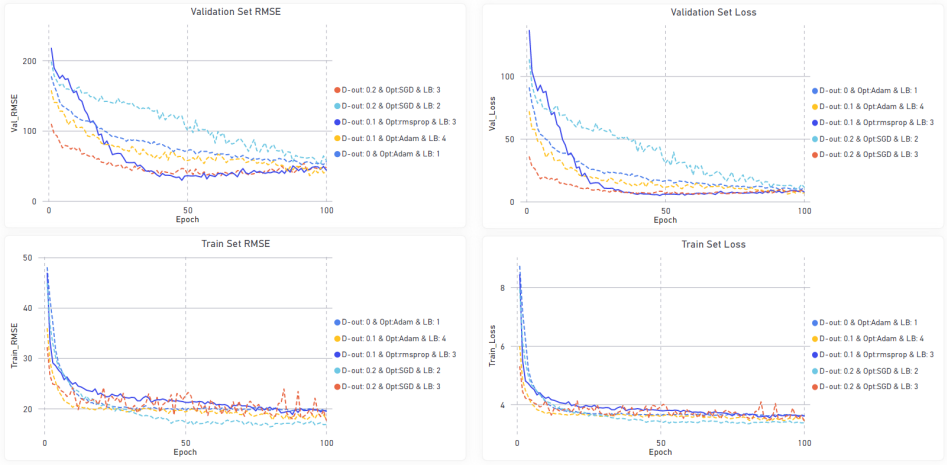

Figure 17: 5 Best Hyperparameters of LSTM Time-Series Model That Uses Price, Trading Volume and Sentiment

8.9 *Appendix I*

As can be seen in Figure 18 the ARIMA model simply seems to extrapolate the previous trend into the future. In contrast, Figure 19 represents the model with the best metrics on the test set, which shows quite strong predictive results. Based on Figure 20 it seems that if a model uses a lookback period of three or greater, its predictions will be smoothed quite a lot.

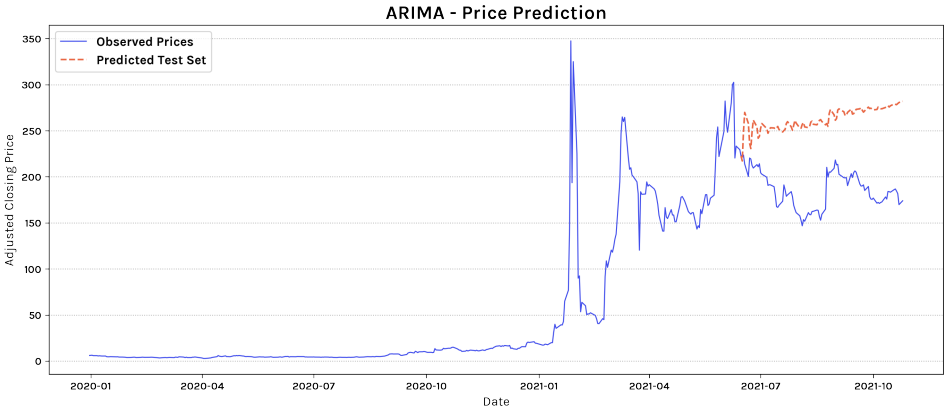

Figure 18: Forecast of ARIMA Model

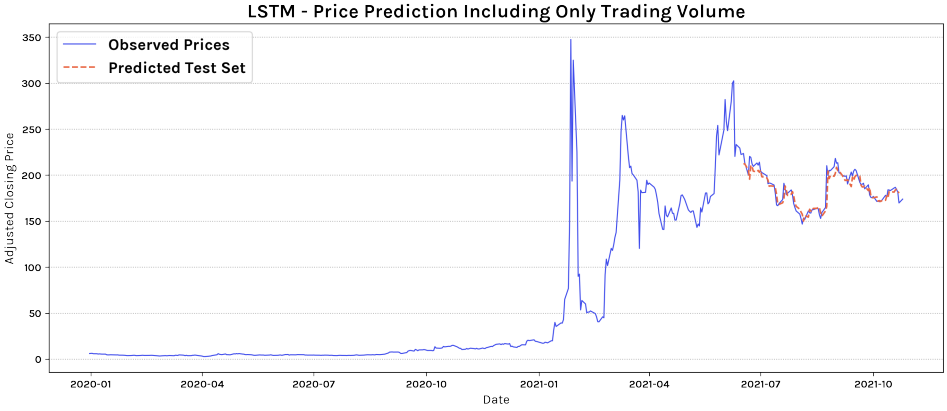

Figure 19: Forecast of Model with Best Metrics on Test Set

Figure 20: Forecast of Models with Higher Lookback