A Comparison of Reddit and Twitter Sentiments In The Prediction of US Stock Prices

A dissertation submitted in partial fulfilment of the requirements for the degree of Master of Science.

by Karthik Kumar Premkumar

Masters Degree in Data Science

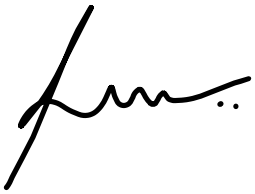
Cardiff School of Technology

Cardiff Metropolitan University, Cardiff

Dec 2021

Declaration

I hereby declare that this dissertation entitled ‘A Comparison of Reddit and Twitter Sentiments In The Prediction of US Stock Prices’ is entirely my own work and it has never been submitted nor is it currently being submitted for any other degree.



Karthik Kumar Premkumar
Date: 17/12/2021

Dr Barry Bentley
Dissertation Supervisor
Date:

Acknowledgements

I would like to extend my thanks and gratitude to my dissertation supervisor, Dr Barry Bentley without the attentiveness and guidance of whom the completion of this dissertation would not have been possible.

I would also like to thank my parents, sister and brother-in-law who were of immense emotional and moral support during my time of work on this dissertation.

Abstract

This study looked at how the investor sentiment from the largest investment community on Reddit compares to that of Twitter in stock price prediction. At a high level, the study also analyses if Reddit is a credible source of investor sentiment by combining sentiments from Reddit with the stock price related technical inputs and using them to predict the stock prices through a Deep Neural Network model. Models which give the best results in recent times were used for both sentiment classification and stock price prediction tasks namely, Bidirectional Encoder Representations from Transformers (BERT) for sentiment classification and a hybrid Convolutional Long Short-Term Memory Neural Network (CNN-LSTM) for stock price prediction. Data from Reddit and Twitter were extracted and processed using python libraries. Four different BERT models, each trained with a different dataset was used to predict the sentiments of five stocks from Reddit and Twitter, after which they were used along with the technical data for each stock to predict the next day closing price via the CNN-LSTM network. Apart from using Root Mean Squared Error (RMSE) to compute the prediction accuracy, directional accuracy was also studied by computing the profits under two assumed scenarios, where the scenarios used the stock price predicted using sentiment data from Reddit and Twitter. The study found that sentiments from Reddit fared better in more cases when compared to that from Twitter. It was also found that, in some cases, combining data from Reddit and Twitter had better performance compared to the respective datasets in isolation.

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

In recent times, with the massive development in Natural Language Processing (NLP) techniques, the use of investor sentiments along with technical data such as the stock price, volume, etc., has increased. In the past few years, research on communities within the Reddit platform have gained importance. The focus peaked starting late 2020, after disruptions in the US Stock Market caused by collaboration of retail investors in the Reddit community r/wallstreetbets led to the closure of a few hedge funds (Frew 2021). From the extensive review done on studies using Reddit data, it was found that studies until now have either tried to research the phenomenon of stock market disruptions and stock market inefficiencies due to collaboration on Reddit (HY Chiu 2021, Long et al. 2021) or used Reddit as the source of investor sentiment to understand stock price movements and predict future stock price (Lubitz 2017, Jung & Jeong 2021, Carvajal, 2021).

Twitter is currently the most used source for investor sentiment from social media. Recent highly cited studies such as (Broadstock & Zhang 2019, Guo & Li 2019, McGurk et al 2020 and Valencia et al 2019) stand testament to this. Review of literature surrounding use of sentiments from Reddit showed that no study has been conducted on comparing the effectiveness of sentiments from Reddit to sentiments from Twitter. This study aims to fill this gap by comparing how sentiments from Reddit measure up to that of Twitter in stock price prediction. This chapter will discuss the background and motivation behind the research, the research problem, followed by the aims and objectives, and finally, the limitations of the study.

1.1 Background And Motivation

In early 2021, the stock markets in the United States of America were perplexed when retail investors on Reddit, specifically on r/wallstreetbets community, traded against positions of short-selling hedge funds and paved the way for the closure of a few hedge funds that had bet against GameStop Corporation (GME) and AMC Entertainment Holdings Inc. (AMC) stocks (Smith & Wigglesworth 2021 and Frew 2021). Collaboration and manipulation of stocks through internet boards and forums have occurred in history (Campbell 2001), for example yahoo message boards were used during the Dot-com bubble to collaborate on investment decisions (Cheung 2021). In recent times, the scale of participation on social media platforms is high. The r/wallstreetbets community within Reddit currently has over 11 million members and is the largest investment community on Reddit (Hartwig 2021). The community had around 6 million members at the start of 2021 (Ghosh 2021) which shows that there was a huge increase in the membership in a brief span of time.

Financial time series prediction and specifically, stock price prediction has been a very challenging research topic because stock price is non-linear and is affected by multiple factors, all of which are not known (Zhang et al. 2018 and Yu & Yan 2020). Starting with linear and parametric models, prediction problems are now increasingly trying to be solved through Machine Learning and Deep Learning models (Sezer et al. 2020). Researchers have also modelled with different types of inputs including fundamental factors of a firm (factors such as earnings per share, profit margin, etc. which try to measure the intrinsic value of a firm) and technical factors (factors that describe the trend and momentum of prices) of stock prices and have had decent success in predicting financial time series albeit never close to a perfect prediction.

Many studies have found that the inclusion of sentiments improve the predictive power of stock price forecasting models. Sentiment analysis can be defined as the study of opinions on a subject of interest (Feldman 2013). In the stock price prediction space, the most used data sources for sentiment mining are a firm's financial statements, financial news – for example, studies by Li et al (2014) and Usmani & Shamsi (2021), and social media posts – for example, studies by Mohan et al (2019) and Nguyen et al (2015), specifically Twitter. But those sources have their disadvantages. A firm's financial statements and reports are often written by the firm and contain an excessively positive tone (Carstens & Freybote, 2019). A financial news article

or report is written by a single person and even if it was peer-reviewed by experts, it would still have some editorial bias. Some reports and articles could be paid releases. With Twitter, the disadvantages lie in the fact that its posts are restricted to 280 characters and there are limits to retrieving data from its API (900 tweets per 15 minutes).

The Reddit saga discussed at the beginning of this section paves the way for an interesting new research area – to analyse if the investor sentiments from the largest investment community on Reddit, r/wallstreetbets are better at predicting stock prices compared to sentiments from Twitter. This serves as an interesting time to do this analysis owing to the rampant increase in membership to the r/wallstreetbets community this year (Ghosh 2021) and also because existing literature on investor sentiments from Reddit have not compared its performance against sentiments from Twitter; this is explained in the section 2.1.2.

1.2 Aim, Objectives, And Research Questions

The purpose of this study is to find if using Reddit as the source of investor sentiments is better for stock price prediction compared to Twitter; a secondary motive is to examine if such a prediction model can be used as a decision-making investment tool by users with less financial domain knowledge.

1.2.1 Aim

The aim of the study is to check how investor sentiments from the Reddit community r/wallstreetbets compare to that of Twitter, which is currently the most used social media platform for investor sentiments in stock price prediction. The study also aims to understand if prediction performance suggests that Reddit can be used as a source for investor sentiment data.

1.2.2 Objectives

The primary objective of the study is to derive investor sentiments from Reddit and Twitter on some stocks from the US Stock Market and to compare how the sentiments perform in

predicting stock prices via a neural network. The comparison is done on both datasets in isolation and in combination.

The best models for sentiment analysis and stock price prediction were determined through a literature review of recent studies focussing on stock price prediction. The research objectives were to determine the best source of investor sentiment between Reddit and Twitter after using them to predict stock prices,

OBJ1. Comparing the prediction accuracy of stock price predictions by calculating a suitable error metric

OBJ2. Comparing directional accuracy of the stock price predictions by computing profit in two assumed scenarios where an initial investment of $100 is made in each stock

1.2.3 Research Questions

Q1. Can stock sentiment information from r/wallstreetbets community on Reddit, in isolation or in combination with sentiments from Twitter provide better stock price predictions compared to sentiments from Twitter alone?

Q2. At a higher level, can the biggest open-source investment community on Reddit, r/wallstreetbets be used as a credible source of investor sentiment data?

Q3. Can a Deep Learning model which uses a stock's technical data and sentiments from Reddit as inputs be used as a passive stock investment tool, where investors with less financial domain experience use the next day predictions for investing in the stock market?

1.3 Limitations

The major limitation of the study is the lack of readily available labelled sentiment dataset for the Reddit community r/wallstreetbets. So, open-source labelled Twitter datasets were used to train the sentiment classification models. Though both are social media platforms, there are significant differences in how people communicate and interact on both platforms. For example, from manual observation, it was noted that the character limit on Reddit comments is 10,000 characters compared to Twitter's 240. The comments on Reddit often contain expletives used colloquially whereas Twitter users are comparatively conservative, especially

in posts relating to investing. This means that the corpus of Reddit comments would have additional words which would be missing without a labelled Reddit dataset. To mitigate this limitation, this study used four different sentiment classification models each trained with a different dataset.

Only the top-level comments in Reddit were used for the sentiment analysis as the computing resources at hand could not handle high volume of data with neural network models. The responses and discussions under the top-level comments were not considered.

Because the membership of r/wallstreetbets increased exponentially in 2021, data from and before 2020 are very limited. Reddit data was collected from the Pushshift project (Baumgartner & Seiler n.d.) which archives Reddit data in near real-time. Data was available for almost all trading days from April to July 2021, with only a few dates missing. Older months had relatively less data, ranging from 10 to 20 days. For two stocks, on a few months in 2020, less than 10 days of data were available. To overcome this, data was filtered on both the Reddit and Twitter datasets to have the same dates. Since sentiments from Reddit and Twitter are compared, it was decided to give equal footing to both sources. Also, to mitigate any limitations to the stock price prediction, the inputs were chosen to have a variety of technical indicators which would incorporate historical data. This way, even if the prediction network does not have inputs for several days explicitly, the historical information of a stock would be intrinsically present in its technical indicators on available days.

The rest of the report is structured as follows. Chapter two discusses the literature reviewed surrounding the stock price prediction models, sentiment analysis models, and the evaluation metrics used for comparing prediction performance. Chapter three focuses on the research methodology, chapter four lists the results followed by chapter five which discusses the results. The final chapter concludes the study and provides recommendations for future studies.

2 LITERATURE REVIEW

2.1 Financial Time Series Prediction

For decades, stock price prediction has been a challenge. Stock price is a non-linear series which can be equated to random motion with a lot of noise. It is seen that, currently, financial time series predictions, stock price forecasting, stock volatility forecasting, etc are increasingly being solved through Deep Learning instead of parametric models used a few years ago.

2.1.1 Neural Networks In Financial Time Series Prediction

When Neural Networks were initially used for financial time series predictions, they were used in combination with parametric models such as Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model and Exponentially Weighted Moving Average (EWMA). Roh (2007) used a combination of Artificial Neural Network and Generalized Autoregressive Conditional Heteroskedasticity (NN-GARCH model) and Artificial Neural Network and Exponential Generalized Autoregressive Conditional Heteroskedasticity (NN-EGARCH model) to forecast volatilities and found improvement in prediction compared to standalone Artificial Neural Networks (ANNs). Predictive power was improved by 29.43% using NN-EGARCH and 7.67% using NN-GARCH and the directional accuracy of the combined models was around 60% while that of a standalone Neural Network was 43%. Hajizadeh et al (2012) found that both Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were halved when using a combination EGARCH and Artificial Neural Networks, compared to the

best tuned EGARCH model in the prediction of price volatility. Such studies showed that the performance of standalone ANNs was inferior to the hybrid models mentioned. This can be attributed to the lack of memory in traditional ANNs, i.e., the lack of ability in traditional ANN models to store prior input values and use them in future predictions.

In the years that followed, with the advent of Deep Learning, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have outperformed parametric models and hybrid combinations of (parametric model + traditional ANN). Recurrent Neural Networks have a memory cell implemented in each neuron, which retains previous data and uses it for future predictions. Long Short-Term Memory (LSTM) Network is a type of RNN that overcomes some disadvantages of traditional RNNs, the main one being, the memory of RNNs are too short to derive accurate predictions in a time series. Due to its ability to retain relevant memory for a longer time, it also overcomes the gradient vanishing problem (situations where the error propagated through a neural network, is too small to update weights in the network hence making the learning of the network slow) present in traditional RNNs (Sundermeyer, et al., 2012). Convolutional Neural Networks, on the other hand, were initially introduced for image processing problems but are increasingly being used for other areas including time-series predictions. Feature extraction or dimensionality reduction is an important aspect of CNNs which helps reduce the dimensions in input dataset and thus helps downstream processes to work with the most important features in a dataset. According to Hu et al (2014), when compared to traditional neural networks, CNNs can share weights across the network more effectively hence preventing the local minima gradient dissent problem (when the gradient descent gets stuck in a local minima rather than the global minima of the loss function). The detailed workings of CNNs and LSTMs are explained in APPENDIX A: STRUCTURE OF CNN AND LSTM NETWORKS.

The popularity of CNN and LSTM in financial time series and stock price prediction is evident from the literature survey by Hu et al (2021). The below chart from their work shows that CNN and LSTM make up the bulk of recent research on predictions in the stock market.

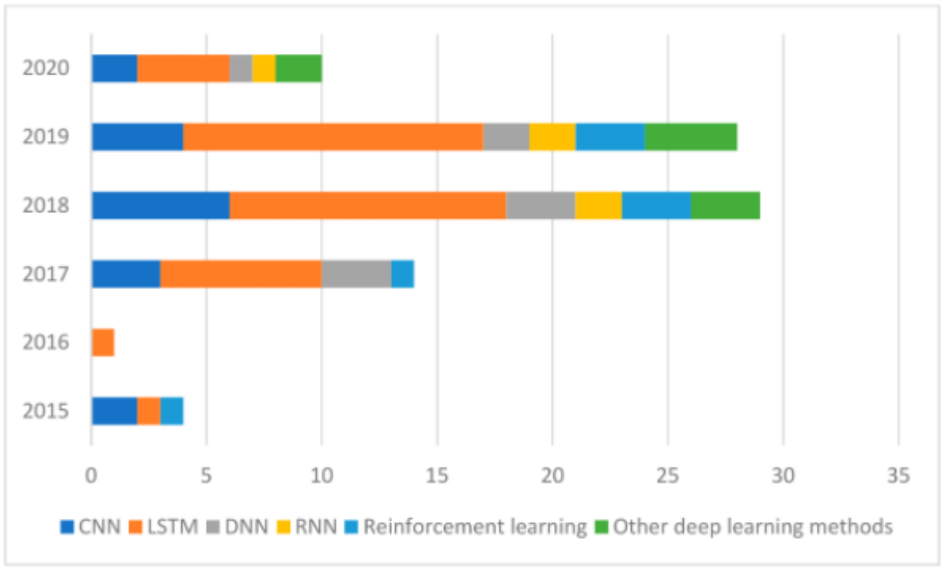


Figure 2.1: Systematic Literature Survey Done By (hu, et al., 2021) On Types Of Neural Networks Used In Stock Price Prediction Problems

From 2019, CNN-LSTM hybrid networks have been shown to have higher prediction power compared to standalone CNN or LSTM.

- Lu et al (2020) implemented a CNN-LSTM model and compared its prediction against five other models – Multilayer Perceptron (MLP), CNN, RNN, LSTM, and a hybrid CNN-RNN model. They found that CNN-LSTM had the best accuracy with an RMSE value of 39.688 compared to 41.003 for a standalone LSTM and 42.967 for a standalone CNN.
- Livieris et al (2020) measured the performance of two different CNN-LSTM networks (with different parameters) against standalone LSTM networks in predicting gold prices. The CNN-LSTM models had a lower prediction error with an RMSE value closer to 0.01 compared to 0.02 for LSTM models.
- Mehtab and Sen (2020) compared the performance of standalone CNN, LSTM, and hybrid CNN-LSTM models to predict NIFTY 50 index. They make a note of the good performance of all networks, but the CNN-LSTM model had the lowest error. They also note that a disadvantage with the CNN-LSTM model is that it is slower than the other models owing to the increased complexity of the network by cascading a CNN followed by an LSTM model.

The focus of this study is to compare how sentiments from Reddit perform in comparison and in combination with sentiments from Twitter, and not to compare different types of prediction models. After reviewing the above literature on neural networks in financial time series prediction, CNN-LSTM model was chosen as the best fit for this study as it seemed to be the model producing the best performance. Since this study does not focus on intra-day price prediction but on daily price prediction (i.e.) prediction of next day's closing price using current day's inputs, the slow performance of the model is acceptable considering its high accuracy.

2.1.2 Sentiment Inputs In Financial Time Series Prediction

Most of the widely cited research in the past few years concerning financial time series predictions with sentiments from social media used Twitter as the data source. McGurk et al (2020) computed sentiments from Twitter data and compare how a model predicting sentiments with a labelled training dataset predicts abnormal movements in stocks compared to a bag-of-words model where sentiment is computed based on groups of positive and negative words. They found that investor sentiments are highly correlated with abnormal movements in stock prices for smaller firms but concluded that they found limited evidence of investor sentiments improving price forecasts. In contrast, Valencia et al (2019) used Twitter to compute the valence and polarity of cryptocurrency related tweets and found that sentiment analysis in isolation and in combination with technical inputs provided a good prediction of cryptocurrency prices They were able to get an accuracy of 0.72 when using both market and Twitter data to predict Bitcoin prices. Guo and Li (2019) used Twitter data to compute a Twitter Sentiment Score by scoring positive and negative words in tweets and then using the score to correlate with FTSE 100 index movement. They had good results in predicting the upward trends (97.87% accuracy) compared to downward movements in trend(5.13% accuracy). Broadstock and Zhang (2019) used Twitter to get investor sentiments and found that both firm and market-wide sentiments affect stock prices, and that pricing power of stocks are affected by sentiments in the stock market.

The use of Reddit as a source of investor sentiments has been limited and is only gaining importance in recent times after the rally of stocks like $GMC where retail investors on Reddit platform made investment moves against hedge funds shorting the stock, which led to the closure of a few hedge funds in the United States of America. Recent studies using Reddit data are more focussed on using Reddit data as an additional input for stock price prediction or to

study the rally of certain stocks such as $GME. For example, Jung and Jeong (2021) used memes data from the Reddit community r/dankmemes and found a positive relationship between the sentiments on the community and large-cap US stocks (there was no relationship with international stock markets). Long et al (2021) found that the sentiments on the Reddit community r/wallstreetbets affected the intra-day price movements of $GME stock.

None of the research surrounding Reddit has compared sentiments from Reddit to the sentiments from Twitter in stock price prediction. This study aims to fill that gap by comparing the performance of Reddit and Twitter sentiments in stock price prediction when used as inputs along with other technical inputs. Additionally, this study also looks at performance of prediction when sentiments from both platforms are combined.

2.2 Technical Indicators

There are two basic types of stock market analysis – fundamental and technical. The former is concerned with long-term investing and looks at the health of the economy and firm(s) in question using financial reports, annual reports, assets and liabilities, profit ratios, economic status, etc. Technical analysis is mostly used in short-term trades like intraday trading and swing trading and involves the price movements of stocks. Traders analyse patterns, trends using different indicators to predict future stock price movement based on historical prices.

Since this study is focussed on short-term investing (swing trading), technical analysis inputs will be most suitable. At a broad level, technical indicators can be classified as trend, momentum, and sentiment indicators (Colby 2003). While trend indicators measure the upward, downward, and sideways movement of prices, momentum indicators measure the rate of change in price and velocity of price changes and sentiment indicators help identify overbought and oversold conditions. Momentum and sentiment indicators are leading indicators, meaning they help identify a trend before it takes.

From the review conducted on technical analysis inputs used in other financial prediction models, it was found that there were three types of inputs used,

1. Raw stock price details such as opening price, closing price, volume, highest intraday price, etc (Moghaddam et al 2016; Selvin et al 2017; Yong et al 2017).

2. Technical indicators like Simple Moving Average, Exponential Moving Average, RSI, William's R%, MACD, etc (Chandar 2021; Gao & Chai 2018; Naik & Mohan 2020; Qi et al 2020)
3. A combination of the above two (Hoseinzade and Haratizadeh 2018; Mehtab and Sen 2020)

There was no pattern or selection principles mentioned in the studies. As most of the studies looked at improving the predictive power, they used inputs mentioned in surrounding literature.

After reviewing the above literature, it was decided to use the technical indicators mentioned in (Gao and Chai 2018). Unlike raw price and volume data, technical indicators contain some historical information (for example, moving averages show how price is moving across several days) and the list of 15 inputs in (Gao and Chai 2018) contain both trend and momentum indicators.

Table 2.1: Technical Indicators Used In The Study Along With The Respective Indicator Type

Input	**Name**	**Indicator type according to definitions by Colby (2003)**
ACD	Accumulation distribution	Trend
MACD	Moving Average Convergence Divergence	Sentiment
CHO	Chaikin Oscillator	Momentum
Highest	Highest closing price	Trend
Lowest	Lowest closing price	Trend
SO-%K	Stochastic Oscillator	Sentiment
SO-%D	Stochastic Oscillator	Sentiment
VPT	Volume Price Trend	Trend
W-R%	Williams %R	Momentum
RSI	Relative Strength Index	Sentiment
MOME	Momentum	Momentum
AC	Acceleration	Momentum
PROC	Price Rate of Change	Momentum
VROC	Volume Rate of Change	Momentum
OBV	On-Balance Volume	Trend
Log_ret	Log return of previous day	Trend

The formula for calculating the various technical indicators is shown in the below Figure 2.2, an excerpt from (Gao and Chai 2018). CL, HI, LO and VO are the closing price, high price, low price, and volume data points for a stock.

Technical Indicator	Formula
ACD	$\mathrm{ACD} = \mathrm{ACD}_{\text{previous - day}} + VO \times \frac{(CL - LO) - (HI - CL)}{HI - LO}$
MACD	$\mathrm{MACD} = \mathrm{EMA}(CL, 12) - \mathrm{EMA}(CL, 26)$
CHO	$\mathrm{CHO} = \mathrm{EMA}(AD, 3) - \mathrm{EMA}(AD, 10)$
Highest	$\mathrm{Highest}\,(t) = \max\,\left(\sum_{i=1}^{t} CL_i\right)$
Lowest	$\mathrm{Lowest}\,(t) = \min\,\left(\sum_{i=1}^{t} CL_i\right)$
SO-%K	$\mathrm{SO\text{-}\%K} = \frac{(CL - \mathrm{Lowest}(5))}{\mathrm{Highest}(5) - \mathrm{Lowest}(5)} \times 100\%$
SO-%D	$\mathrm{SO\text{-}\%D} = \mathrm{MA}(\mathrm{STOS\text{-}\%K}, 3)$
VPT	$\mathrm{VPT} = \mathrm{VPT}_{\text{previous - day}} + VO \times \frac{CL - CL_{\text{previous - day}}}{CL_{\text{previous - day}}}$
W-R%	$\mathrm{W\text{-}R\%} = \frac{\mathrm{Highest}(n) - CL}{\mathrm{Highest}(n) - \mathrm{Lowest}(n)} \times 100\%$
RSI	$\mathrm{RSI} = 100 - \frac{100}{1 + \mathrm{RS}}$
MOME	$\mathrm{MOME}\,(n) = CL_t - CL_{t-n}$
AC	$\mathrm{AC}(t) = \mathrm{AO} - \mathrm{MA}(AO, t)$
PROC	$\mathrm{PROC} = \frac{CL - CL_{t-12}}{CL_{t-12}} \times 100\%$
VROC	$\mathrm{VROC} = \frac{VO - VO_{t-12}}{VO_{t-12}} \times 100\%$
OBV	If $CL \geq CL_{\text{previous - day}}$, $\mathrm{OBV} = \mathrm{OBV}_{\text{previous - day}} + VO$
	If $CL < CL_{\text{previous - day}}$, $\mathrm{OBV} = \mathrm{OBV}_{\text{previous - day}} - VO$

Moving average (MA) and exponential moving average (EMA).

$$\mathrm{MA}\,(x, t) = \left(\sum_{i=1}^{t} x_i\right)/t$$

$$\mathrm{EMA}\,(x, t) = \alpha \times x + (1 - \alpha) \times \mathrm{EMA}_{\text{previous - day}},\ \ \alpha = 2/(t + 1)$$

$$\mathrm{RS} = \frac{\text{Average of upward price change}}{\text{Average of downward price change}}.$$

$$\mathrm{MedianPrice} = (HI + LO)/2$$

$$\mathrm{AO}\,(t_1, t_2) = \mathrm{MA}\,(\mathrm{MedianPrice}, t_1) - \mathrm{MA}\,(\mathrm{MedianPrice}, t_2)$$

Figure 2.2: Formulae For Computing The Technical Indicators Used In The Study

Table 2.2: Short Definition Of Technical Indicators Used In The Study

Indicator	Brief Definition	Reference
RSI	RSI is a value between 1 to 100 which signifies if the stock is overbought or oversold. It is calculated using average upward price change and average downward price change of past n days. This study used 14 past days to calculate the averages.	Utthammajai and Leesutthipornchai (2015)
PROC	PROC and VROC compare the current price to price before n days and the current volume to volume before n days. This study used n=12.	
VROC		
MACD	MACD is calculated using 12- and 26-day Exponential Moving Averages. Like RSI, it specifies overbought and oversold signals.	
ACD	ACD is a cumulative indicator which specifies the supply and demand in a stock using price and volume information. A key component of ACD is the Money Flow Multiplier which is [((CL-LO)-(HI-CL))/(HI-LO)]. This is then multiplied by the volume to get the current day's Money Flow Volume. THE ACD value for each day is computed by adding the Money Flow Volume to previous day's ACD.	
CHO	Chaikin Oscillator is computed by getting the difference between the 3-day exponential moving average of ACD and 10-day exponential moving average of ACD.	
W-R%	Williams %R shows how the closing price of current day is related to the highest and lowest closing prices over the n previous days. This study used n=14.	Shynkevich et al (2017)
	Stochastic indicators signify how the price has moved in relation to the price range over a period. Stochastic %K and %D are momentum indicators which signify oversold and overbought conditions for a stock. %K is calculated as the difference between current closing price and the lowest closing price from past n days to the difference between the highest and lowest closing prices in the past n days. This study used n=5. Stochastic %D is the 3-day moving average of Stochastic %K.	Wu et al (2014)
Highest	Highest Close Price (Highest) and Lowest Close price (Lowest) specify the highest and lowest closing prices I the past n days. This study used n=14 for these indicators.	
Lowest		
VPT	Volume Price Trend (VPT) is a measure of price velocity which is a measure of strength in a trend. It can be used to forecast trend reversals.	
MOME	Momentum (MOME) which is the difference between current closing price and closing price n days earlier states the tendency of the prices to keep their upward or downward trends.	
OBV	The On Balance Volume Indicator (OBV) is a cumulative indicator which weights price action by volume to determine upcoming trends in stock price.	Thomsett (2017)
AC	Acceleration (AC) is used to predict the strength of a trend and to forecast trend reversals. It is based on Awesome Oscillator (AO) which is calculated as the difference between 5-day median price and 34-day median price. AC is the difference between the current day AO and the 5-day average AO.	ThinkMarkets (n.d.)

2.3 Sentiment Analysis

Sentiment Analysis (or opinion mining) is a branch of NLP which is used to identify if a text has a positive, neutral, or negative tone. Sentiment analysis (and NLP in general) is a rapidly developing field with the advent of neural networks. From a survey of methods on stock price prediction using sentiment analysis (Alzazah & Cheng, 2020), Machine Learning and Deep Learning were found to be the most current technologies in this space.

Within the Deep Learning space, there are two major methods in practice,

1. Converting text to word embeddings (vector form) through tools like word2vec and then training a model for sentiment classification such as the work by Acosta et al (2017) where they were able to predict sentiments with 72% accuracy.
2. Using pre-trained models like Bidirectional Encoder Representations from Transformers (BERT) developed by Devlin et al (2019). These models are pre-trained on a large corpus using a lot of resources, which would be impossible with a standalone machine.

Recent works also show that BERT outperforms the competition since BERT models are context-dependent compared to normal word embeddings such as word2vec (Devlin and Chang 2018). Context improves the NLP process, for example, a model which can differentiate between Apple Corporation and the fruit apple will be a better model compared to one which embeds both instances using the same vector.

Transformers are a type of neural network created for NLP tasks by Google, with a study (Vaswani et al 2017) proving its effectiveness in language translation efficiency (converting English to French). The Transformer model has an encoder that processes the input and a decoder that produces the output. The encoder creates initial embeddings for each word in a sentence and then updates the word embeddings, creating a final embedding for each word using the initial embeddings of all words in a sequence (Uszkoreit 2017). Thus, the encoder in a Transformer acts as a context-based encoder where word embeddings are contextual. BERT model is based on a Transformer. There is no decoder in BERT, but the encoder acts in the same way to a Transformer model. Devlin and Chang (2018) also mention how BERT is highly bi-directional when compared to similar context-dependent models like Embeddings from Language Model (ELMo) and Universal Language Model Fine-tuning (ULMFit). This helps BERT provide high context embeddings.

According to Devlin and Chang (2018), BERT can be used in two ways,

1. By just fine-tuning the available open-source BERT model for specific NLP tasks (like sentiment analysis)
2. Use the pre-training code shared by google to pretrain the BERT Model using custom datasets specific to the research area or domain, then fine-tune it and use it. This is shown in Figure 2.3.

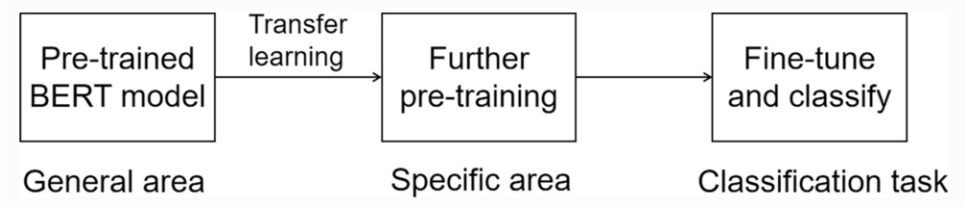


Figure 2.3: Training Process Of BERT (Li, 2021)

There are multiple versions of the BERT model available (Devlin, 2018) with BERT BASE UNCASED being the simplest model. A majority of highly cited studies which used BERT model followed the second method. The following works showed that a BERT model trained and fine-tuned with financial datasets have the best performance with NLP tasks when compared with other state of the art models,

Table 2.3: Details Of Various Research Works On Sentiment Analysis Comparing BERT Model To Other NLP Models

REFERENCE	**MODELS COMAPRED**
(Hiew et al 2019)	• BERT • Transformer + attention • Multichannel CNN initialized with Shifted Positive pointwise mutual information (PMI) • BiLSTM initialized with Shifted Positive pointwise mutual information (PMI) • FastText
(Araci 2019)	• LSTM • LSTM with Embeddings from Language Model (ELMo) • Universal Language Model Fine-tuning (ULMFit) • BERT pre-trained on financial corpus (FinBERT)
(Liu et al 2020)	• Rule-based model from (Fatima et al 2019) • BiLSTM-CRF (Du et al 2019) • Deep-Att (Tian and Peng 2019) • BERT-S (Du et al 2019) • FinBERT-BASE • FinBERT-LARGE
(Sebastian & Isa 2020)	• BERT • Word2Vec
(Devlin et al 2019)	• Pre-OpenAI State Of The Art model • Bi-directional LSTM + ELMo + Attention • Generative Pre-trained Transformer • BERT-BASE • BERT-LARGE

From the various studies, it was seen that BERT trained on financial corpus achieved above 80% accuracy (in some studies over 95% accuracy) on various NLP tasks. The most incorrect predictions occurred where the model could not detect positive tone without the presence of

words like increasing, profit, etc. However, this level of accuracy is very high in comparison to other models especially with respect to financial text data.

To pre-train the BERT model for proficiency with financial corpus, Liu et al (2020) used a combination of general English corpus and financial domain corpus. The financial domain corpora used were,

- FinancialWeb dataset from CommonCrawl News dataset which had 6.38 billion words and was of 15 GB.
- FinancialWeb dataset from FINWEB which was of 9 GB.
- YahooFinance dataset which was obtained by web crawling through yahoo finance website for articles from 2016 to 2020. This dataset had about 4.71 billion words and was of 19 GB.
- RedditFinanceQA which was collected from the Reddit on posts with financial context where the posts contained four or more upvotes. This dataset was of 5 GB with about 1.62 billion words.

Araci (2019) used,

- TRC2-financial dataset, which is a subset of TRC2 Reuters dataset, filtered with financial terms. It had 46143 documents and over 29 million words.

These datasets posed a few challenges hence were not used in our study,

- these datasets were large and the resources available would not be able to process such huge datasets
- pre-training tasks usually take about 54 hours if done using a resourceful system with two TPUs. On open-source platform such as Google Collab, TPUs are assigned based on availability and assigning two TPUs is often improbable (Antyukhov 2019).
- some of the datasets mentioned above seemed to involve manual work where quality could be compromised. For example, in the dataset used by Araci (2019), the corpus is obtained from Reuters dataset by filtering on financial terms but there is no existence of an exhaustive financial term dictionary. Also, using manual effort to classify financial text would add complexity and delays to the study.

From the datasets above, it could be deduced that the studies mentioned above had access to large scale resources to process such large datasets. Since this study had resource constraints, some review was done to check if BERT models could be used without pre-training, just by

fine-tuning it for sentiment classification task. Few research studies had found that BERT models performed better than other models even when only fine-tuned (without any additional pre-training). For example, Manish Munikar (2019) compared a fine-tuned BERT BASE model with word vectors, RNN, LSTM, CNN, BiLSTM and found it to have a higher accuracy in sentiment prediction. The only other model which outperformed BERT BASE was BERT LARGE which is a more complex BERT model with more layers and parameters. Also, the authors of BERT (Devlin and Chang 2018) state that the purpose of BERT was to have a pretrained model which could be used by other just by fine-tuning it. This seemed to be the best approach considering the scope of our study and the computing resources available.

2.4 Evaluation Metrics

The below table 2.4 shows details about the evaluation metrics used in the recent studies focussed on CNN-LSTM models.

Table 2.4: Inputs Used, And Evaluation Metrics Used In The Reviewed Studies On CNN-LSTM Models

REFERENCE	**EVALUATION METRICS**
(Lu et al 2020)	Mean Absolute Error (MAE) Root Mean Square Error (RMSE) R-square (R^2)
(Livieris et al 2020)	Mean Absolute Error (MAE) Root Mean Square Error (RMSE)
(Mehtab & Sen 2020)	Root Mean Square Error (RMSE)

Studies with a hybrid CNN-LSTM model used the following metrics to measure performance,

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- R-square (R^2)

And among the metrics, Root Mean Square Error is the common metric.

From a systematic literature review of financial time series forecasting with Deep Learning between 2005 and 2019 (Sezer et al 2020), the following were the most used metrics,

- Accuracy
- F1-score
- Root Mean Square Error (RMSE)
- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)

Among the metrics Root Mean Square Error has the most advantage. Mean Absolute Error (MAE) has the same problem as any mean value has in a dataset – it considers both small and large errors in predictions equally. Mean Squared Error (MSE) penalizes larger prediction errors because it squares the error values due to which, the scale of the metric does not conform to the scale of the data. RMSE is of the same scale as the data, and it also penalizes larger prediction error more than smaller errors. Section 3.6.4 in the Methodology chapter gives additional challenges with usage of error metrics.

3 METHODOLOGY

3.1 Research Philosophy, Approach, And Strategy

As per (Saunders et al 2015), this study has a positivist research philosophy as the data used were structured with large samples and there was no influence of the researcher in the study. Processing steps and evaluations metrics are quantitative, and this study was done objectively. It also has an inductive approach, as no deductions were based on existing theory. Instead, data was used to make conclusions. Moreover, the research strategy is experimental because this study analysed the association between different variables to the price movement in the stock market. From a research approach perspective, this study used a mixed-method approach since both quantitative data (technical analysis indicators) and qualitative data (posts and comments from Reddit) were used. Also, the data collection timeline was longitudinal since historical data was collected and used to predict future output values.

The following sections will expand on the above points made regarding research philosophy, approach, and strategy.

3.2 Data Collection

All data collection and processing activities were performed in python language using Anaconda Jupyter Notebook environment. The data collected included technical data for each

of the stocks chosen for analysis and data from Reddit and Twitter for the stocks of interest, as explained in the following sections.

3.2.1 Reddit Dataset

The investment community (termed 'subreddit' in Reddit) with the most members in Reddit platform is r/wallstreetbets. This subreddit has a thread called "Daily Discussion Thread for <Date>" for each trading day where members discuss about stocks, their positions, predictions, etc. This is where data was collected on specific stocks for performing sentiment analysis using BERT models.

Figures 3.1 and 3.2 show a daily discussion thread on r/wallstreetbets and how people discuss under a thread. The usernames have been masked for anonymity.

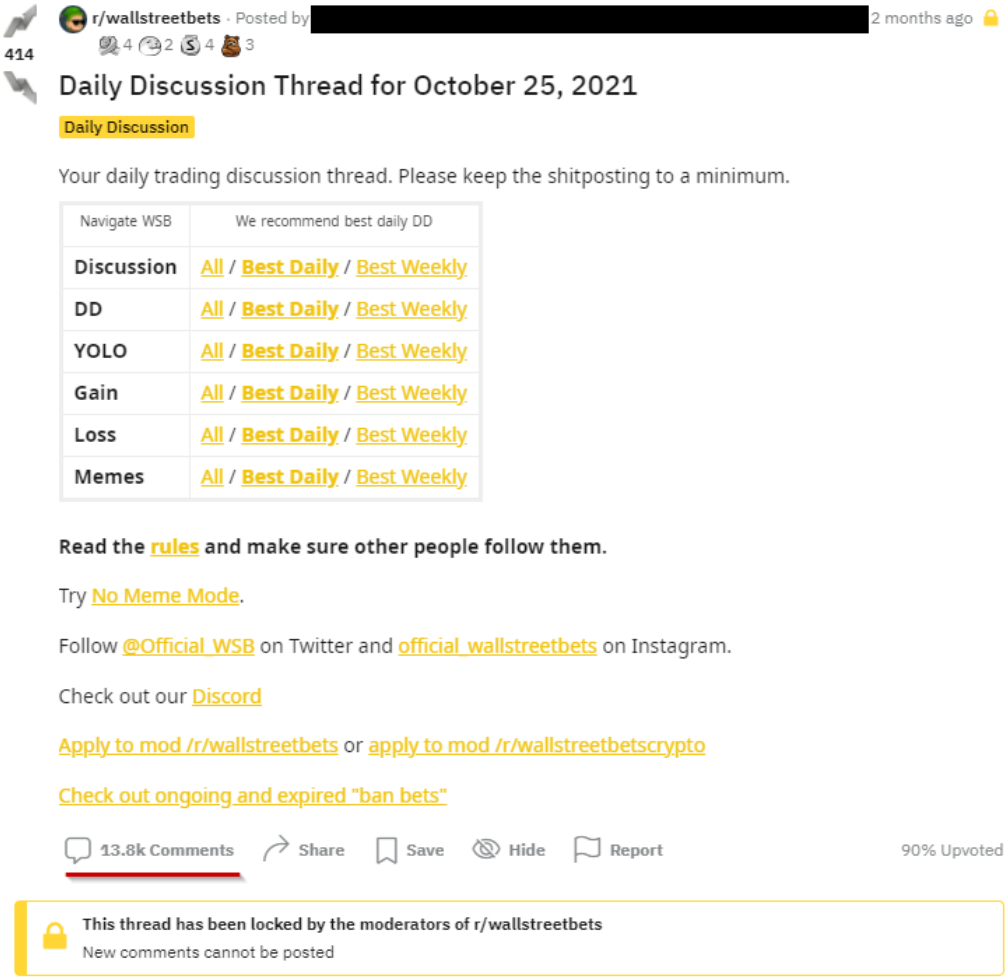


Figure 3.1: Daily Discussion Thread On r/wallstreetbets Reddit Community For October 25, 2021

Figure 3.2: Discussion In The Comments Section Of A Thread

There are two ways to collect data from Reddit,

1. An open-source API provided by Reddit to retrieve its data. It requires users to request API access on its website which is a straightforward process. This method has a request limit of 60 requests per minute and 100 items per request.
2. An external open-source project called Pushshift archives Reddit data in near real-time. There are python libraries available to communicate with the Pushshift API to retrieve required data. This method does not have low-rate limits like the former, but a minor disadvantage is that data for some days may not available. Although the Pushshift project does near real-time data ingestion to the archive, ingestion failures and updates are only resolved periodically.

PMAW python library which gets data from Pushshift archive was used for this study. It was the best option considering the limited time and resources available for this study. A single day in the mentioned Reddit thread has on average over 10,000 comments and using Reddit's own API with it's rate limit would have required several days to extract the data.

3.2.2 Selection Of Stocks

The stocks to be used for the study were selected from the extracted Reddit data. To check for stocks being discussed in the Reddit data, the ticker symbol of stocks (for example, $TSLA is the ticker symbol for Tesla, Inc.) were used as extensive manual observation found that the community discussed stocks based on their ticker symbols either with or without the '$' sign. This is shown in the below Figure 3.3, where a few screenshots of comments discussing certain stocks are presented.

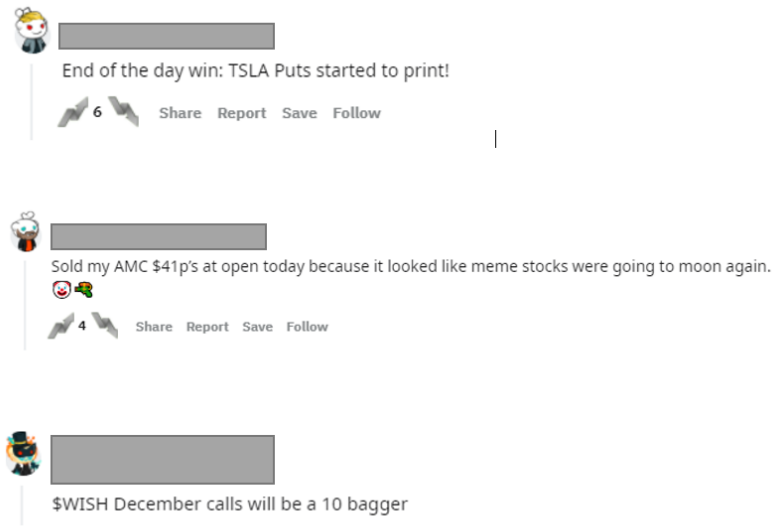


Figure 3.3: Screenshots Of Comments Mentioning Stocks On r/wallstreetbets Daily Discussion Thread

Due to the limited computing resources at hand which would make it difficult to process around 10,000 comments per day, it was decided to use only the top-level comments (i.e., the replies under each main comment was ignored). The ticker sign of all stocks was saved manually in a csv file and this list was used to check which stock each comment on Reddit data talked about. The list of stocks was taken from (Nasdaq n.d.). After this, the below are the criteria for selecting the stocks to be used in this study,

1. The stocks should be discussed on Reddit on a significant number of days, so we can extract sentiments for many days and use them as inputs along with the technical indicators for price prediction.
2. There are certain stocks termed 'meme stocks' which are very volatile. A mix of stocks including and excluding meme stocks were chosen (Table 3.1)

Table 3.1: Stocks Selected For Analysis In This Study From Reddit Data

TICKER	COMPANY	COMMENTS
AMC	AMC Entertainment Holdings, Inc.	Meme stock
TESLA	Tesla, Inc.	Meme Stock
AMD	Advanced Micro Devices, Inc.	Meme Stock with strong fundamentals
BABA	Alibaba Group Holding Limited	General Stock
DKNG	DraftKings	General Stock

The web article by Sirois (2021) explains the phenomenon of meme stocks and how they are volatile, often riding on the back of the momentum of investors investing into them, even when the respective company does not have strong fundamentals. As explained in the article, $AMC is a meme stock with poor fundamentals while $TSLA is a meme stock with good fundamentals, with the market value of $TSLA being too high for a firm of its stature and business. $AMD is an example of a meme stock with excellent fundamentals to back up its rally in the stock market. $BABA and $DKNG cannot be classified as meme stocks but were discussed on a lot of days in the Reddit community.

3.2.3 Twitter Dataset

After selecting stocks (ticker symbols) from the Reddit dataset, data for the selected tickers were extracted from Twitter. Like Reddit, there are two ways to collect data from Twitter,

1. Using Twitter's own API which requires a payment to be used to collect extensive data and on historical timeframes.
2. Using open-source scrappers, which would enter our search term in Reddit and extract the publicly available content on specific dates.

For similar reasons to the Reddit case, the second option was used. A python library called 'snscrape' was used for extracting Twitter data. Data for all the five aforementioned stocks were collected from Twitter. Since Twitter does not have a hard filtering mechanism on the search results, often, sparsely related content would start showing up in the search results. To overcome this, 1000 comments were extracted each day for each stock to minimize the effect of getting unrelated search results. Again, data was extracted from Twitter using the search terms in the form of "$<ticker>" and "<ticker>".

3.2.4 Technical Indicators

The stock price data was retrieved by using the 'yfinance' package in python, which retrieves a stock's historical price based on the parameters given by the user. Seen below is an excerpt of the code where 'yfinance' package was used to get AMD stock's historical price data for a week.

```
[1]: import yfinance as yf

[2]: amd = yf.Ticker("AMD")

     # get stock info
     amd.info

     # get historical market data
     amd_week_hist = amd.history(period="7D")

[3]: amd_week_hist
```

[3]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2021-06-24	84.389999	87.139999	84.370003	86.099998	42217700	0	0
2021-06-25	86.339996	86.360001	85.099998	85.620003	27804500	0	0
2021-06-28	86.379997	88.000000	86.150002	87.080002	30262000	0	0
2021-06-29	87.410004	90.300003	86.660004	89.519997	46181000	0	0
2021-06-30	90.820000	94.339996	90.599998	93.930000	70721500	0	0
2021-07-01	94.040001	94.180000	91.699997	93.309998	58059000	0	0
2021-07-02	93.279999	95.269997	92.209999	94.699997	51316700	0	0

Figure 3.4: Sample Python Code Showing AMD Stock's Price Data Using yfinance Module

Daily data for Opening, Closing, High and Low prices and trading volume were extracted. The Closing price, High price, Low price, and Volume are represented in the technical indicator formulae by CL, HI, LO and VO, respectively.

The 15 technical indicators mentioned in the Literature Review section in Table 2.1 were calculated using their respective formulae mentioned in Figure 2.2 with the time periods mentioned in Table 2.2.

3.3 Preparation Of Bert Model

3.3.1 BASE BERT Model

The BASE BERT UNCASED model from 'huggingface' transformers library (Platen n.d.) was used for our study as it is the simplest BERT model.

It was decided to only fine-tune the BERT model using readily available labelled datasets on the internet and not to pretrain it with a financial corpus dataset. This was because,

- There is no open-source dataset with Reddit corpus
- As mentioned in the literature review chapter, the datasets with financial corpus were too large (in the order of Gigabytes of data) for the resources at our expense.

Also, as discussed in the literature review chapter, a BERT model just fine-tuned with sentiment classification task performs better compared to other models. Because no labelled Reddit datasets are available, it was decided to use readily available open-source datasets. To overcome any negative effects due to this multiple BERT models were used in the study. A total of four BERT models were used,

- Three of which were fine-tuned with datasets mentioned in the below Table 3.2
- The fourth model was chosen as the readily available finBERT model from huggingface library (ProsusAI n.d.). This finBERT model was pre-trained and fine-tuned with financial corpus and financial sentiment classification. This model is the implementation of BERT described in (Araci 2019).

Table 3.2: Datasets Used To Fine-Tune The Different BERT Models In The Study

Model name used in this study	**Base model**	**Dataset used for fine-tuning**	**Sentiment Labels In The Dataset**
BERT Model 1	BASE BERT	Airline sentiment dataset from Kaggle (after normalizing data count for positive, negative, and neutral labels) (Kaggle 2019)	Positive, Neutral, Negative
BERT Model 2	BASE BERT	Airline sentiment dataset from Kaggle (without normalizing data count for positive, negative, and neutral labels) (Kaggle 2019)	Positive, Neutral, Negative
BERT Model 3	BASE BERT	Sentiment140 Twitter dataset (Go et al n.d.)	Positive, Negative
BERT Model 4	finBERT	Not Applicable since finBERT is a ready-to-use model already trained and fine-tuned. It output probabilities for positive, neutral and negative sentiments.	

3.3.2 Data Pre-Processing For Fine-Tuning Bert

- URLs, additional spaces, and character references were removed from the fine-tuning datasets.
- The fine-tuning datasets were then split into training and validation datasets where 25% of the data was used for validation. No testing data was required as this is only a fine-tuning operation, and no prediction would take place on these datasets.
- The split data were then tokenized and encoded
- Tensors were created from the above
- Finally, Pytorch dataloaders were created from the tensor datasets

All NLP tasks involve tokenizing and encoding. Since neural networks and other mathematical models cannot work on text data, encoding is necessary to convert strings to a numerical format (Rafail & Freitas 2020). Tokenization is the process of splitting an input into smaller units (e.g., paragraphs to sentences and words).

Tensors and dataloaders are formats of the inputs data which improve resource efficiency while training and predicting. Tensors are a format of data like arrays, but they are immutable and are highly efficient when models are run in a Graphic Processing Unit (GPU) (Pytorch n.d.b). Dataloaders make it easy to load data in batches and take advantage of parallelization; it creates an iterable dataset from the given input, which can be used efficiently for both training and predictions without clogging a systems memory resource (Pytorch n.d.a).

3.3.3 Fine-Tuning Of Bert Models

The BERT Models 1,2 and 3 were fine-tuned with their respective datasets. Though the term fine-tuning may seem like a different process, it is the same as training a model with labelled training data. During this process, each model was fine-tuned with different parameter values to find the best combination of parameter values which yielded the best results with the validation dataset.

To find the best parameters to use for the BERT models, the parameters from the original BERT research study (Devlin et al 2019) were considered as the options. Batch sizes of 16 and 32 along with learning rates of $1e^{-5}$, $2e^{-5}$, $3e^{-5}$, $4e^{-5}$, $5e^{-5}$ were used to find the best parameter combination. AdamW was used as the optimizer function as it is the most widely used general-purpose optimizer function.

3.4 Sentiment Predictions With Bert Models

After the BERT Models 1, 2, and 3 were fine-tuned, all the BERT models (including BERT Model 4 which does not require fine-tuning) were used to find the sentiments of the comments extracted from Reddit and Twitter.

3.4.1 Pre-Processing Of Reddit And Twitter Data

In the case of Reddit, the following data pre-processing and cleaning operations were done,

- Filtering comments which have the ticker symbol for each stock in the format '$<ticker>' or '<ticker>'
- Filtering comments which talk about only one stock or ticker
- Filtering only top-level comments
- Removing URLs
- Removing additional spaces
- Removal of character references – for example '&amp' for ampersand
- Removal of additional square brackets

For Twitter data, apart from the above-mentioned changes, the Twitter ID handle which starts with a '@' symbol was also removed. Emojis were not removed from both datasets as BERT is capable of handling emojis.

After data cleansing operation, comments which had a length of over 512 were split into multiple comments each with a maximum length of 512. This was done because BERT can only handle a maximum input length of 512. This does not create any impact since sentiments from all comments for a ticker on a single day are averaged out to find the final sentiment value for the day. For example, the probabilities of the positive, negative, and neutral sentiments for stock $AMC on any single day will be the corresponding average of positive, negative, and neutral sentiments of all comments on that day. This step was not necessary during the fine-tuning process as all the fine-tuning datasets were from Twitter and hence had a word limit for each tweet.

The datasets are then encoded and tokenized from which tenor datasets are created. Finally, dataloaders are created from the tensor datasets which is used as the input for BERT model.

3.4.2 BERT Output

BERT computes the probabilities of positivity, neutrality, and negativity for each post or comment. Thus, to get the sentiments of posts related to a stock for the entire day, the sentiments of all the posts for the entire day for that stock were averaged. For example, the positive, negative, and neutral sentiments of $AMC will be the average of positive, negative, and neutral probabilities of all posts and comments on $AMC for that day.

So, for a stock or ticker, on any given day, the positive, negative, and neutral probabilities are given as,

$$BSI_{pos} = \Sigma POS_i / N$$

$$BSI_{neg} = \Sigma NEG_i / N$$

$$BSI_{neu} = \Sigma NEU_i / N$$

- $BSI_{pos,}$ BSI_{neg} and BSI_{neu} are the average positive, negative, and neutral sentiment scores with respect to the total number of comments.
- POS_i, NEG_i, and NEU_i are the scores of positive, negative, and neutral sentences.
- N is the total number of comments in the entire dataset for the day.

Hiew et al (2019) and other studies have used just one composite output from BERT (i.e., if positive sentiment had the highest probability, then only the average positive sentiment probability would be used), but as we have a CNN model, it is beneficial to provide more data to the model and allow the model to value the importance of the different sentiments.

The below figure 3.5 shows how three sets of output are derived from each BERT Model for each stock.

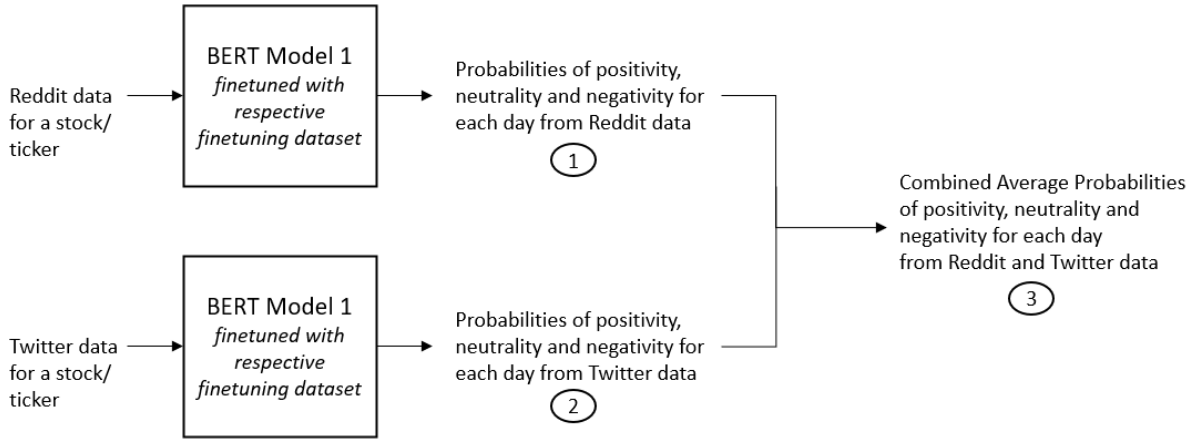


Figure 3.5: Outputs From Each BERT Model

We get three sets of outputs from each BERT Model for every stock,

1. Probabilities of positive, negative, and neutral sentiments from the Reddit data for each day
2. Probabilities of positive, negative, and neutral sentiments from the Twitter data for each day
3. Corresponding sentiments from both Reddit and Twitter sentiments are averaged to get combined probabilities for positive, negative, and neutral sentiments

The only difference in the outputs of the BERT Models apart from the probabilities is that, with BERT Model 3 which was trained with Sentiment140 dataset, only positive and negative sentiments can be derived since sentimet140 dataset does not have neutral label in its dataset. Since BERT Model 3 was trained with sentiment140 dataset, it only outputs two sentiments viz. positive and negative. Though the neutral sentiment is missing, there are still three sets of outputs from BERT Model 3 as with other models.

Each of these output sets were then combined with the technical inputs to form three sets of inputs. This is explained in the following sections.

3.5 CNN-LSTM Model

The below figure 3.6 gives an overview of the prediction process after BERT outputs are obtained,

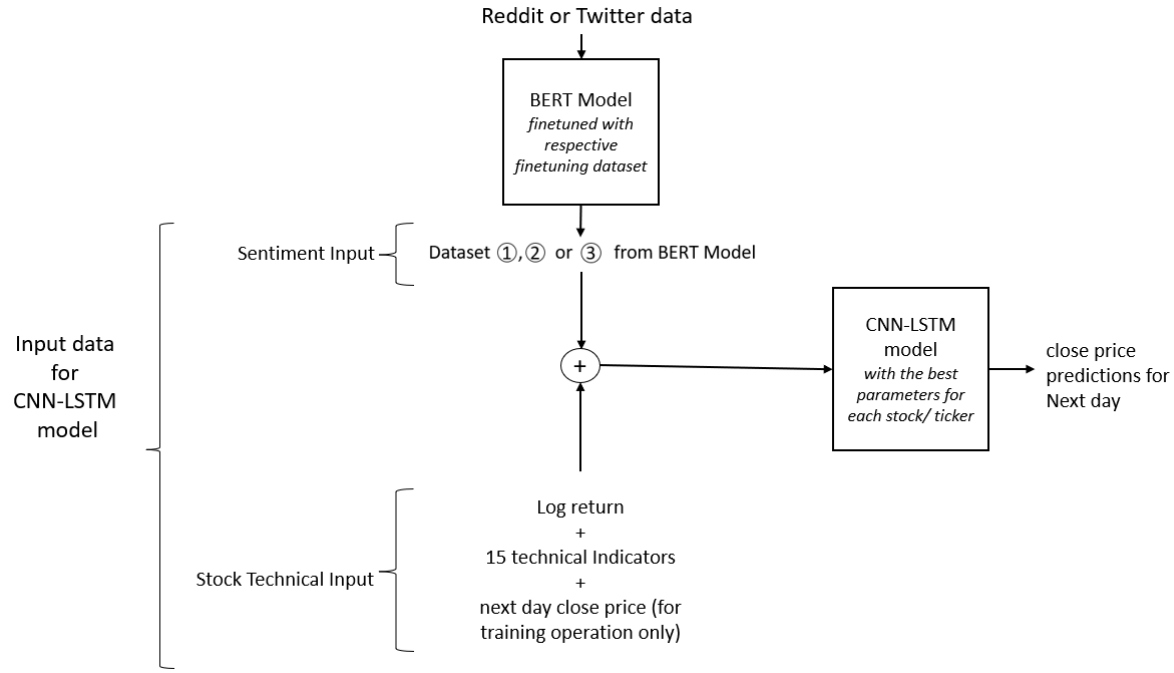


Figure 3.6: Process Of Stock Price Prediction Through CNN-LSTM Model Using Technical Inputs And Sentiment Inputs From BERT

Prior to prediction, there were several steps done to pre-process the input data, finding the best parameters for the CNN-LSTM model, and then training and predicting the next day close prices with the trained model.

3.5.1 Normalizing Timeframes For Input Data

There were discrepancies in the dates where data was available in Reddit and Twitter due to the following reasons,

- On both Reddit and Twitter, data were found even on non-trading days. Because stock prices were not available for such days, these data could not be used in the study i.e., there would be no technical indicator data to combine with sentiments from such days.
- Pushshift archive does not have data for a few days
- In some cases, Twitter did not have data whereas Reddit had data for those dates (e.g., $AMD) and in some cases it was vice versa (e.g., $AMD).

To overcome this difference, the sentiment output datasets from the BERT Models were filtered so that data on the same dates were available on Reddit, Twitter, and combined output sets. This was the most suitable solution as this study focusses on comparing the effectiveness

of Reddit and Twitter sentiments in predicting stock prices. Hence having equal footing among the data was essential.

The below figure 3.7 shows the data availability on Reddit and Twitter individually and after normalizing the dates and accounting only for trading days.

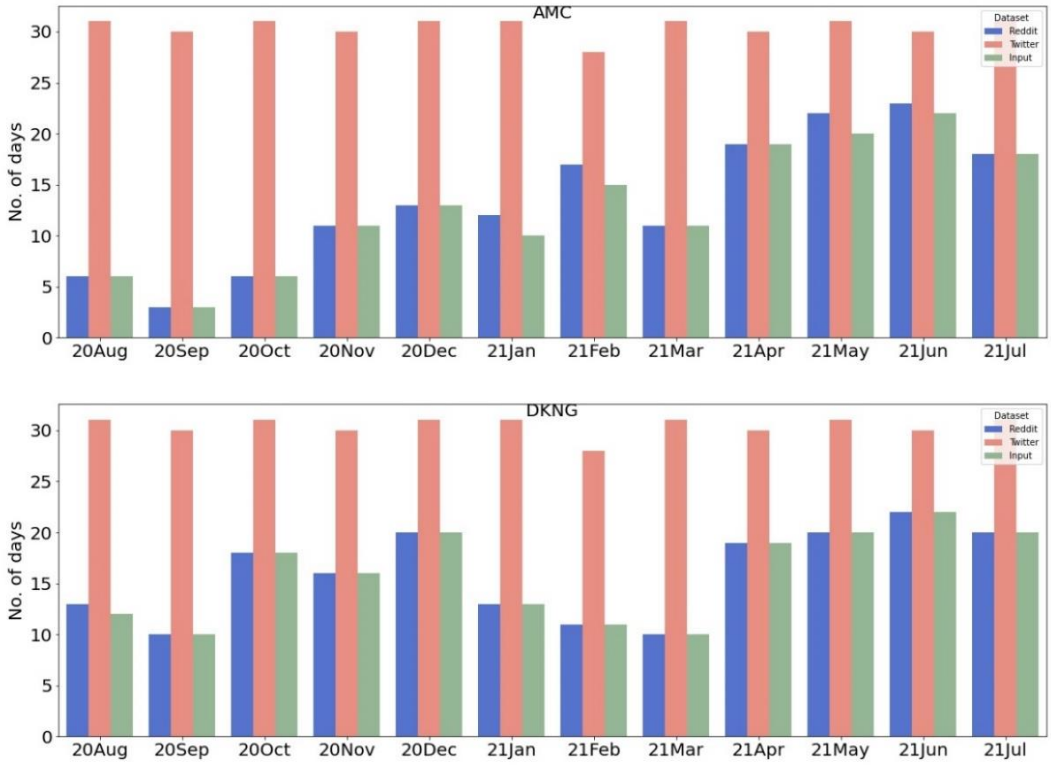


Figure 3.7: Data Availability Of $AMC And $DKNG Post Normalization Of Dates In Reddit And Twitter Datasets

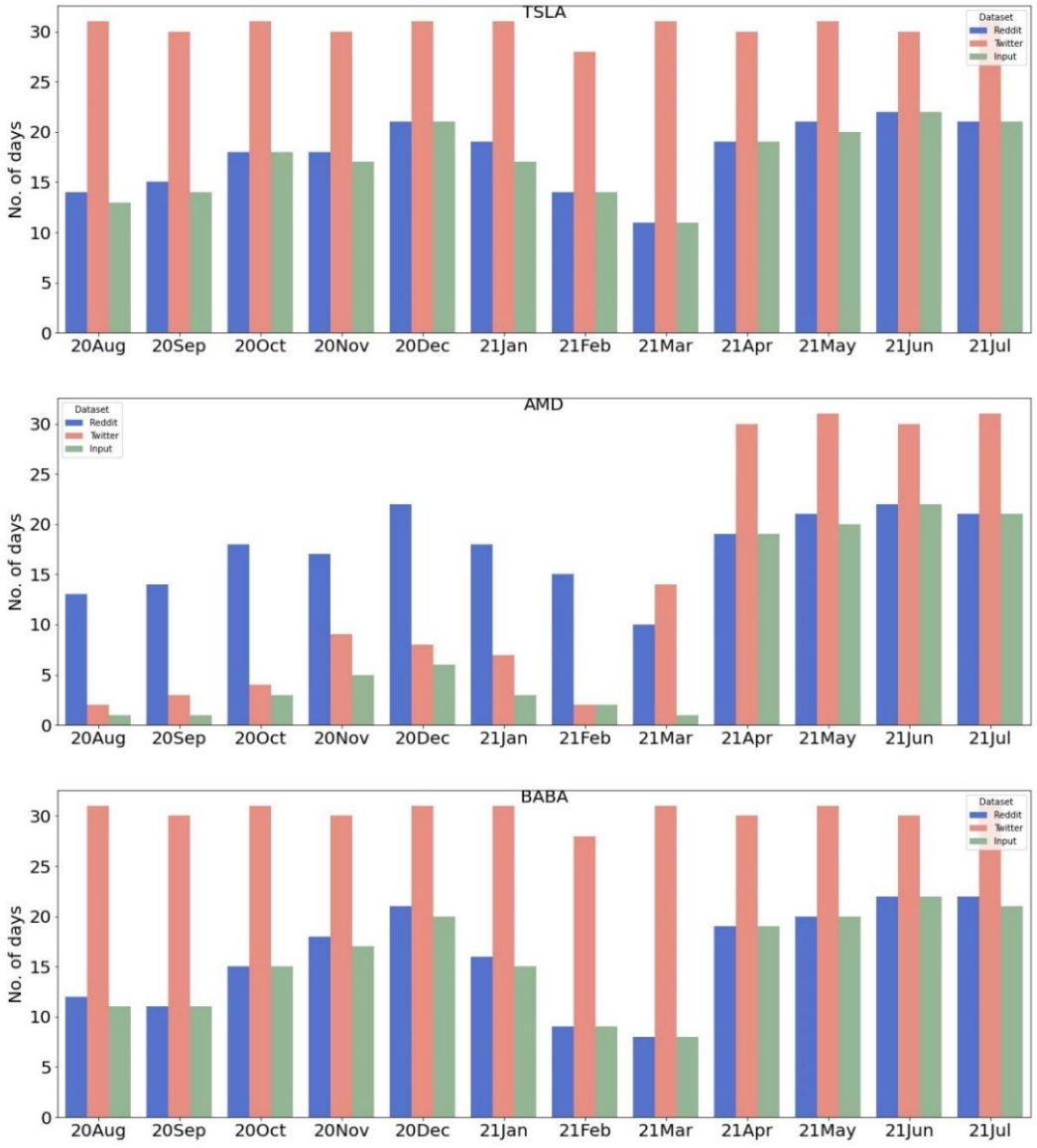


Figure 3.8: Data Availability Of $TSLA, $AMD And $BABA Post Normalization Of Dates In Reddit And Twitter Datasets

3.5.2 Input Data For The CNN-LSTM Model

Since stock price is non-stationary data, using current day's closing price as one of the inputs would lead to inconsistency. To convert the non-stationary data into stationary data, the current day's log return on stock price was instead used. (Ou & Wang, 2009) mention how using log returns removes trend pattern from stock prices thus making it more stationary.

Along with the current day's log return, the 15 technical indicators and the sentiment probabilities for positive, negative, and neutral sentiments were used as the input for CNN-

LSTM model. Thus, the input data contains 19 features. The closing price of the next day was used as the output variable.

The below table 3.3 shows the three sets of inputs used for CNN-LSTM model for prediction of output, for the case of one stock and one BERT Model,

Table 3.3: Three Sets Of Input Data From Each BERT Model For Each Ticker

Input Set	**INPUTS**			**OUTPUT**
	Technical inputs		**Sentiment inputs**	
1	Log return for day N	15 Technical indicators calculated for day N	positive, negative, and neutral sentiments for each day computed from **Reddit dataset** through BERT Model 1	Close price of day N+1
2	Log return for day N	15 Technical indicators calculated for day N	positive, negative, and neutral sentiments for each day computed from **Twitter dataset** through BERT Model 1	Close price of day N+1
3	Log return for day N	15 Technical indicators calculated for day N	combined positive, negative, and neutral sentiments for each day computed from **combining Reddit & Twitter sentiments** through BERT Model 1	Close price of day N+1

3.5.3 Data Processing Of Input For CNN-LSTM

Input data pre-processing steps which were performed before feeding into the CNN-LSTM model are as follows,

- The data was windowed into lengths of 5 and steps of 1
- Then the data was split into training, validation, and testing dataset with 72% training data, 11% validation and 11% testing data
- Tensor datasets were created from the training, validation, and testing datasets

3.5.4 Hyperparameter Tuning

Since each stock has its own time series, the CNN-LSTM model was created in a parameterized way and the best hyperparameters for each stock were found. GridSearchCV module from scikit-learn python library was used to find the best parameters for each stock. Since the technical inputs are always the same for each stock and only the sentiments vary, a single set of inputs for each stock consisting of [Technical inputs (as mentioned in Table 3.3 above + sentiments from Reddit through BERT Model 1] was used to determine the best hyperparameters for each stock.

The research by (Lu et al 2020, Livieris et al 2020 and Mehtab & Sen 2020) were used as reference for this. Each of the studies included CNN-LSTM model for financial series prediction and had some details about the hyperparameters which gave the best performance. These parameters along with a few other options were used to get the best hyperparameters for each stock.

3.6 Prediction And Performance Comparison

3.6.1 Selection Of Performance Metric

As discussed in the literature review chapter, among the metrics widely used in CNN-LSTM models and financial time series prediction models in general, Root Mean Square Error (RMSE) was best suited for this study. To compare how sentiments from Reddit and Twitter affect the prediction accuracy of stock prices, using RMSE will be the most apt choice as it would penalize larger errors more than small prediction errors while also having the same scale as the data.

3.6.2 Stock Price Prediction With CNN-LSTM Model

- As mentioned in Table 3.3, we get three sets of input datasets for the CNN-LSTM model from each ticker and each BERT Model.
- We processed each input dataset as mentioned in the section 3.5.2
- Additionally, two copies of the training data were created, one shuffled and used for the training process (since windowing concept is used, shuffling does not affect the time series learning for the network), another copy was unshuffled and used for

predictions after training. This way, the entire stock price timeline was predicted while not providing the exact same sequence of data as the training data to the model again for prediction. Validation and testing data would be completely new to the network for prediction as the network does not use them for learning during its training process.

- Then the CNN-LSTM model was trained using the shuffled training dataset
- Finally, the closing price for next day was predicted for the entire timeframe of the input.

3.6.3 Performance Comparison With Metrics

After training and predicting using data from all the BERT Models for all the five stocks, the performance of the sentiment data from Reddit, Twitter, and combined sentiments were compared using the RMSE values. This is in line with the research objective OBJ1 to measure the prediction accuracy when using sentiments from different sources.

For performance comparison, it was assumed that an investor would invest in all 5 stocks using each BERT Model. The average RMSE value was computed for the prediction of all five stocks through each BERT Model while using sentiments from Reddit, Twitter, and combined sentiments.

3.6.4 Performance Comparison With Scenarios

To create an additional aspect of performance comparison, we assumed two scenarios where $100 would be invested at the start date of each stock's data and calculated how much profit or loss an investor would achieve by using the predictions in the case of Reddit, Twitter, and combined sentiments. This was necessary because lower error metrics although useful to compare model performance might not always yield the highest profits. In their highly cited work, Leitch & Tanner (1991) found that conventional error metrics did not always provide forecasts with the highest profits. They also found that profits from forecasts are closely related to directional accuracy than error metrics. This formed a strong background in the decision to include the following scenarios for comparisons. Comparison using these scenarios fulfill the research objective OBJ2.

Scenario 1:

This scenario assumes a moderately experienced investor who does both buying and short selling of stocks. The investor would initially invest a sum of $100. Each day the investor decides using the predictions for the next day's closing price.

- If the predicted closing price for day n+1 is greater than day n, then the investor buys the stock at the current day's closing price
- If the predicted closing price for day n+1 is less than day n, then the investor borrows the stock at the current day's (day n) closing price and settles the borrowed stocks at the end of day n+1. This is nothing but a short sale. If the prediction is right, the investor earns the difference but if the prediction is wrong and the closing price increases the next day, the investor loses the difference.
- The profit and loss are cumulative across the entire timeline (i.e.) if the initial $100 increases to $120 by day 10, then the next move (buying or short selling) will involve $120.

Scenario 2:

This scenario assumes a novice investor who only buys the stock if the predicted price for day n+1 is higher than the closing price for day n. Like scenario 1, the profit and loss are cumulative across the entire timeline.

We calculated the average profit the investor would make under each scenario by investing $100 in each stock using sentiments from each BERT Model.

4 RESULTS

4.1 Dataset Statistics

Table 4.1 shows the statistics for the entire Reddit and Twitter data, which was extracted from the respective platforms,

Table 4.1: Statistics Of The Entire Reddit And Twitter Datasets

	Both		Reddit			Twitter		
	Minimum Date	**Maximum Date**	**Total Unique Authors**	**Total Comments**	**Number of Days**	**Total Unique Authors**	**Total Comments**	**Number Of Days**
Ticker								
AMC	2020-08-05	2021-07-30	10816	27128	154	11393	31923	154
AMD	2020-08-05	2021-07-30	1670	4895	104	4428	14913	104
BABA	2020-08-05	2021-07-30	1285	2793	188	7359	25837	188
DKNG	2020-08-05	2021-07-30	1015	2246	191	6967	23914	191
TSLA	2020-08-05	2021-07-30	4344	13806	207	19729	77789	207

Tables 4.1 to 4.6 and Figures 4.1 to 4.5 show the statistics for each stock. It was seen that,

- On average, Twitter had more posts on every stock compared to Reddit. A possible explanation for this is due to this study only considering the top-level comments in the Reddit thread and not the replies for it.

- Because Reddit allows down voting (similar to a ‘dislike’, the sum of scores in Reddit data could have negative values

Table 4.2: Descriptive Statistics For $AMC

	Reddit			Twitter		
	Authors Per day	**Comments Per Day**	**Sum of Score Per Day**	**Authors Per day**	**Comments Per Day**	**Sum of Score Per Day**
count	154.00	154.00	154.00	154.00	154.00	154.00
mean	110.99	176.16	1491.66	156.19	207.29	3459.44
std	301.23	544.55	4828.60	83.98	115.02	2906.92
min	1.00	1.00	-69.00	10.00	12.00	3.00
max	1817.00	3410.00	29089.00	380.00	557.00	13345.00

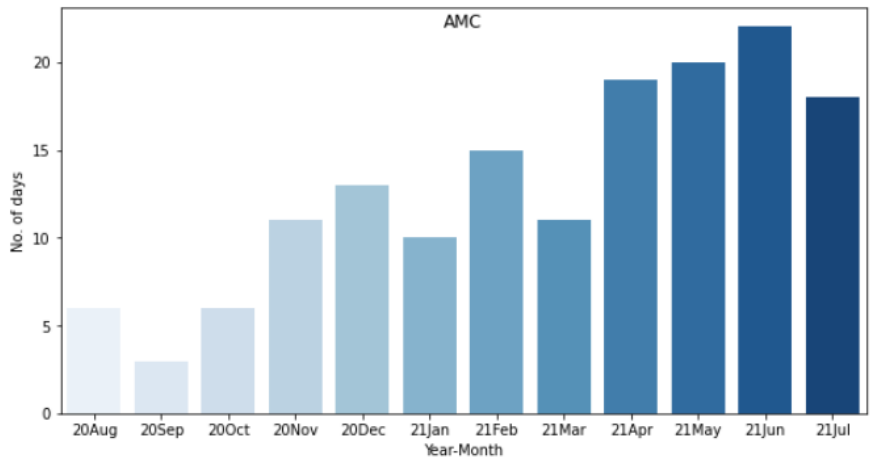


Figure 4.1: Data Availability For $AMC Across Months

Table 4.3: Descriptive Statistics For $AMD

	Reddit			Twitter		
	Authors Per day	**Comments Per Day**	**Sum of Score Per Day**	**Authors Per day**	**Comments Per Day**	**Sum of Score Per Day**
count	104.00	104.00	104.00	104.00	104.00	104.00
mean	34.78	47.07	179.10	99.71	143.39	487.00
std	34.56	52.40	254.08	74.48	103.98	627.94
min	1.00	1.00	-30.00	1.00	1.00	0.00
max	171.00	285.00	1410.00	320.00	427.00	2890.00

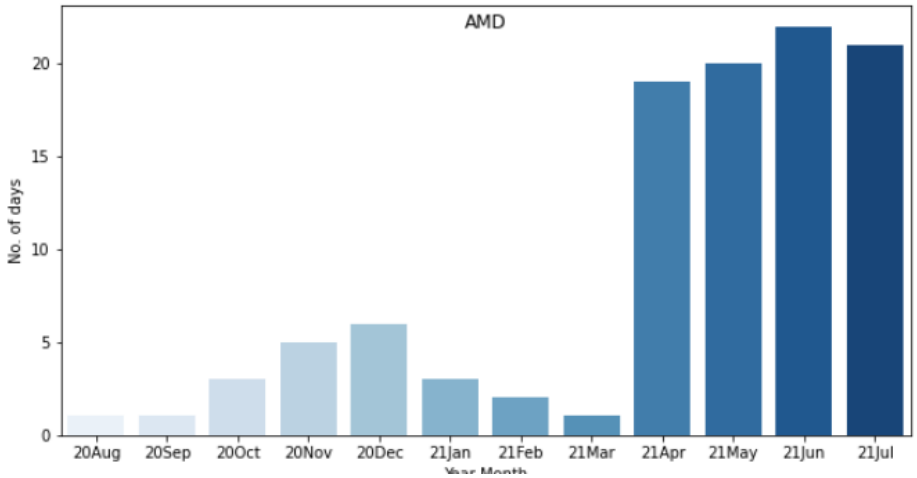


Figure 4.2: Data Availability for $AMD across months

Table 4.4: Descriptive Statistics For $BABA

	Reddit			**Twitter**		
	Authors per day	**Comments Per Day**	**Sum of Score Per Day**	**Authors Per day**	**Comments Per Day**	**Sum of Score Per Day**
count	188.00	188.00	188.00	188.00	188.00	188.00
mean	12.28	14.86	45.30	104.16	137.43	467.90
std	15.17	19.12	66.17	65.12	88.08	462.97
min	1.00	1.00	-14.00	27.00	28.00	22.00
max	120.00	123.00	410.00	349.00	421.00	2119.00

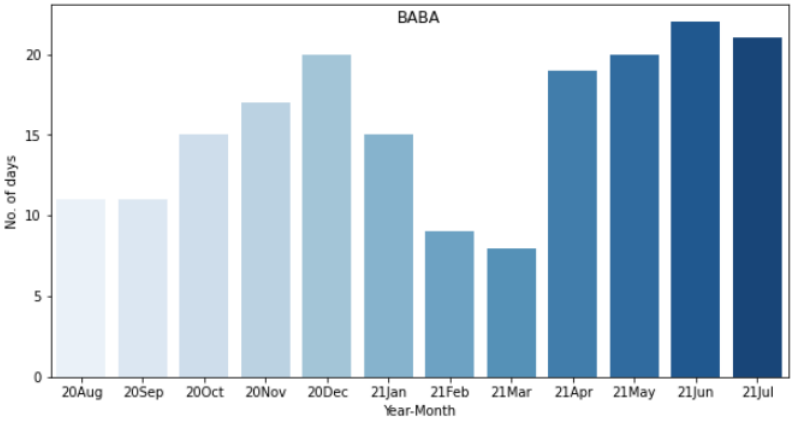


Figure 4.3: Data Availability For $BABA Across Months

Table 4.5: Descriptive Statistics For $DKNG

	Reddit			Twitter		
	Authors Per day	**Comments Per Day**	**Sum of Score Per Day**	**Authors Per day**	**Comments Per Day**	**Sum of Score Per Day**
count	191.00	191.00	191.00	191.00	191.00	191.00
mean	9.01	11.76	54.86	98.51	125.20	553.56
std	19.93	32.85	316.97	60.38	80.95	521.37
min	1.00	1.00	0.00	17.00	23.00	22.00
max	253.00	430.00	4286.00	327.00	458.00	3230.00

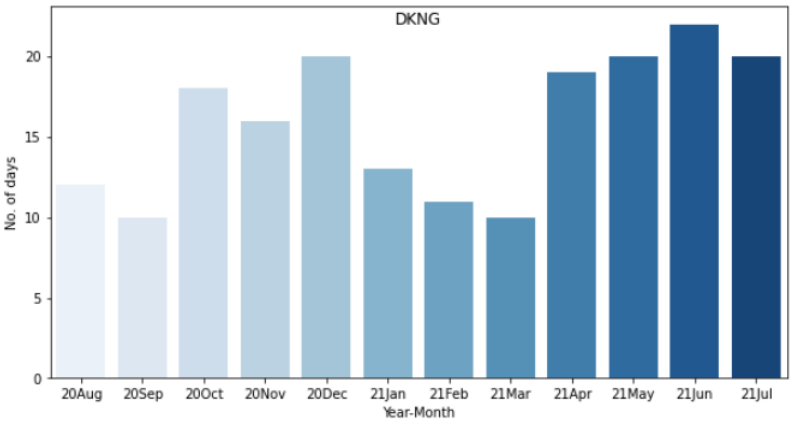


Figure 4.4: Data Availability For $DKNG Across Months

Table 4.6: Descriptive Statistics For $TSLA

	Reddit			Twitter		
	Authors Per day	**Comments Per Day**	**Sum of Score Per Day**	**Authors Per day**	**Comments Per Day**	**Sum of Score Per Day**
count	207.00	207.00	207.00	207.00	207.00	207.00
mean	47.39	66.70	247.13	281.38	375.79	6241.36
std	39.31	68.35	336.07	48.79	57.93	2817.78
min	1.00	1.00	-202.00	138.00	172.00	1910.00
max	242.00	495.00	3237.00	445.00	556.00	20387.00

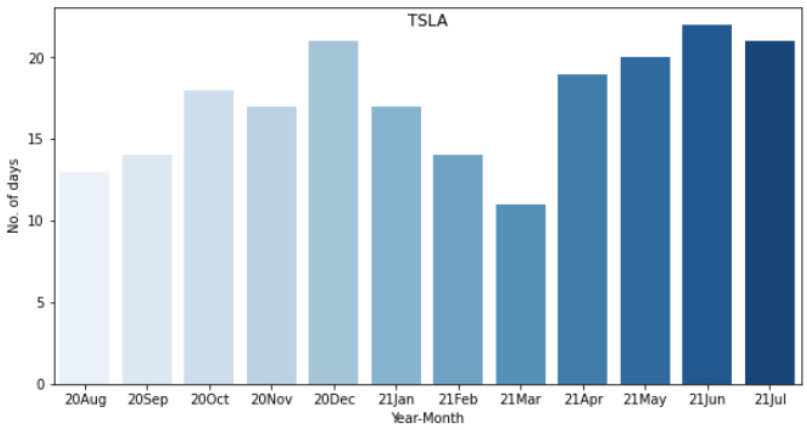


Figure 4.5: Data Availability For $TSLA Across Months

4.2 Tuned Parameters Of BERT Models

BERT Models were tuned using different parameters mentioned in section 3.3.3. Table 4.7 shows the parameters with which each BERT model gave the best performance.

Table 4.7: Metrics For Each BERT Model Which Gave The Best Perofrmance

BERT MODEL	**BATCH SIZE**	**LEARNING RATE**	**EPOCH**
BERT Model 1	32	4e-05	3
BERT Model 2	32	4e-05	3
BERT Model 3	16	4e-05	2
BERT Model 4	Not Applicable. Predictions were done on each comment using the readily available finBERT model from huggingface library.		

4.3 Data Analysis And Visualization

This section discusses the results from BERT Models, which are the sentiment output from each BERT Model using Reddit and Twitter data for each stock.

Figure 4.6 shows a confusion matrix between sentiment predictions of Reddit and Twitter, considering the highest scored probability (among positive, negative, and neutral) as the true

sentiment. Note that for BERT Model 3, only two sentiments are shown since the model was trained using sentiment140 data which had positive and negative sentiments alone (without a neutral sentiment).

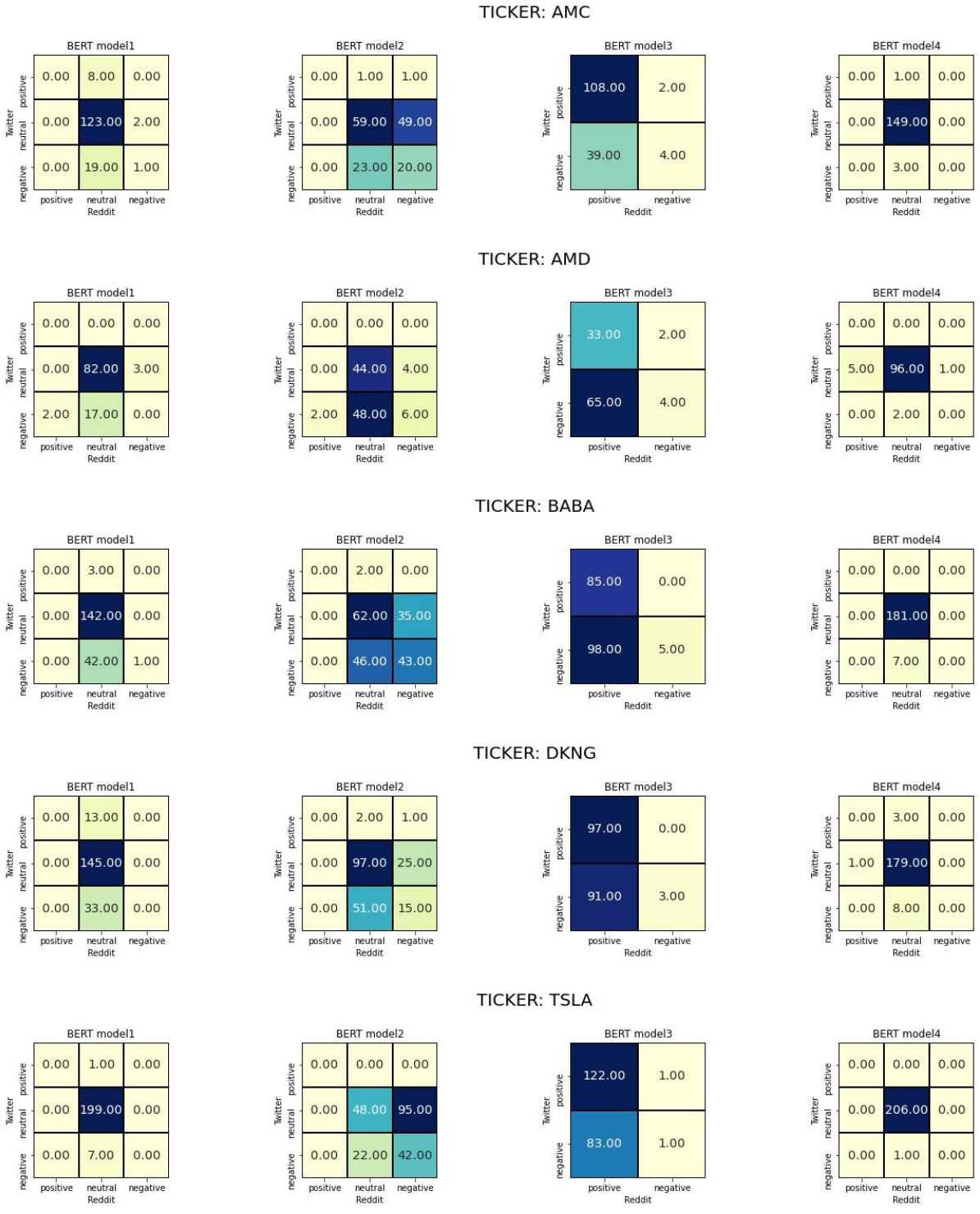


Figure 4.6: Confusion Matrix Taking Highest Probability Score As Sentiment

The below figures from Figure 4.7 to 4.11 show the histogram distribution of probabilities for positive, negative, and neutral sentiments for each stock in each BERT Model. It was noted that, in general, sentiments from Reddit had a wider spread of probabilities compared to Twitter.

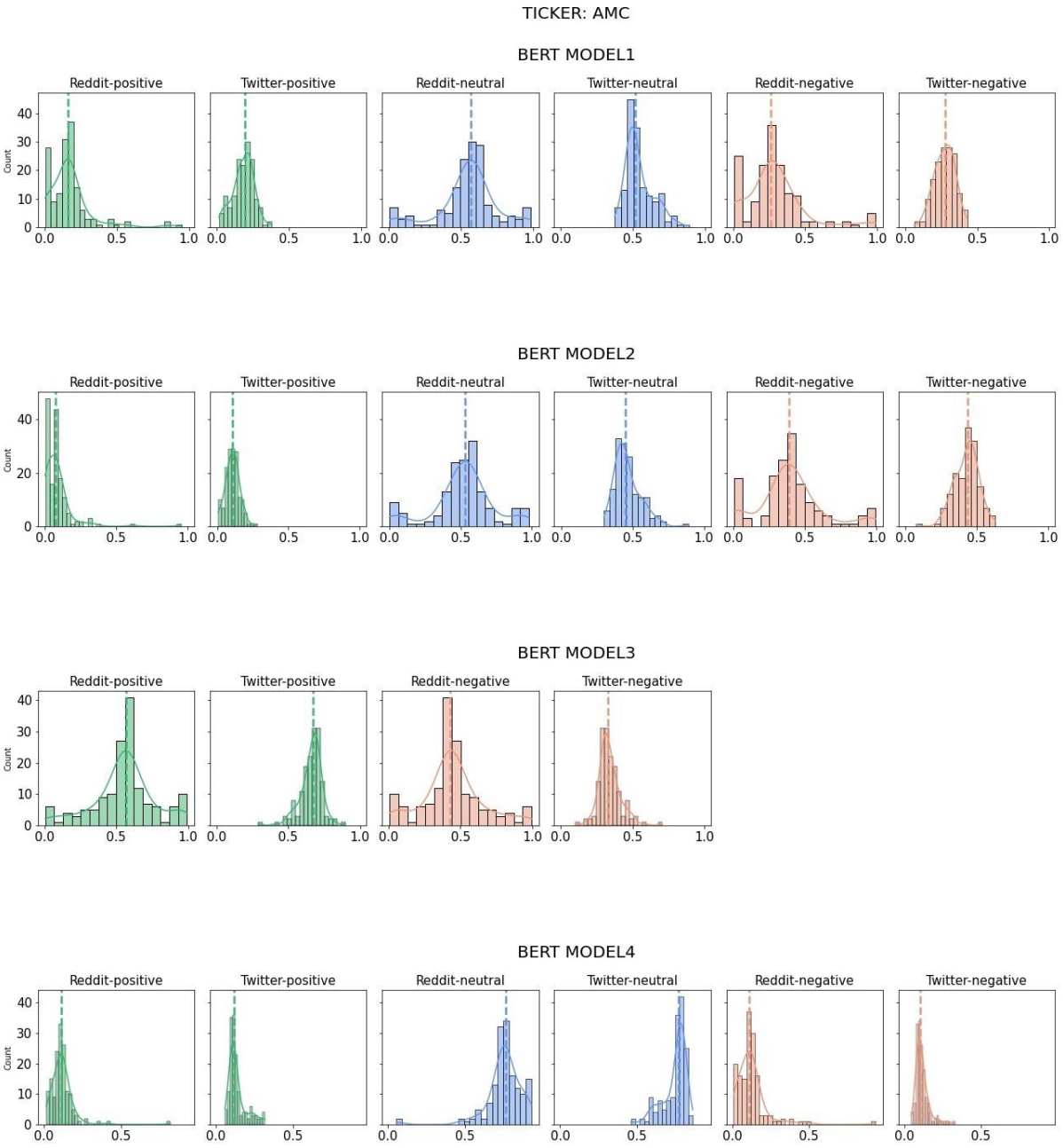


Figure 4.7: Distribution Of Positive, Neutral And Negative Probabilities For $AMC

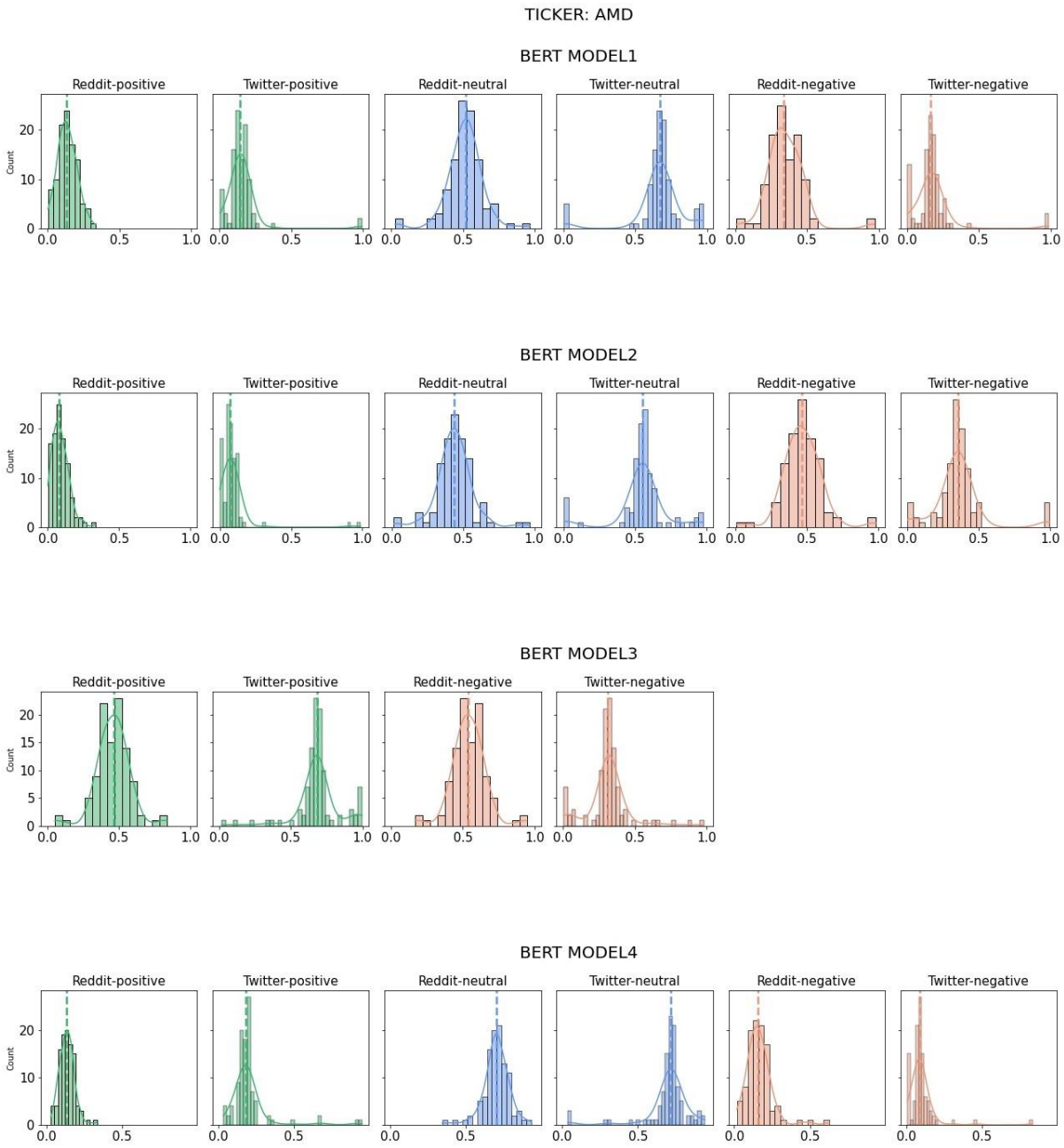


Figure 4.8: Distribution Of Positive, Neutral And Negative Probabilities For $AMD

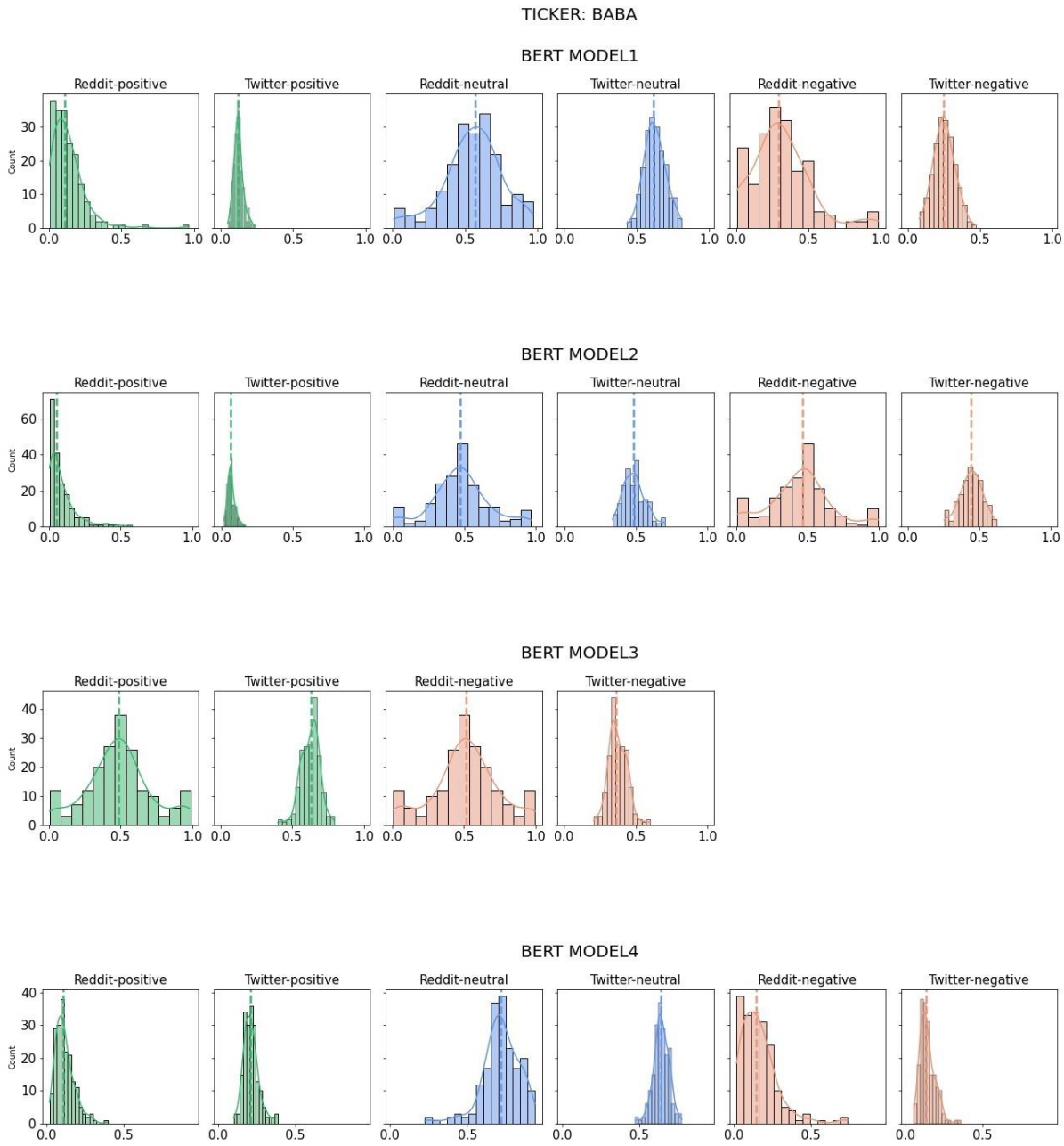


Figure 4.9: Distribution Of Positive, Neutral And Negative Probabilities For $BABA

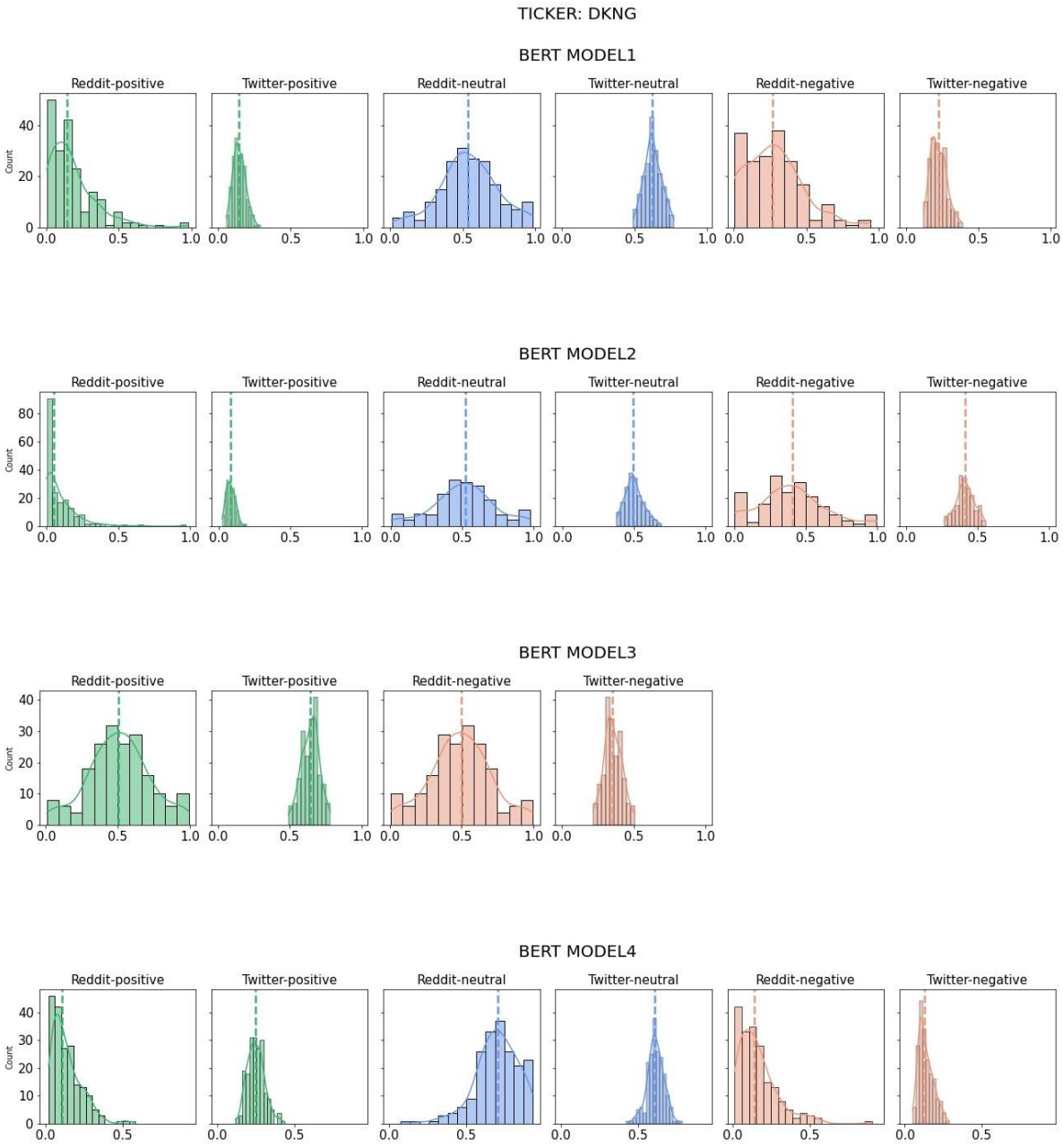


Figure 4.10: Distribution Of Positive, Neutral And Negative Probabilities For $DKNG

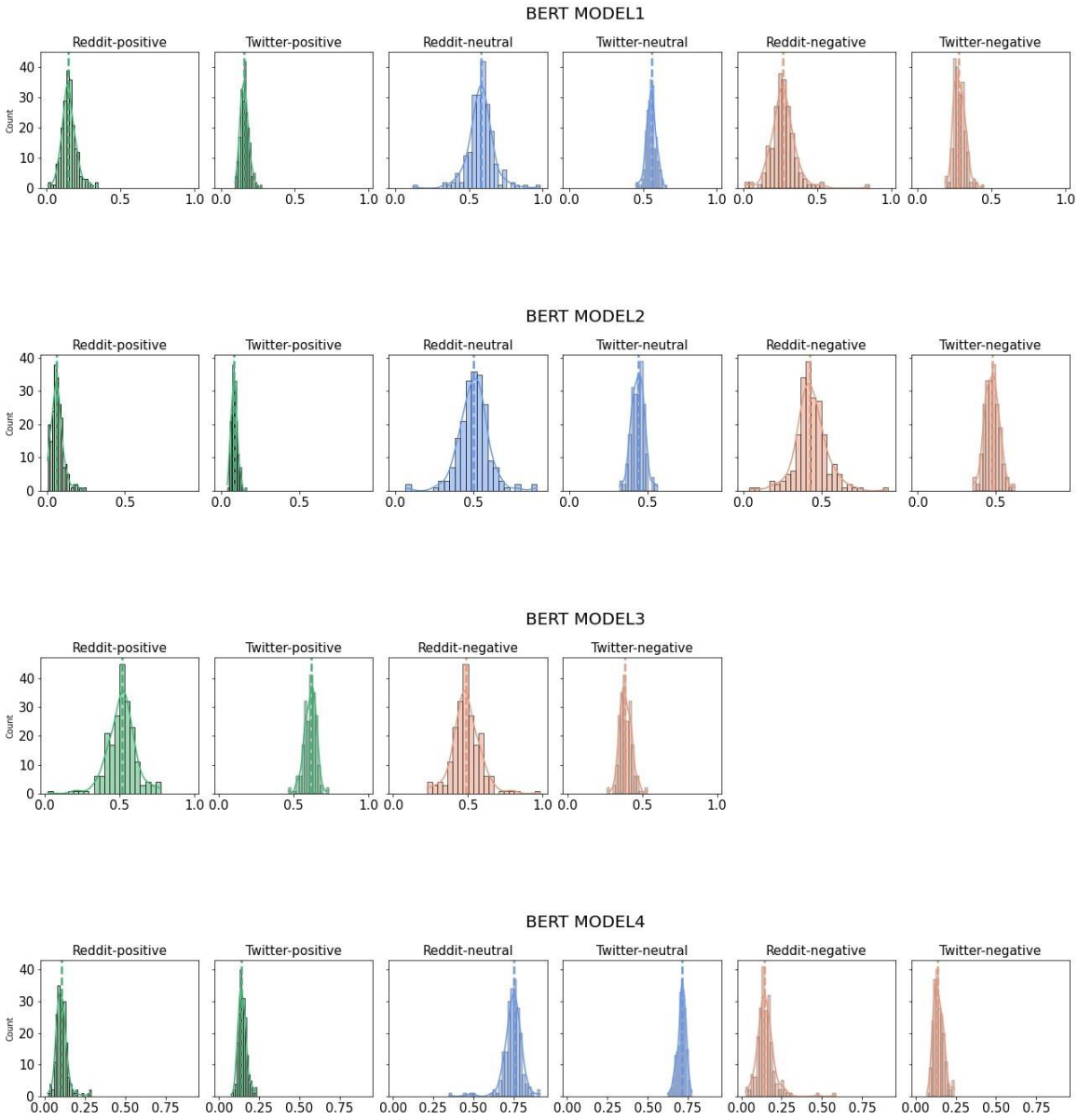


Figure 4.11: Distribution Of Positive, Neutral And Negative Probabilities For $TSLA

The below correlation plots show how Reddit and Twitter sentiment probabilities are correlated in the outputs of each BERT Model. It was clearly seen that there was no correlation between them. Even though the confusion matrices above showed some correlation in the highest probable sentiment in BERT Models 1 and 4, the Pearson's correlation coefficient in each subplot shows that there is no correlation between the corresponding probabilities of Reddit and Twitter.

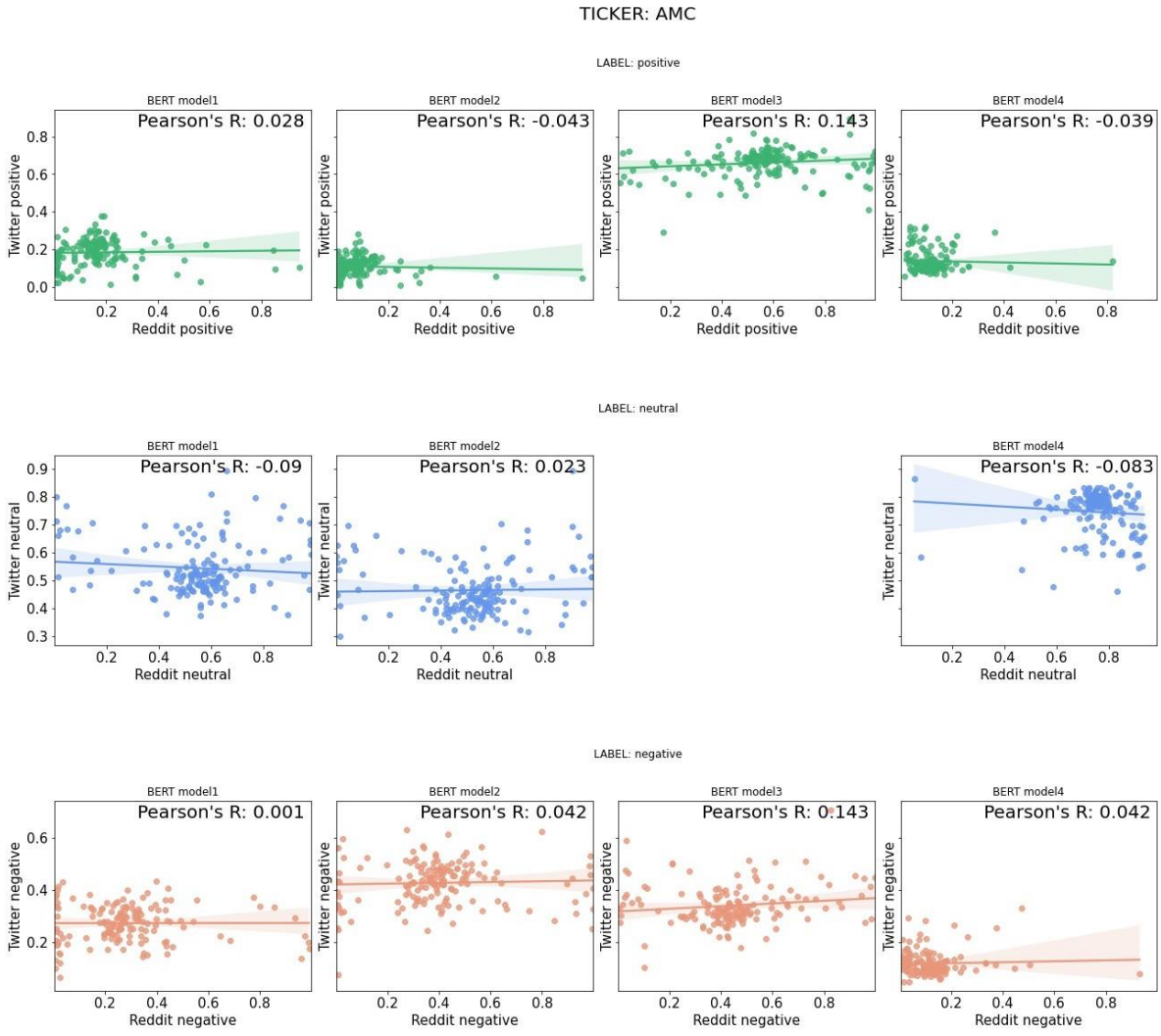


Figure 4.12: Sentiment Probability Correlation Plot For $AMC

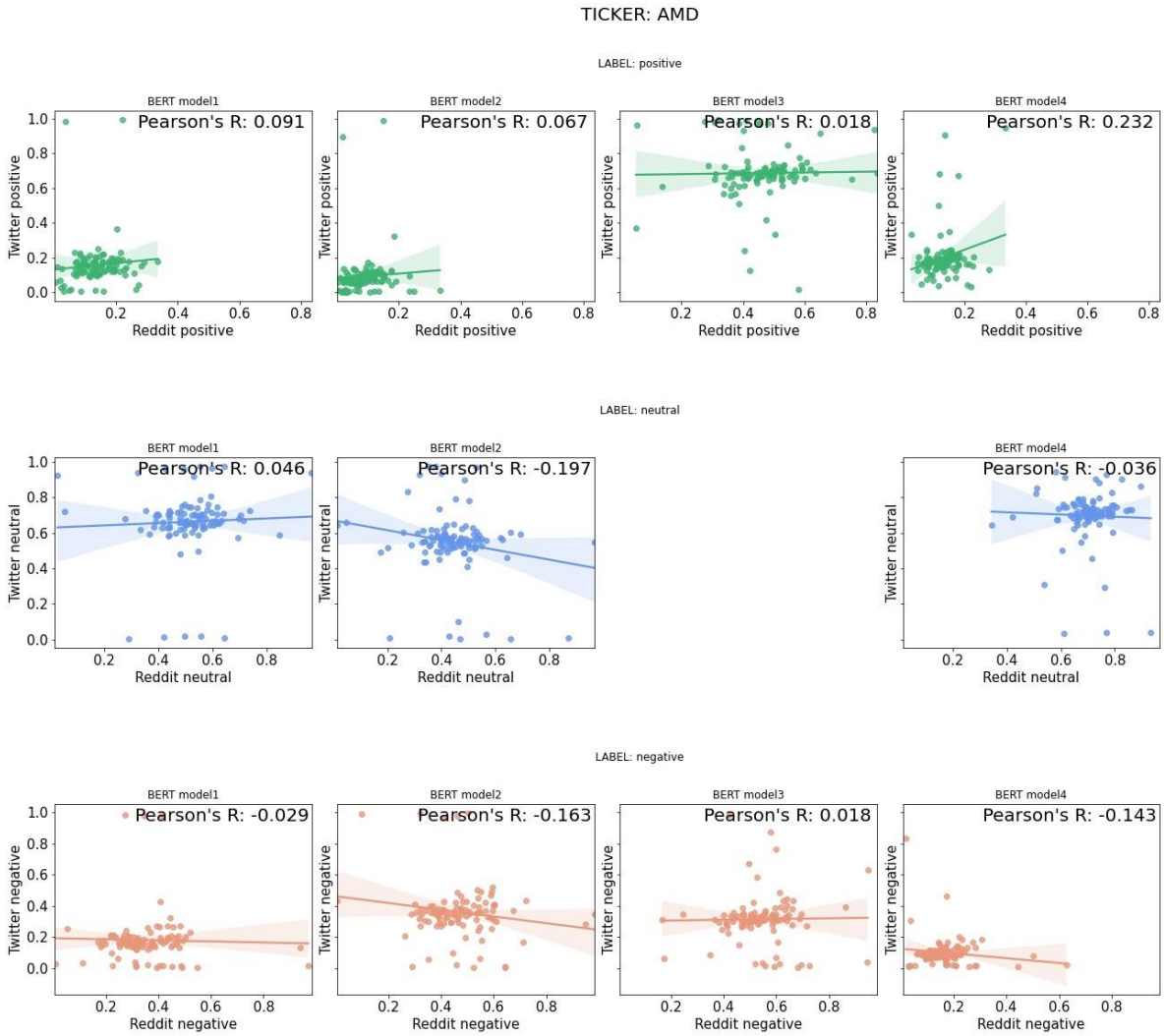


Figure 4.13: Sentiment Probability Correlation Plot For $AMD

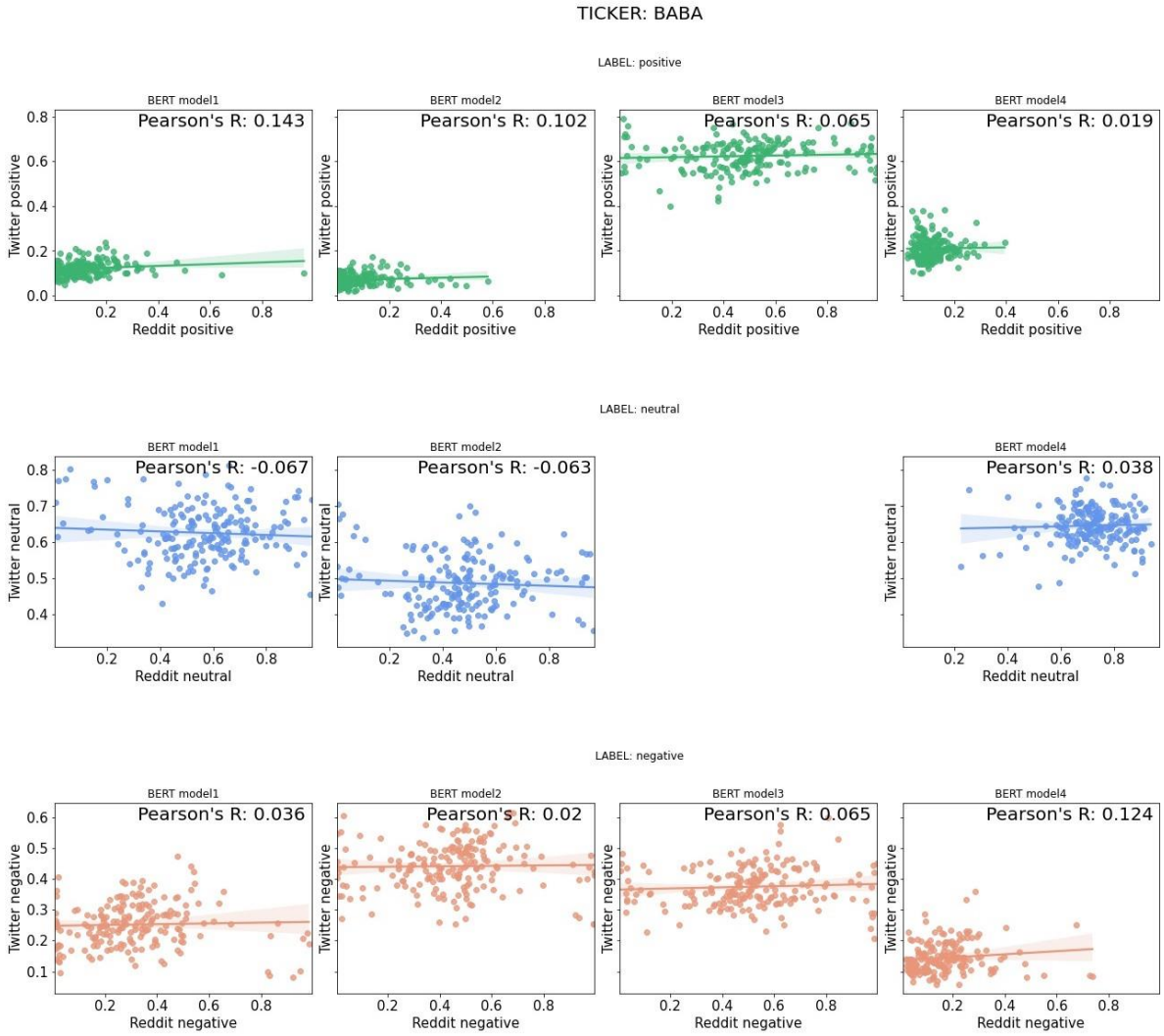


Figure 4.14: Sentiment Probability Correlation Plot For $BABA

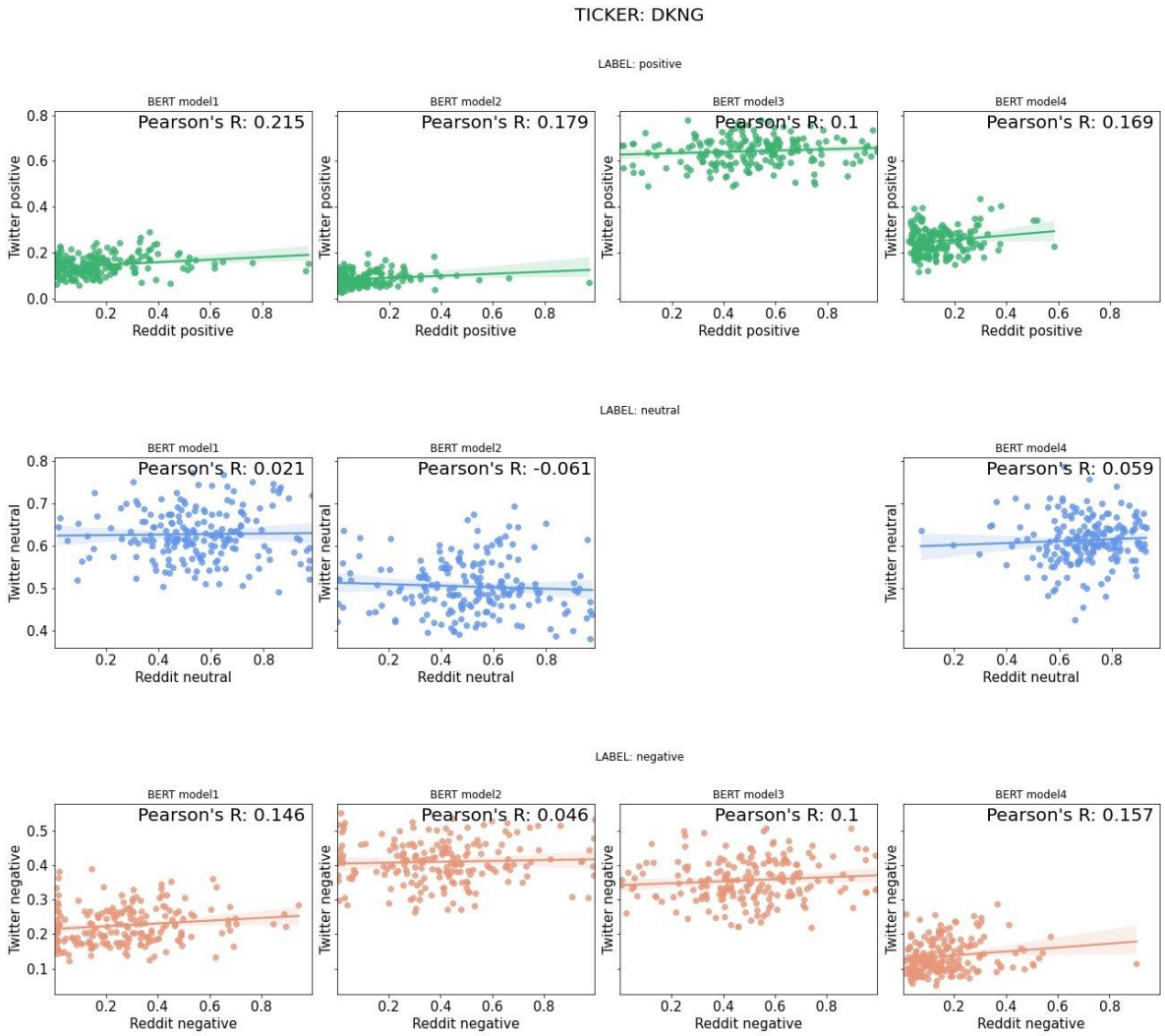


Figure 4.15: Sentiment Probability Correlation Plot For $DKNG

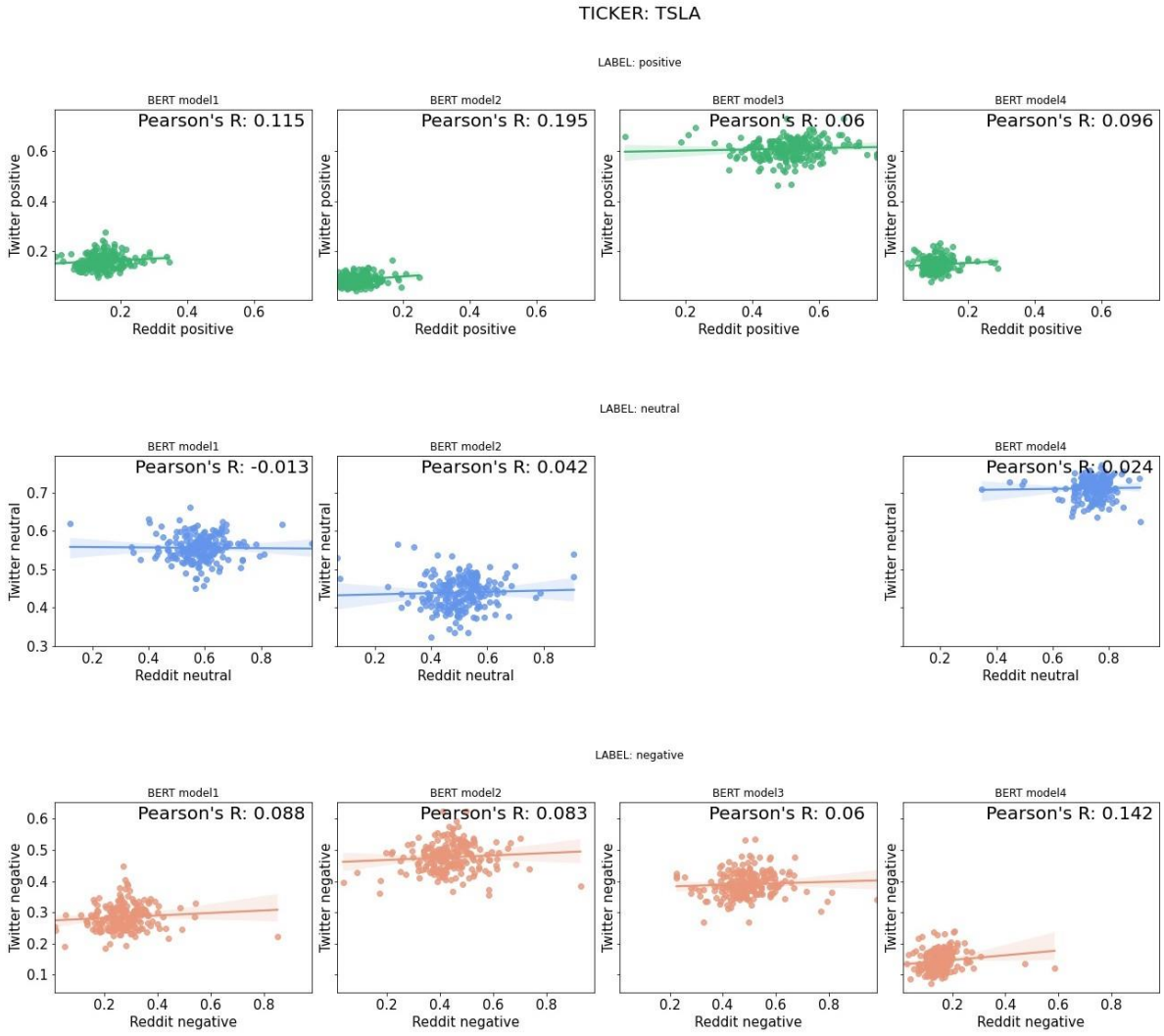


Figure 4.16: Sentiment Probability Correlation Plot For $TSLA

4.4 Parameters For CNN-LSTM Model

The below table 4.8 shows the parameters which gave the best performance for each stock in the CNN-LSTM Model,

Table 4.8: Tuned Parameter Values For Each Stock For The CNN-LSTM Prediction Model

	AMC	**AMD**	**BABA**	**DKNG**	**TSLA**
batch_size	16	5	10	5	10
epochs	50	100	100	100	100
optimizer_fnc	Adam	Adam	Adam	SGD	SGD
learning_rt	1.00E-05	1.00E-05	1.00E-05	0.001	0.001
conv_activation	relu	tanh	tanh	tanh	tanh
conv_filters	64	64	64	64	64
conv_kernel_size	2	2	2	2	2
second_conv_layer	TRUE	TRUE	TRUE	TRUE	TRUE
second_conv_activation	relu	tanh	tanh	tanh	tanh
second_conv_filters	128	128	128	128	128
second_conv_kernel_size	1	2	2	2	2
pool_size	2	2	2	2	2
lstm_units	200	200	200	200	200
second_dense_layer	TRUE	TRUE	TRUE	TRUE	TRUE
dense_units	32	32	200	200	200

where,

- ticker - The ticker symbol for which hyperparameter tuning was performed
- batch_size - The size of the batches in which input is passed to the CNN-LSTM model
- epochs - The number of total passes through the input dataset in training
- optimizer_fnc - The function used to reduce error (finding the minima in gradient descent)
- learning_rt - The rate by which weights in the network can be changed by the optimizer function
- conv_activation - The activation function used in the first convolution layer
- conv_filters - The number of filters in the first convolution layer
- conv_kernel_size - The size of each filter in the first convolution layer
- second_conv_layer - Boolean value: Determines if a second convolution layer is required
- second_conv_activation - The activation function used in the second convolution layer

- second_conv_filters - The number of filters in the second convolution layer
- second_conv_kernel_size - The size of each filter in the second convolution layer
- pool_size - The size of the max-pooling filter in CNN
- lstm_units - The number of nodes or units in the LSTM layer
- second_dense_layer - Boolean value: Determines if a second dense layer is required prior to the final output dense layer
- dense_units - The number of units in the second dense layer

4.5 Predictions

Table 4.9 details the prediction timeframe for each stock. Though all stocks had data from 05 August 2020, because of the difference in the number of days of data present and creating input windows of length 5, the start date of prediction is the 6th day in each stock's dataset. Because $AMC and $AMD had very little data (Refer section 3.5.1) in August 2020, their first prediction day is in September 2020.

Table 4.9: Date Range Of Predictions For Each Stock

Ticker	Start Date Of Prediction	End Date Of Prediction
AMC	2020-09-02	2021-07-30
AMD	2020-11-11	2021-07-30
BABA	2020-08-18	2021-07-30
DKNG	2020-08-17	2021-07-30
TSLA	2020-08-14	2021-07-30

4.5.1 Predictions From CNN-LSTM Using Sentiments From BERT Model 1

Figures 4.17 to 4.21 show the prediction of each stock by the CNN-LSTM model using sentiments predicted by BERT Model 1 from Reddit, Twitter, and a combination of both, respectively. The dashed plot represents the original price and the solid, orange plot represents the predicted price.

TICKER: AMC

Reddit

dataset	RMSE
Train	0.06184
Validation	1.01847
Test	1.14708

Twitter

dataset	RMSE
Train	0.0574
Validation	1.00891
Test	1.27496

Combined

dataset	RMSE
Train	0.06736
Validation	1.18171
Test	1.62624

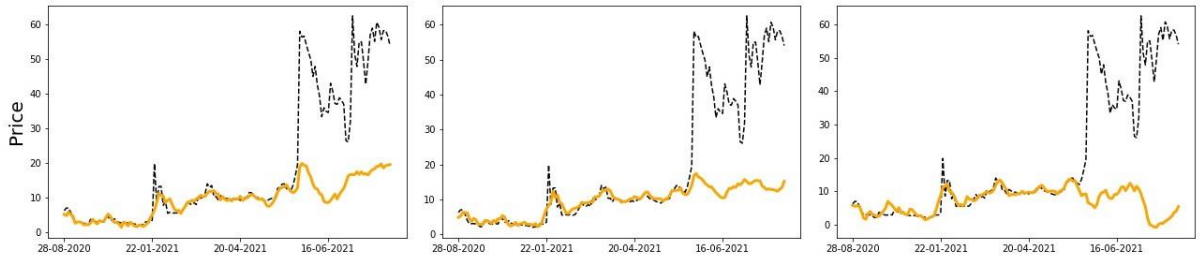


Figure 4.17: Prediction Plot For Price Prediction Of $AMC Using Sentiments From BERT Model 1

TICKER: AMD

Reddit

dataset	RMSE
Train	0.12846
Validation	0.31021
Test	0.14353

Twitter

dataset	RMSE
Train	0.12575
Validation	0.33992
Test	0.14991

Combined

dataset	RMSE
Train	0.12915
Validation	0.37297
Test	0.16689

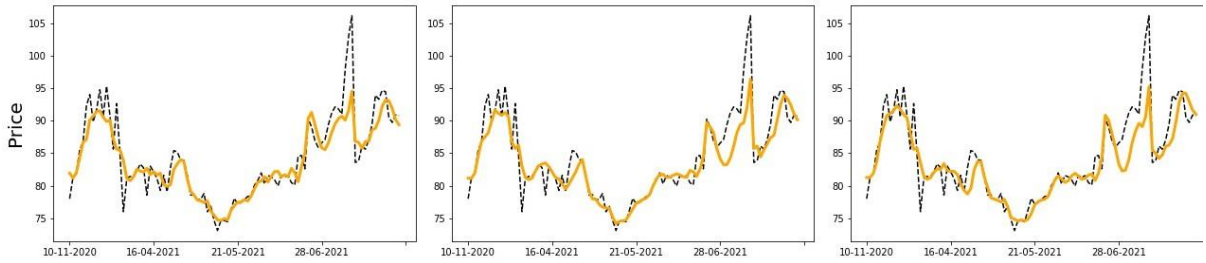


Figure 4.18: Prediction Plot For Price Prediction Of $AMD Using Sentiments From BERT Model 1

TICKER: BABA

Reddit

dataset	RMSE
Train	0.098
Validation	0.14207
Test	0.07801

Twitter

dataset	RMSE
Train	0.08853
Validation	0.13069
Test	0.07055

Combined

dataset	RMSE
Train	0.09422
Validation	0.12507
Test	0.0565

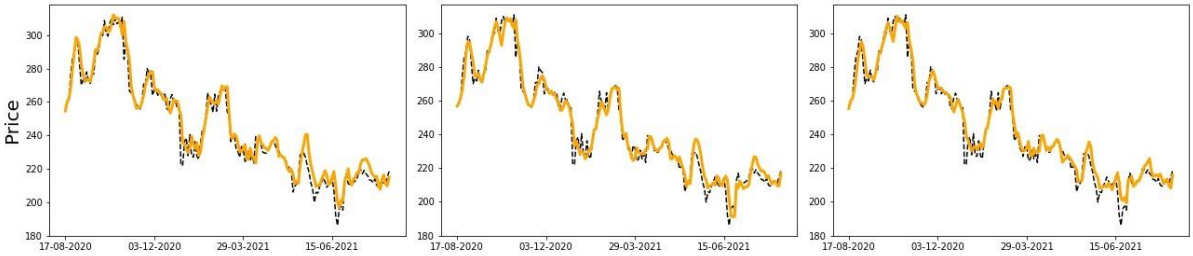


Figure 4.19: Prediction Plot For Price Prediction Of $BABA Using Sentiments From BERT Model 1

TICKER: DKNG

Reddit

dataset	RMSE
Train	0.11127
Validation	0.07825
Test	0.10767

Twitter

dataset	RMSE
Train	0.1132
Validation	0.11211
Test	0.11699

Combined

dataset	RMSE
Train	0.10767
Validation	0.08977
Test	0.09556

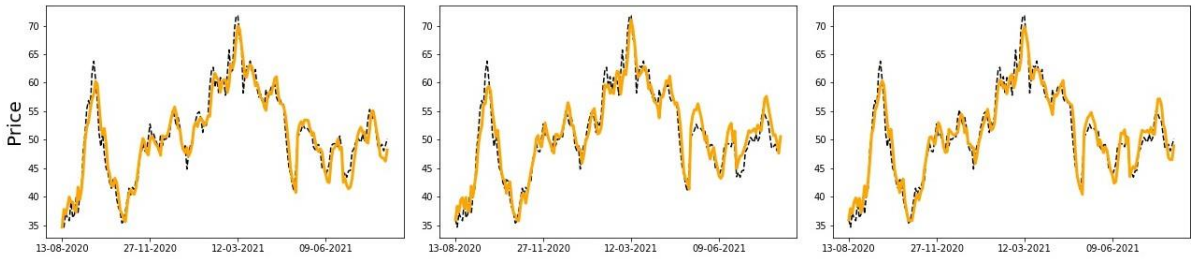


Figure 4.20: Prediction Plot For Price Prediction Of $DKNG Using Sentiments From BERT Model 1

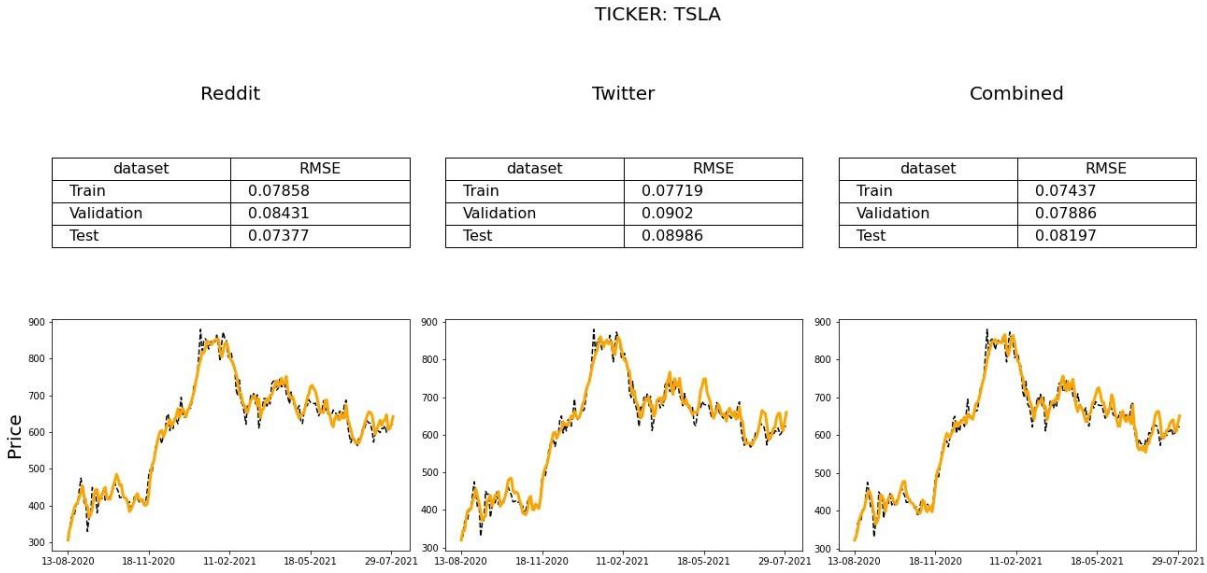


Figure 4.21: Prediction Plot For Price Prediction Of $TSLA Using Sentiments From BERT Model 1

4.5.2 Predictions From CNN-LSTM Using Sentiments From BERT Model 2

Figures 4.22 to 4.26 show the prediction of each stock by the CNN-LSTM model using sentiments predicted by BERT Model 2 from Reddit, Twitter, and a combination of both respectively. The dashed plot represents the original price and the solid, orange plot represents the predicted price.

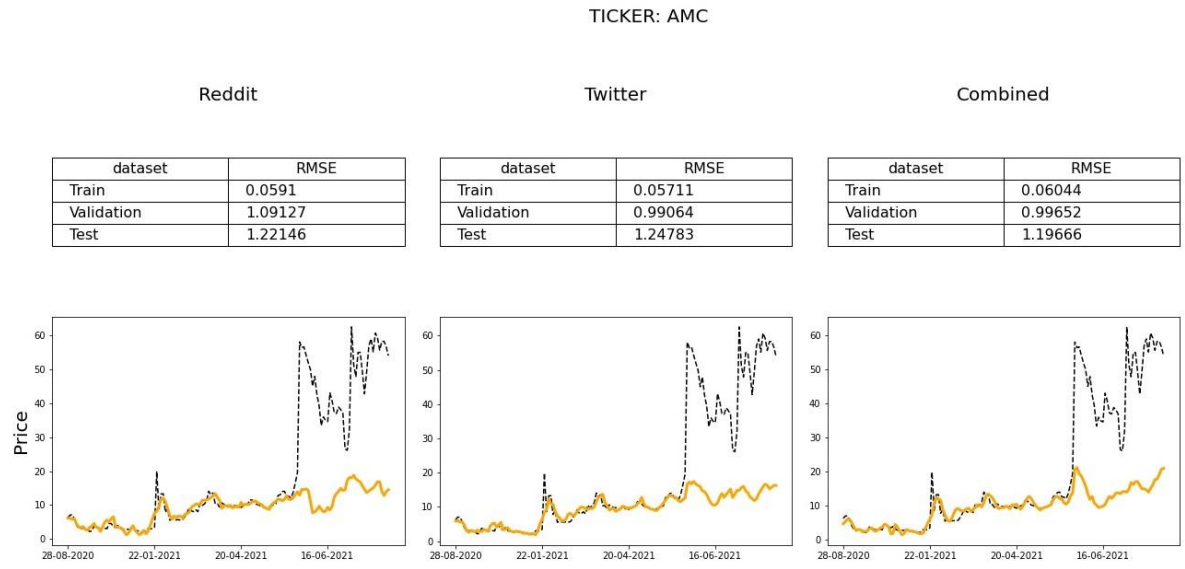


Figure 4.22: Prediction Plot For Price Prediction Of $AMC Using Sentiments From BERT Model 2

TICKER: AMD

Reddit

dataset	RMSE
Train	0.13662
Validation	0.38876
Test	0.16006

Twitter

dataset	RMSE
Train	0.12203
Validation	0.40677
Test	0.18852

Combined

dataset	RMSE
Train	0.12196
Validation	0.44121
Test	0.2221

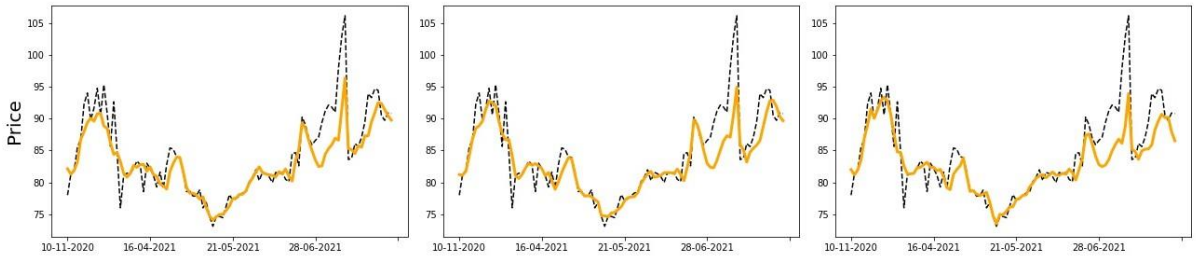


Figure 4.23: Prediction Plot For Price Prediction Of $AMD Using Sentiments From BERT Model 2

TICKER: BABA

Reddit

dataset	RMSE
Train	0.09225
Validation	0.14891
Test	0.07042

Twitter

dataset	RMSE
Train	0.08998
Validation	0.16132
Test	0.11793

Combined

dataset	RMSE
Train	0.08609
Validation	0.18897
Test	0.15856

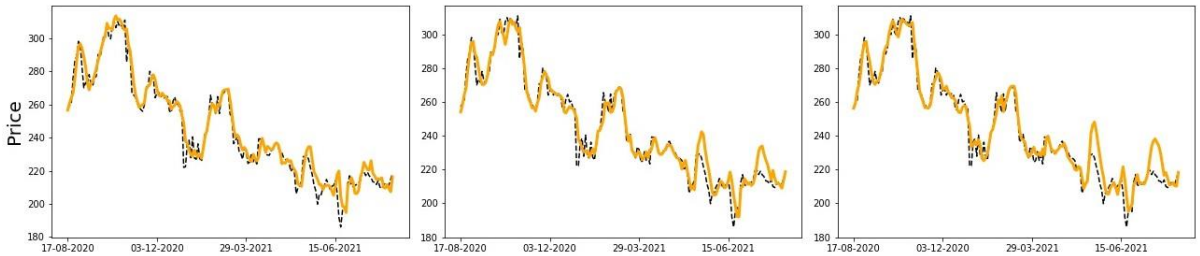


Figure 4.24: Prediction Plot For Price Prediction Of $BABA Using Sentiments From BERT Model 2

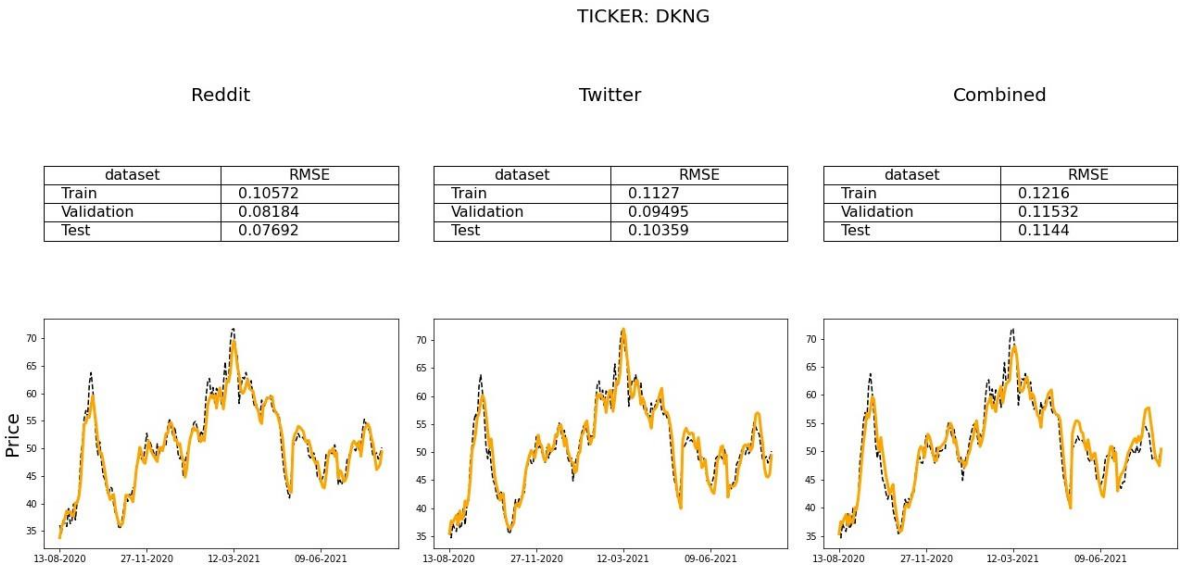


Figure 4.25: Prediction Plot For Price Prediction Of $DKNG Using Sentiments From BERT Model 2

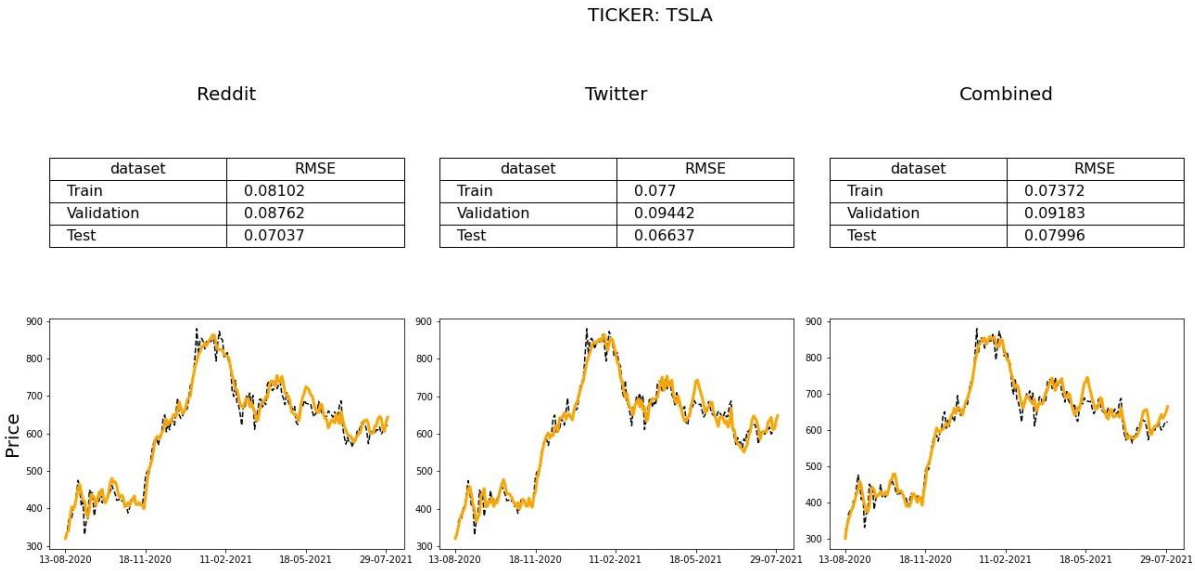


Figure 4.26: Prediction Plot For Price Prediction Of $TSLA Using Sentiments From BERT Model 2

4.5.3 Predictions From CNN-LSTM Using Sentiments From BERT Model 3

Figures 4.27 to 4.31 show the prediction of each stock by the CNN-LSTM model using sentiments predicted by BERT Model 3 from Reddit, Twitter, and a combination of both respectively. The dashed plot represents the original price and the solid, orange plot represents the predicted price.

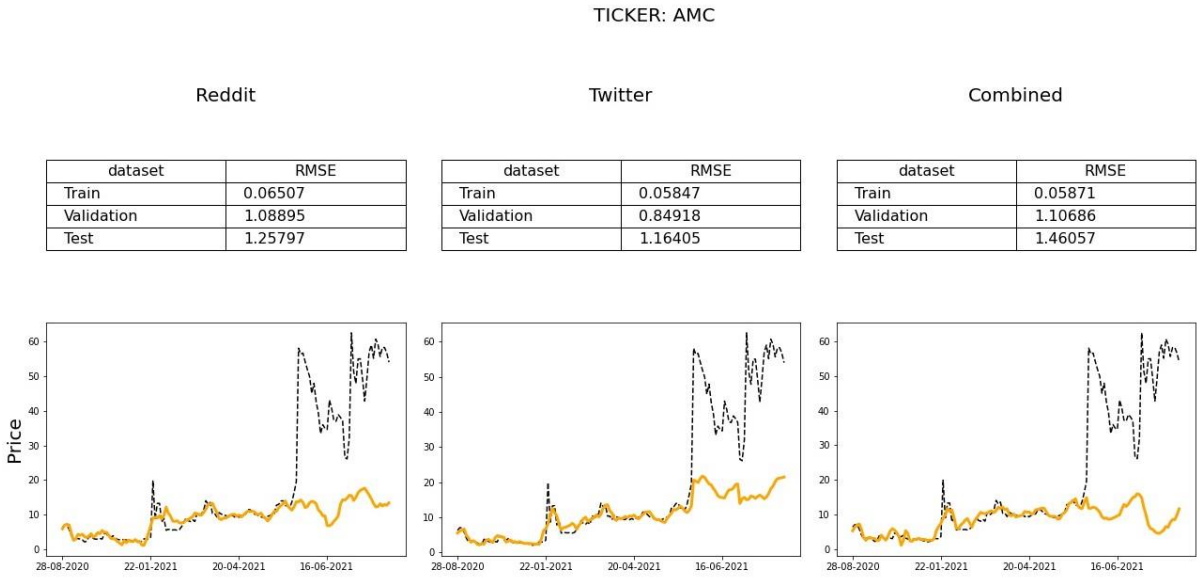


Figure 4.27: Prediction Plot For Price Prediction Of $AMC Using Sentiments From BERT Model 3

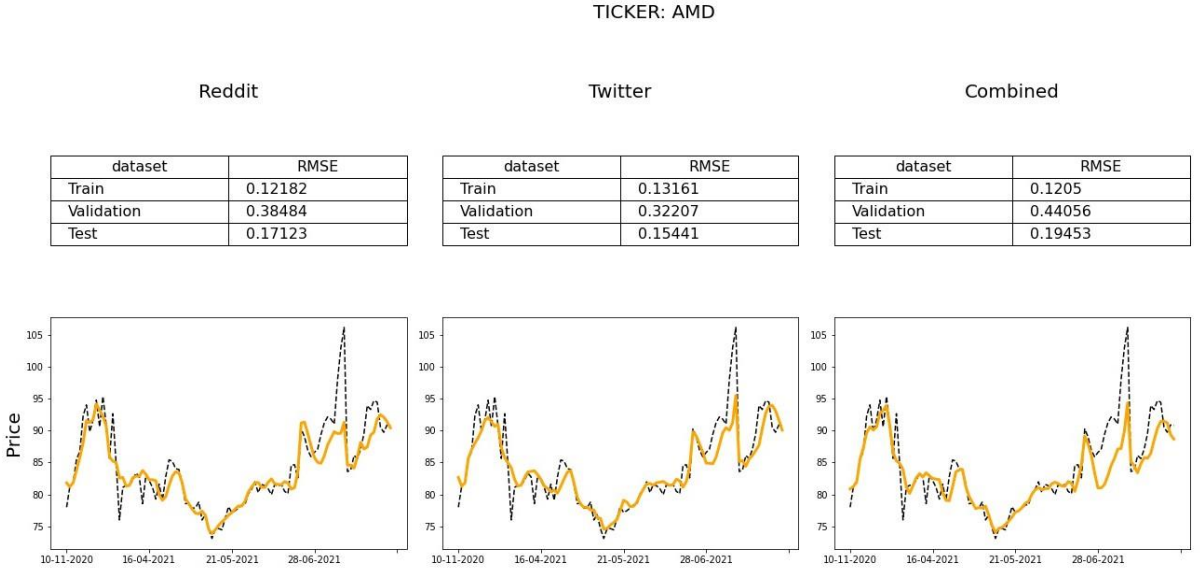


Figure 4.28: Prediction Plot For Price Prediction Of $AMD Using Sentiments From BERT Model 3

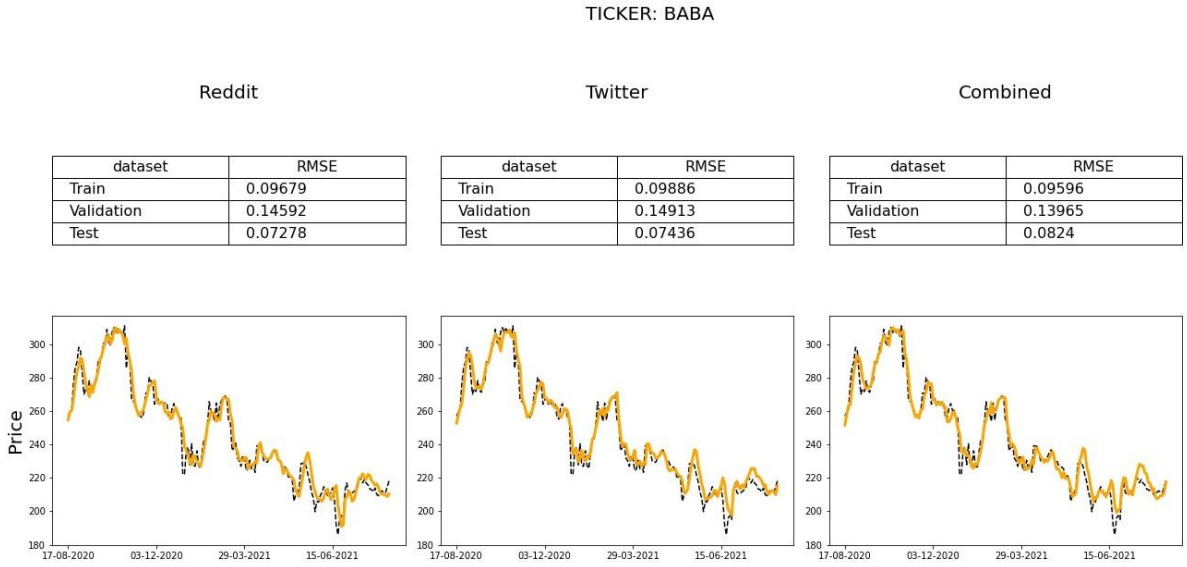


Figure 4.29: Prediction Plot For Price Prediction Of $BABA Using Sentiments From BERT Model 3

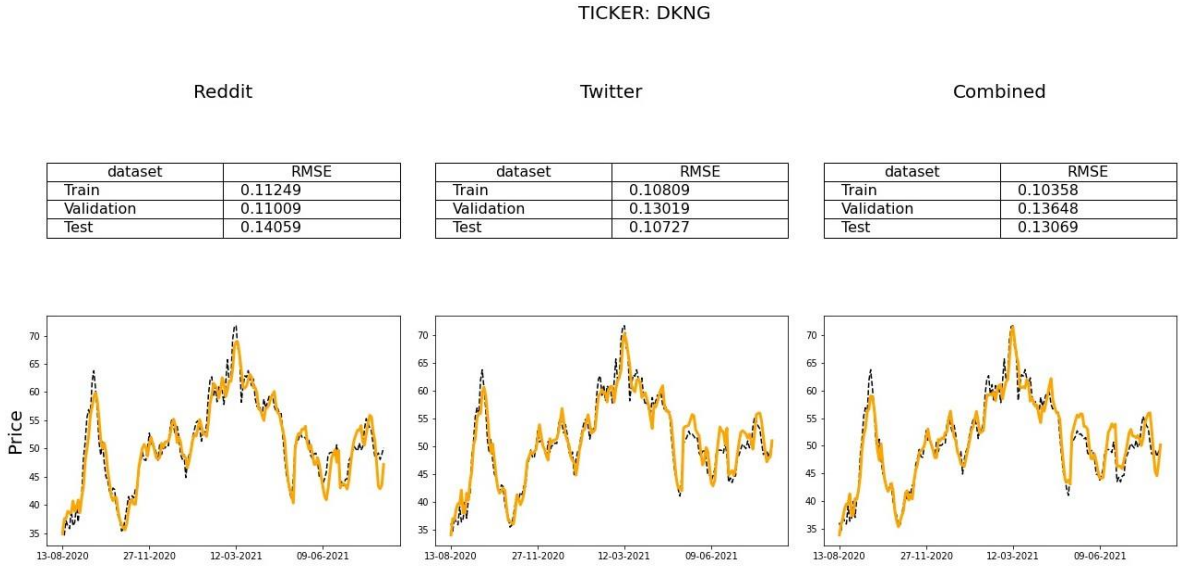


Figure 4.30: Prediction Plot For Price Prediction Of $DKNG Using Sentiments From BERT Model 3

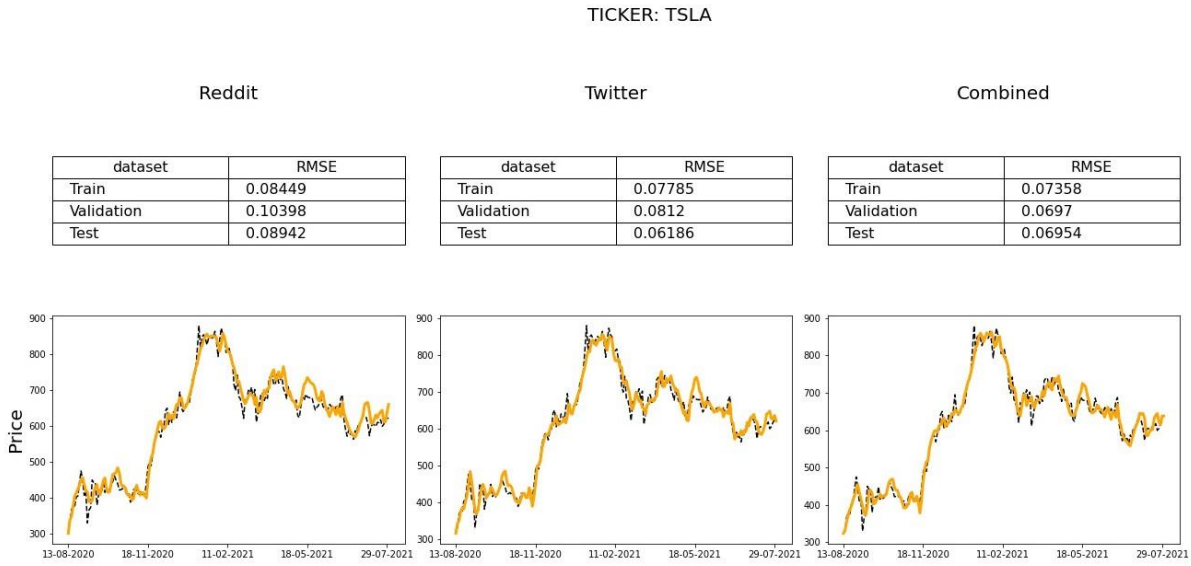


Figure 4.31: Prediction Plot For Price Prediction Of $TSLA Using Sentiments From BERT Model 3

4.5.4 Predictions From CNN-LSTM Using Sentiments From BERT Model 4

Figures 4.32 to 4.36 show the prediction of each stock by the CNN-LSTM model using sentiments predicted by BERT Model 4 from Reddit, Twitter, and a combination of both respectively. The dashed plot represents the original price and the solid, orange plot represents the predicted price.

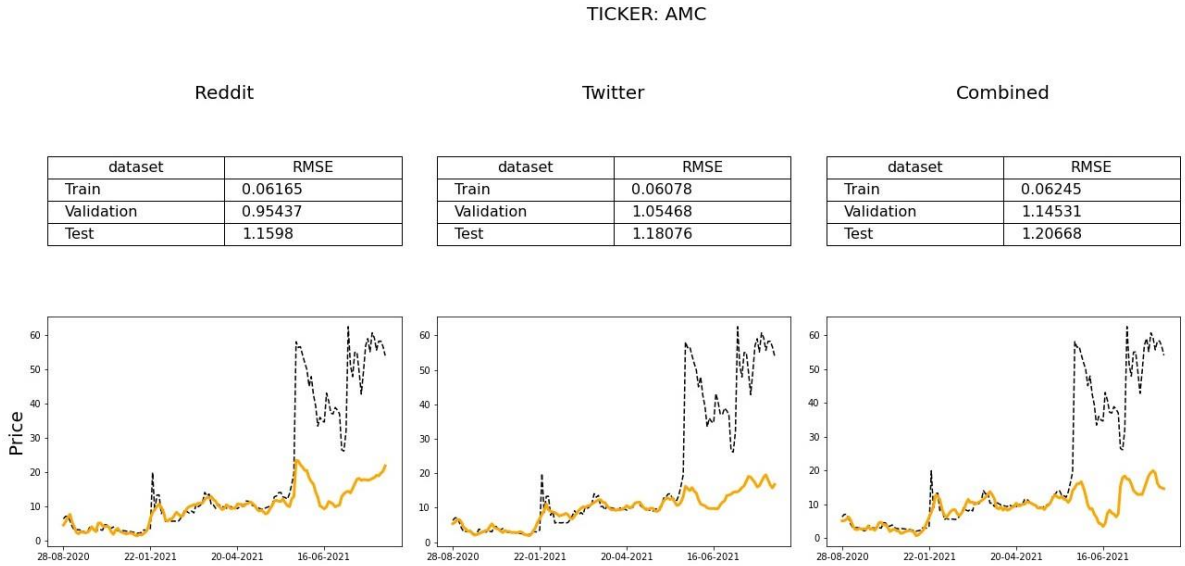


Figure 4.32: Prediction Plot For Price Prediction Of $AMC Using Sentiments From BERT Model 4

TICKER: AMD

Reddit

dataset	RMSE
Train	0.1376
Validation	0.41077
Test	0.15392

Twitter

dataset	RMSE
Train	0.1313
Validation	0.38526
Test	0.18567

Combined

dataset	RMSE
Train	0.13476
Validation	0.40779
Test	0.20351

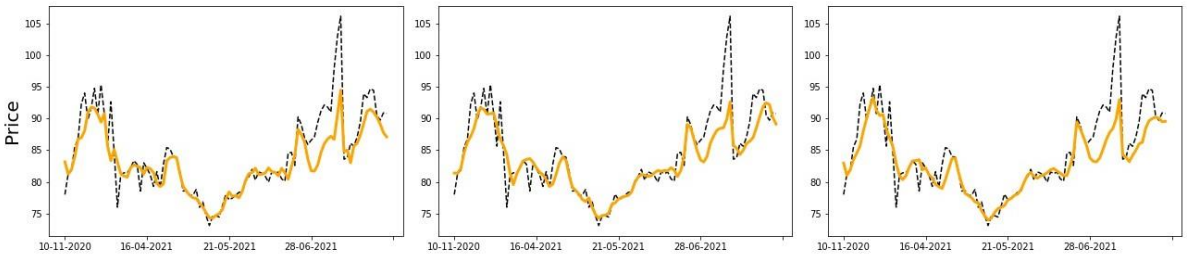


Figure 4.33: Prediction Plot For Price Prediction Of $AMD Using Sentiments From BERT Model 4

TICKER: BABA

Reddit

dataset	RMSE
Train	0.09928
Validation	0.13473
Test	0.05582

Twitter

dataset	RMSE
Train	0.09409
Validation	0.14121
Test	0.08141

Combined

dataset	RMSE
Train	0.0898
Validation	0.17513
Test	0.08551

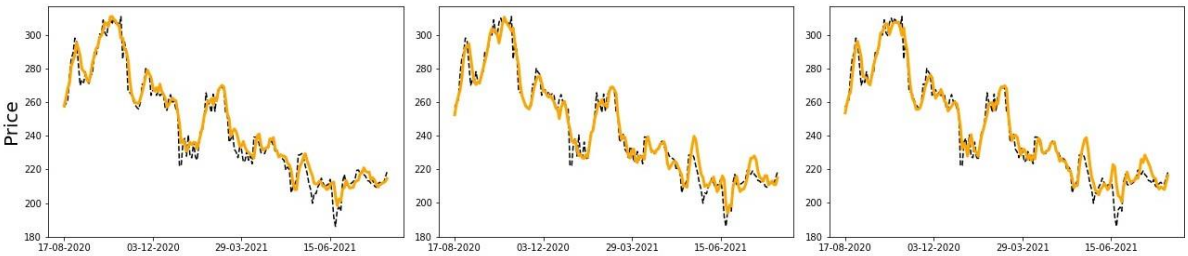


Figure 4.34: Prediction Plot For Price Prediction Of $BABA Using Sentiments From BERT Model 4

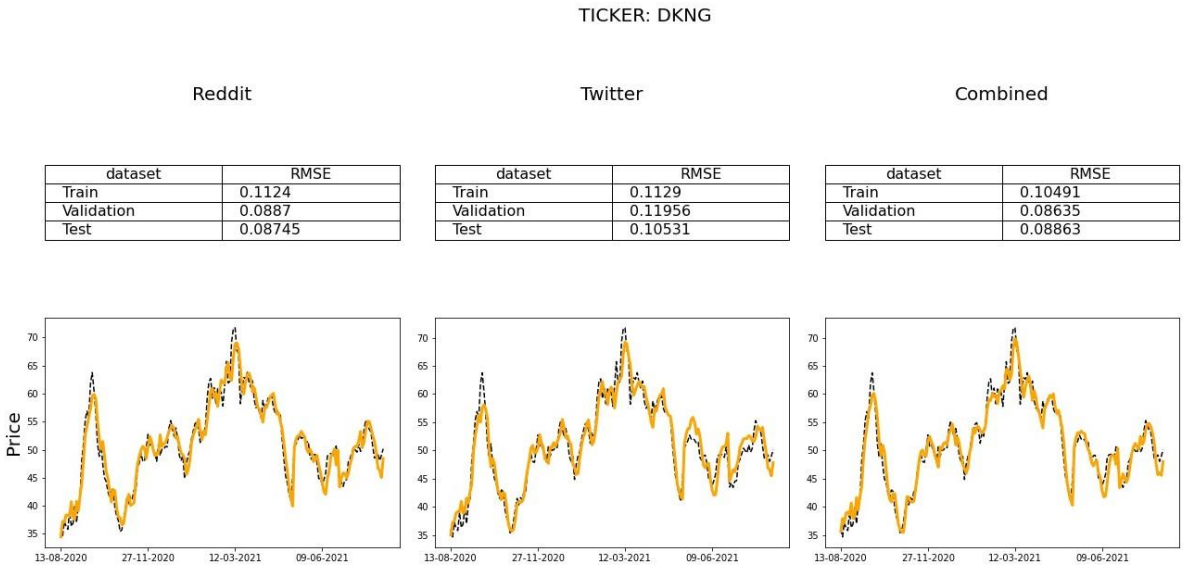


Figure 4.35: Prediction Plot For Price Prediction Of $DKNG Using Sentiments From BERT Model 4

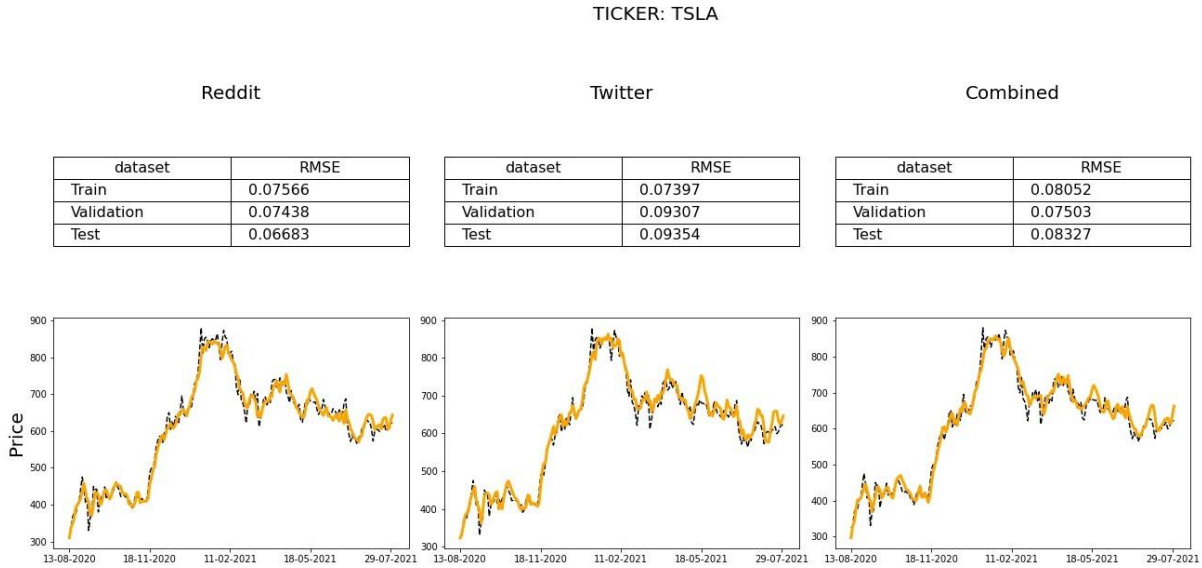


Figure 4.36: Prediction Plot For Price Prediction Of $TSLA Using Sentiments From BERT Model 4

4.6 Performance Comparisons

4.6.1 Metrics Comparison

Table 4.10 shows the average RMSE values for the prediction of all five stocks using sentiments from each BERT Model.

Table 4.10: Mean RMSE For Price Predictions Using Sentiments From Respective BERT Models

Model	Reddit RMSE	Twitter RMSE	Combined RMSE	Best RMSE
BERT Model 1	0.318337	0.33841	0.387554	Reddit
BERT Model 2	0.339763	0.347234	0.360553	Reddit
BERT Model 3	0.356577	0.309372	0.383098	Twitter
BERT Model 4	0.318677	0.344047	0.355721	Reddit

APPENDIX B: DETAILED VIEW OF RMSE VALUES provides a detailed view of the RMSE values where the values for each stock across each BERT model are shown.

4.6.2 Investment Scenario Examples

Tables 4.11 and 4.12 show the total profit and average profit with an initial investment of $100 in each of the five stocks under scenario 1 detailed in section 3.6.4 when stock prediction is done using sentiments from each BERT Model along with the respective technical inputs of each stock. Figures 4.37 to 4.40 show the cumulative profit as Scenario 1 is played out for each day in the prediction timeframe. To the side of each cumulative plot are bar plots representing the prediction error values (true price – predicted price) for Reddit, Twitter and combined datasets.

Table 4.11: Total Profit From Assumed Scenario 1

model	combined	Reddit	Twitter
BERT Model 1	1726.36	1557.48	1588.34
BERT Model 2	2000.7	1641.84	1539
BERT Model 3	1658.04	1540.44	1977.16
BERT Model 4	1661.58	2134.68	1757.42

Table 4.12: Average Profit From Assumed Scenario 1

model	combined	Reddit	Twitter
BERT Model 1	345.272	311.496	317.668
BERT Model 2	400.14	328.368	307.8
BERT Model 3	331.608	308.088	395.432
BERT Model 4	332.316	426.936	351.484

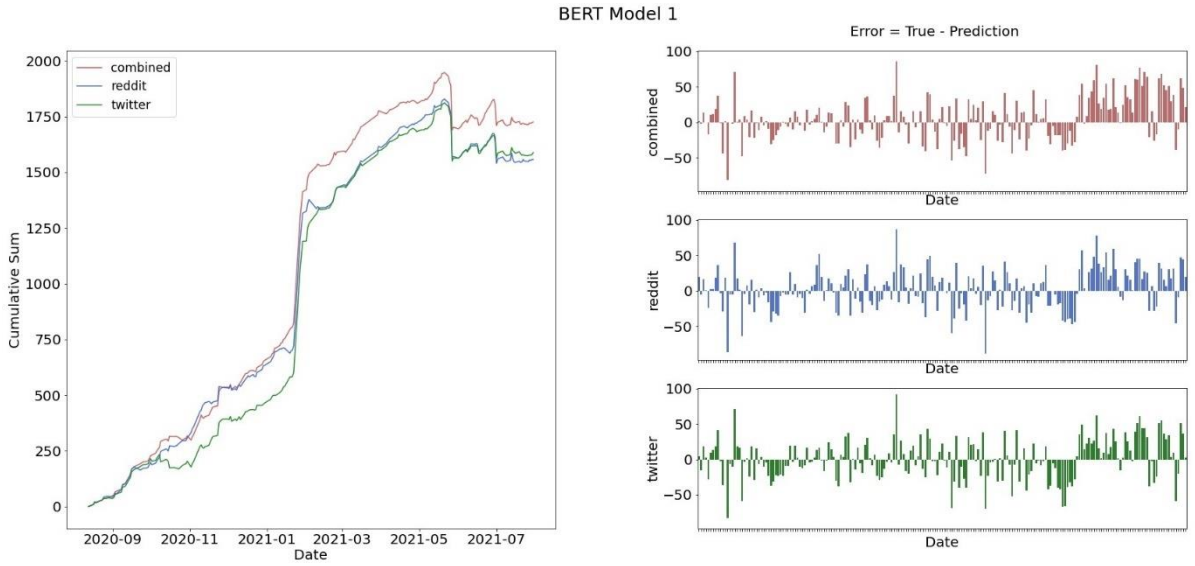


Figure 4.37: Cumulative Profit Under Scenario 1 Using Sentiments From BERT Model 1

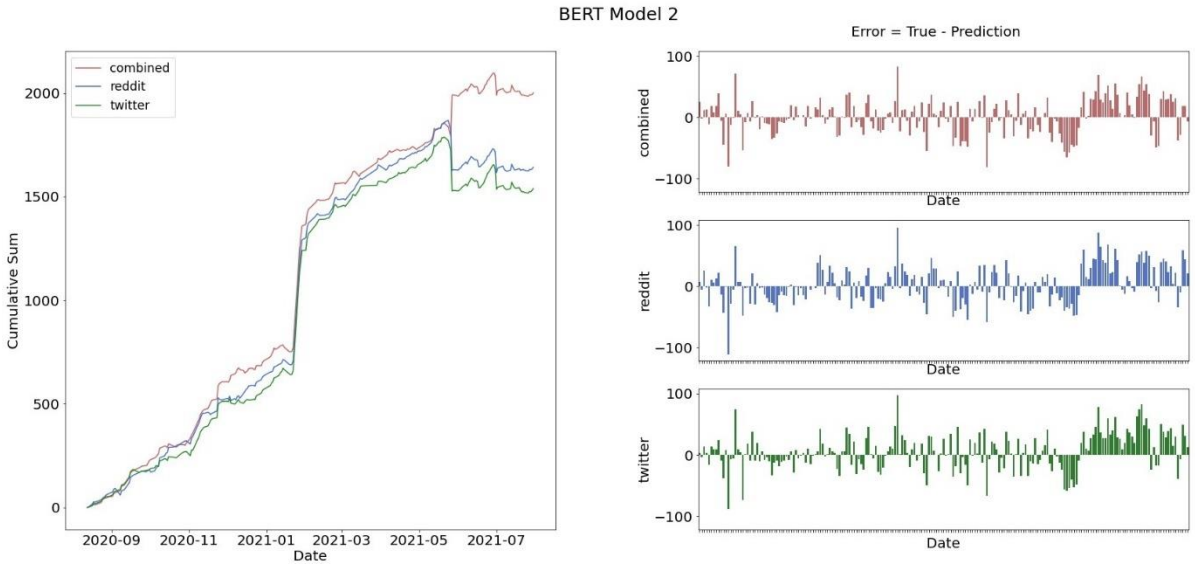


Figure 4.38: Cumulative Profit Under Scenario 1 Using Sentiments From BERT Model 2

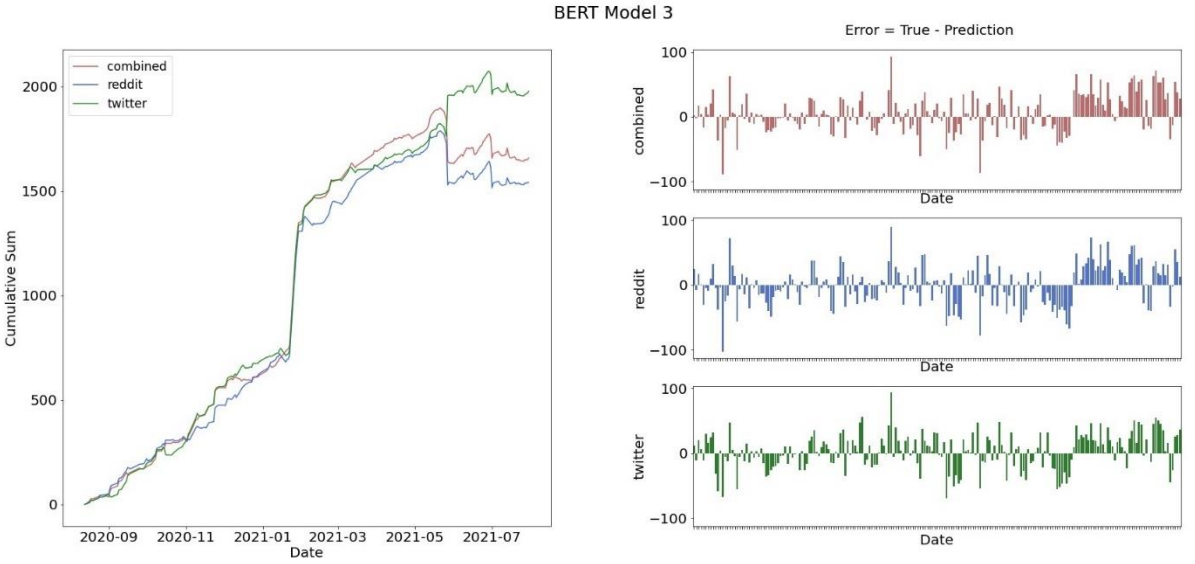


Figure 4.39: Cumulative Profit Under Scenario 1 Using Sentiments From BERT Model 3

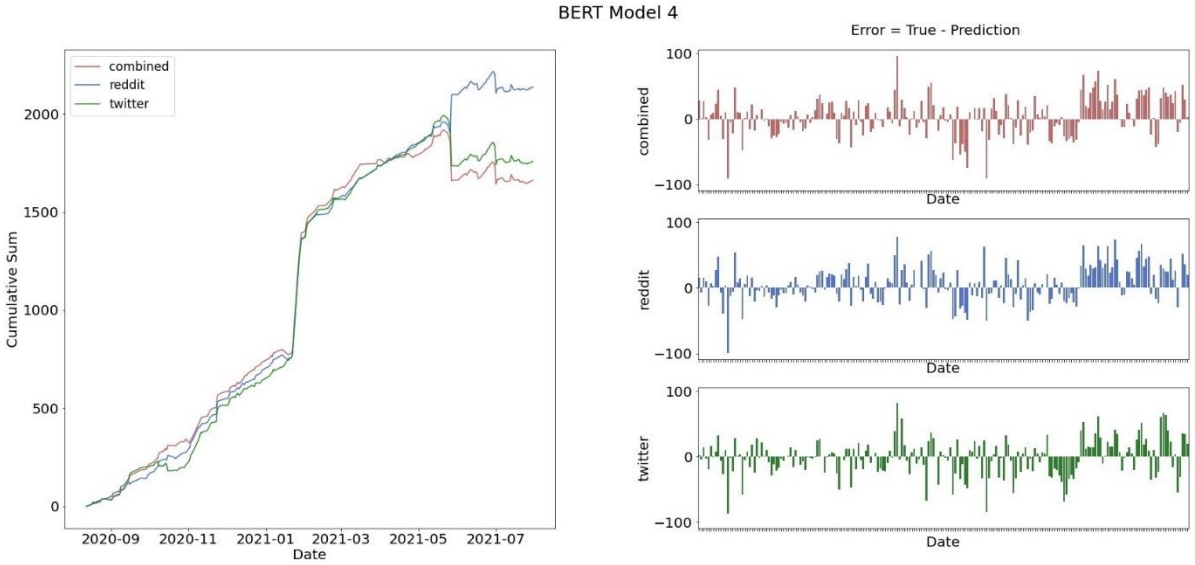


Figure 4.40: Cumulative Profit Under Scenario 1 Using Sentiments From BERT Model 4

Tables 4.13 and 4.14 show the total profit and average profit with an initial investment of $100 in each of the five stocks under scenario 2 detailed in section 3.6.4 when stock prediction is done using sentiments from each BERT Model along with the respective technical inputs of each stock. Figures 4.37 to 4.40 show the cumulative profit as Scenario 1 is played out for each day in the prediction timeframe.

Table 4.13: Total Profit From Assumed Scenario 2

model	combined	Reddit	Twitter
BERT Model 1	1333.62	1249.18	1264.61
BERT Model 2	1470.79	1291.36	1239.94
BERT Model 3	1299.46	1240.66	1459.02
BERT Model 4	1301.23	1537.78	1349.15

Table 4.14: Average Profit From Assumed Scenario 2

model	combined	Reddit	Twitter
BERT Model 1	266.724	249.836	252.922
BERT Model 2	294.158	258.272	247.988
BERT Model 3	259.892	248.132	291.804
BERT Model 4	260.246	307.556	269.83

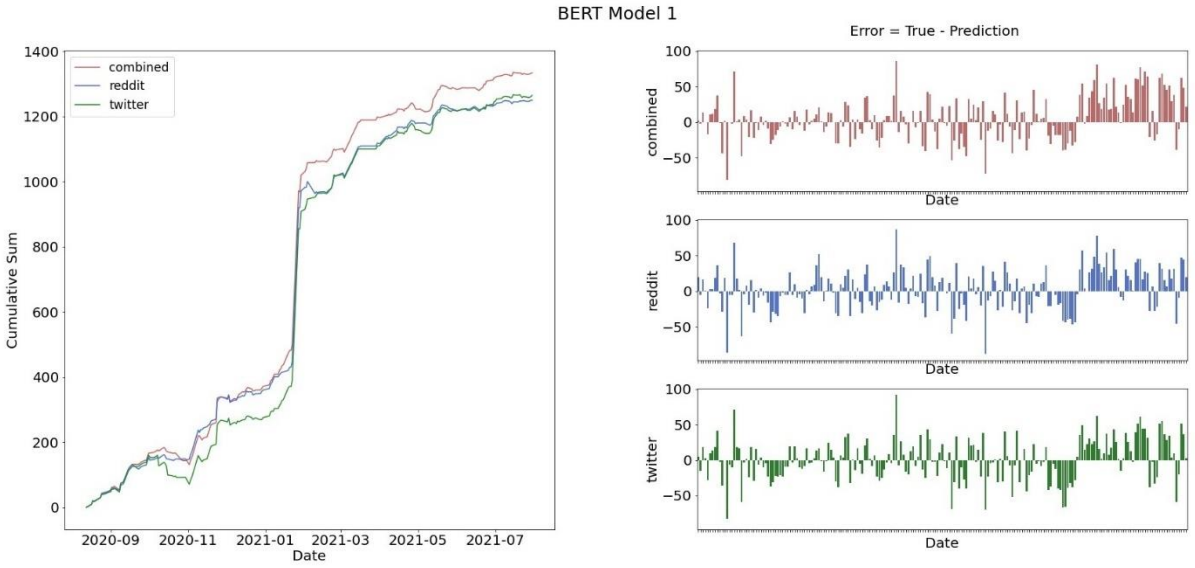


Figure 4.41: Cumulative Profit Under Scenario 2 Using Sentiments From BERT Model 1

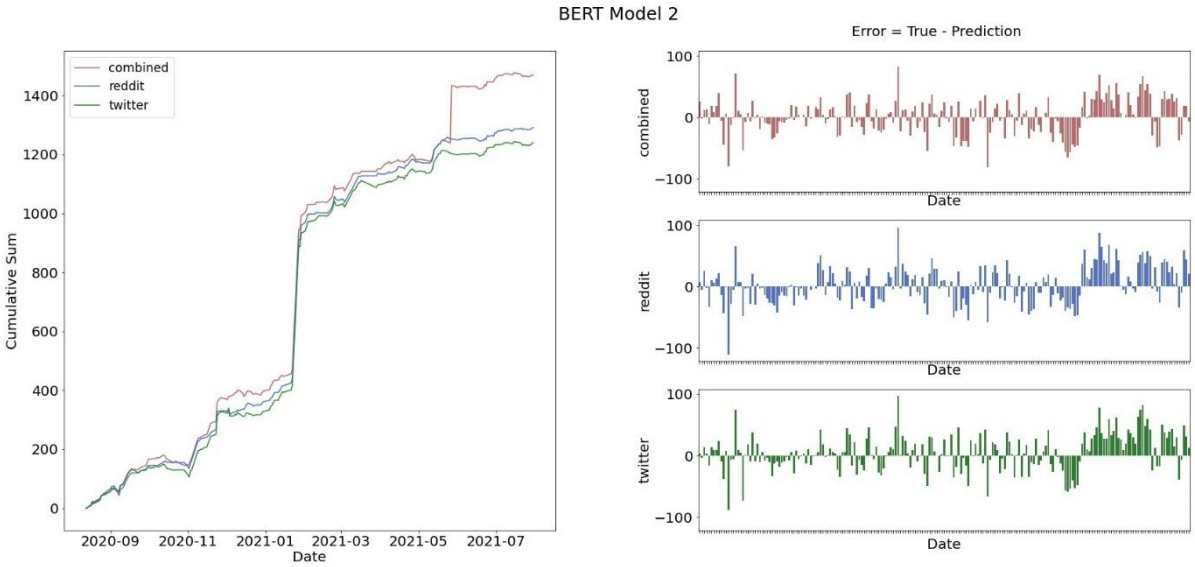


Figure 4.42: Cumulative Profit Under Scenario 2 Using Sentiments From BERT Model 2

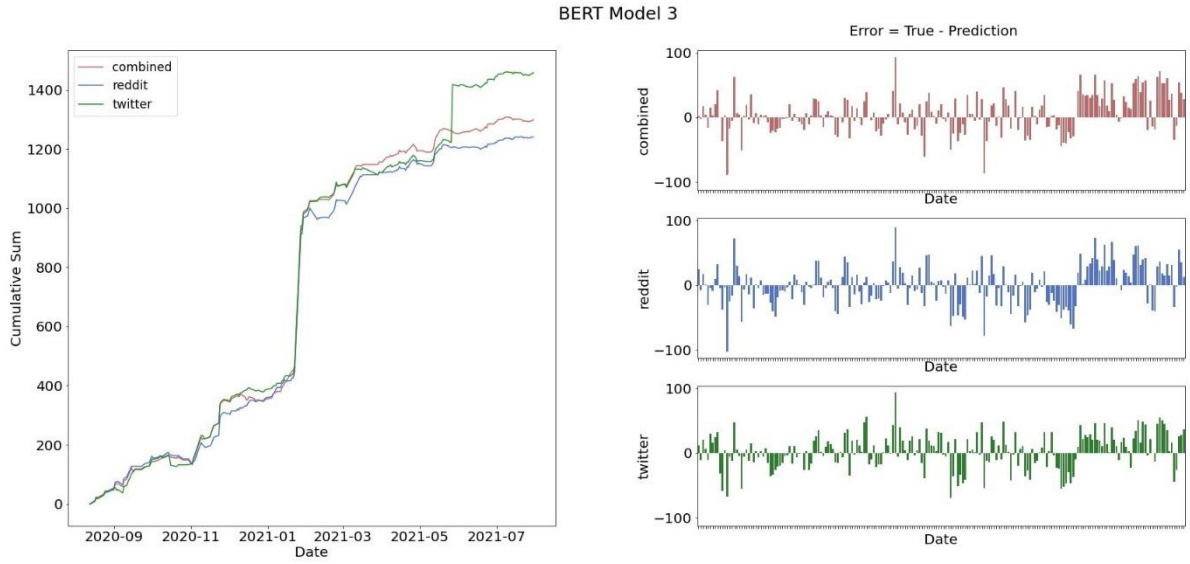


Figure 4.43: Cumulative Profit Under Scenario 2 Using Sentiments From BERT Model 3

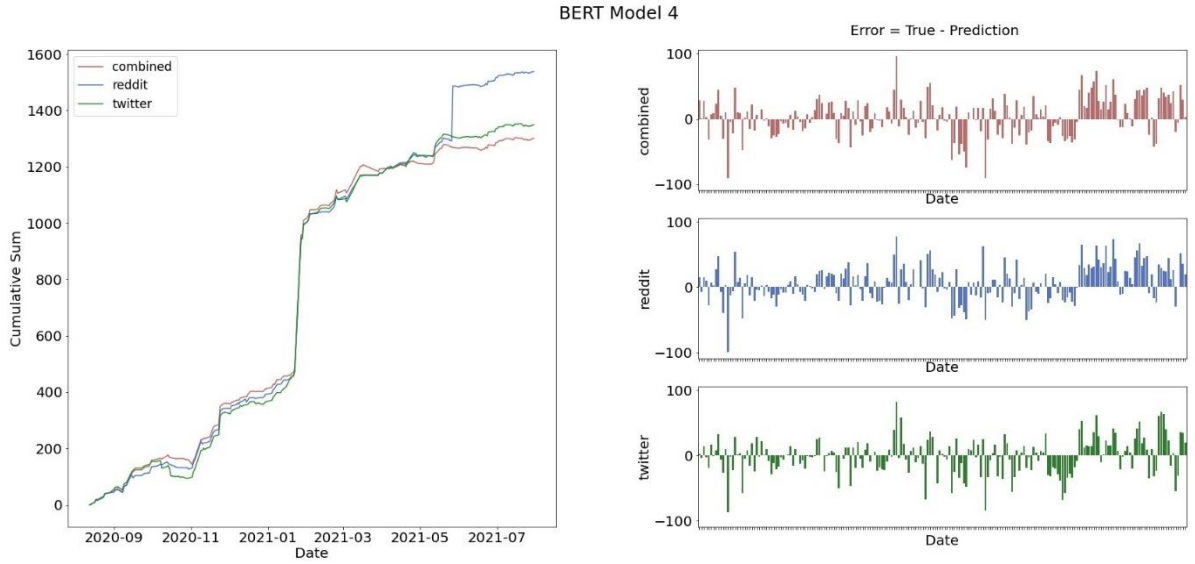


Figure 4.44: Cumulative Profit Under Scenario 2 Using Sentiments from BERT Model 4

5 DISCUSSION

5.1 Reddit And Twitter Datasets

Looking at the statistics of each stock in section 4.1, it was seen that even though the Reddit community where data was extracted from had over 11 million members, fewer people are talking about each stock on Reddit. One reason could be because this study used only using top-level comments on Reddit while ignoring the replies under it. However, looking at the sum_of_scores_per_day, which is the sum of likes of all posts or comments, Twitter seems to have more engagement compared to Reddit.

It was also seen that all the stocks had a good amount of data (over 12 days a month) each month apart from $AMC and $AMD. Reddit had a good amount of data for $AMD but very less for $AMC from August 2020 to March 2021. This was vice versa for $AMC where Twitter had a lot of data with Reddit having low data from August 2020 to March 2021. Due to the normalization of datasets to contain the same dates, both these stocks had the least amount of data.

5.2 Sentiment Outputs From BERT Models

From the confusion matrix in Figure 4.6, it could be seen that BERT Model 4 and BERT Model 1 have the best conformance between sentiments from the two datasets respectively when we consider the highest probable sentiment as the only sentiment. This does not mean that sentiments across the platforms are the same because, for stock price prediction, we consider

the probabilities of all three sentiments and not just the highest probable sentiment. This statement is substantiated through the histograms and correlations in Figures 4.7 to 4.16.

An interesting point to note from the histograms of probabilities is that Reddit data had a wider spread of probability values compared to Twitter; not just in one BERT Model but in all the models. The only exception was the sentiment probabilities for $AMD which had a wider spread for Twitter data in all models.

The correlation plots show that there is no correlation between the sentiment probabilities of Reddit data and Twitter data from all BERT models. The Pearson's correlation coefficient in each of the subplots in Figures 4.12 to 4.16 were seen to be very low.

From the confusion matrix, histograms and correlations, it was clear that each BERT Model behaved in a different way which stands to show that the fine-tuning dataset used for BERT Models have an influence on how the model predicts sentiment probabilities. This also shows that the use of multiple models was the right choice as performance and accuracy with differently tuned models could be compared.

5.3 Prediction Performance

From the prediction graphs shown in sections 4.5.1 to 4.5.4, it was seen that the CNN-LSTM model was able to provide decent predictions for all stocks apart from $AMC (the reason for which is explained in APPENDIX C: ANALYSIS OF POOR PREDICTION FOR $AMC). It was also seen that even-though training data used when training the model was shuffled and during prediction was not shuffled, the prediction accuracy is better for the data overlapping with the training data. Owing to this, the mean RMSE value of validation and testing datasets were used for performance comparison.

The mean RMSE value from the prediction of validation and testing datasets for all five stocks using sentiments inputs from each BERT Model were computed and are shown in Table 4.9. It was seen that the input datasets using sentiments from Reddit outperformed input datasets using sentiments from Twitter in three of the four BERT Models used. Twitter had a better prediction accuracy with BERT Model 3 which had only two sentiment outputs namely positive and negative. Though this is an area for investigation in future studies, one possible explanation can be inferred from the confusion matrix and histogram values in Figures 4.6 to 4.11. Both BERT Models 1 and 4 showed that a lot of comments had neutral sentiment as the highest

probable outcome. So, this might explain why the lack of neutral sentiment output in BERT Model 3 gives advantage to Twitter data.

5.4 Directional Accuracy

Tables 4.11 and 4.12 show the total and average profits when using Scenario 1. Tables 4.13 and 4.14 show the total and average profits when using scenario 2. None of the datasets produced a loss in any scenario which signifies that all datasets are good with directional accuracy.

In both scenarios,

- considering only Reddit and Twitter data, Reddit performed better with sentiments from BERT Models 2 and 4 while Twitter performed better with BERT Models 1 and 3. Even with RMSE values, Twitter performed better with BERT Model 3
- Combined data from Reddit and Twitter had the best overall directional accuracy with BERT Models 1 and 2 while Twitter and Reddit data in isolation had the best directional accuracy with BERT Models 3 and 4 respectively

A strong positive point for Reddit is that its data performed the best with BERT Model 4 which was the finBERT model trained with financial corpus data. This model provided equal footing for both Reddit and Twitter data since the other BERT models were finetuned with Twitter data and some performance bias towards Twitter is expected in those models. Though, this alone could not be used as a confirmation that Reddit data is particularly better than Twitter data as a source of investor sentiments, this adds more weightage to the positive conclusions for Reddit with regards to research questions Q2 and Q3.

6 CONCLUSION

The study was successful in answering the research questions that it set out at the outset. The research questions Q2 and Q3 can be answered unequivocally. It was seen from the various outputs and prediction metrics that sentiments from Reddit are indeed a credible source of investor sentiments data. Also, both assumed scenarios produced profits with Reddit sentiment data –in isolation and in combination with Twitter data, showing that a prediction model using sentiments from Reddit has the directional accuracy required for turning profits.

Even though Reddit data had better prediction accuracy with three of the four BERT models and a better directional accuracy with two of the BERT models (including finBERT), it can only be concluded that Reddit has the potential to be more effective than Twitter data with regards to investor sentiments. The performance of Reddit is highly commendable considering it had less data compared to Twitter (average number of authors and comments per day).

However, to unequivocally prove this, we hope future studies with more resources will look to overcome the limitations of this study mentioned in the section 1.3 by,

- using more time and resources to extract data from the official APIs of Reddit and Twitter so that no data is missed
- using high-resource machines to pre-train BERT models from scratch with huge financial corpus datasets
- having access to experts on sentiment analysis who can create labelled Reddit and Twitter datasets related to investment news and posts and training BERT models for Reddit and Twitter with the respective labelled datasets

TABLE OF ABBREVIATIONS

ANN	Artificial Neural Network
API	Application Process Interface
BERT	Bidirectional Encoder Representations from Transformers
BERTBASE	Base version of BERT
BERTLARGE	Large version of BERT
CNN	Convolutional Neural Network
DNN	Deep Neural Network
EGARCH	Exponential Generalized Autoregressive Conditional Heteroskedasticity
ELMo	Embeddings from Language Model
FinBERT	A BERT model pre-trained with financial data
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GPT	Generative Pre-trained Transformer
LSTM	Long Short-Term Memory (Type of RNN)
MLP	Multilayer Perceptron
NLP	Natural Language Processing
NN-EGARCH	Hybrid model of [Neural Network + EGARCH]
NN-GARCH	Hybrid model of [Neural Network + GARCH]
NSE	National Stock Exchange
PLM	Pre-Training Models
PMI	Partial mutual information
ReLu	Rectified Linear Unit function
SOTA	State of the Art
Subreddit	Generic name for communities within Reddit
ULMFit	Universal Language **Model** Fine-tuning

BIBLIOGRAPHY

Acosta, J., Lamaute, N., Luo, M., Finkelstein, E. and Andreea, C., 2017. Sentiment analysis of twitter messages using word2vec. *Proceedings of Student-Faculty Research Day, CSIS, Pace University*, *7*, pp.1-7.

Alzazah, F. S. & Cheng, X., 2020. Recent Advances in Stock Market Prediction Using Text Mining: A Survey. *E-Business-Higher Education and Intelligence Applications.*

Antyukhov, D., 2019. *Pre-training BERT from scratch with cloud TPU.* [Online]. Available at: https://towardsdatascience.com/pre-training-bert-from-scratch-with-cloud-tpu-6e2f71028379 [Accessed 23 November 2021].

Araci, D., 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063.*

Fatima, M. and Mueller, M.C., 2019, August. HITS-SBD at the FinSBD Task: Machine Learning vs. Rule-based Sentence Boundary Detection. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing* (pp. 115-121).

Baumgartner, J. M. & Seiler, A., n.d. *Pushshift Reddit API Documentation.* [Online]. Available at: https://github.com/pushshift/api [Accessed 15 October 2021].

Broadstock, D. C. & Zhang, D., 2019. Social-media and intraday stock returns: The pricing power of sentiment. *Finance Research Letters,* Volume **30**, pp. 116-123.

Campbell, J., 2001. Internet finance forums: Investor empowerment through CMC or market manipulation on a global scale?.

Campbell, J.A., 2001, January. In and out, scream and shout: an Internet conversation about stock price manipulation. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences* (pp. 10-pp). IEEE.

Carstens, R. & Freybote, J., 2019. Tone in REIT financial statements and institutional investments. *Journal of Property Research,* **36**(3), pp. 227-244.

Carvajal, J. A. T., 2021. Social media Effects on the market: Reddit Data analysis on Stocks.

Chandar, S. K., 2021. Grey Wolf optimization-Elman neural network model for stock price prediction. *Soft Computing,* **25**(1), pp. 649-658.

Cheung, B., 2021. *Before WallStreetBets: A history of online message boards and 'stonks'*. [Online]. Available at: https://uk.style.yahoo.com/before-wall-street-bets-a-history-of-online-message-boards-and-stonks-134818361.html [Accessed October 2021].

Colby, R.W., 2003. *The encyclopedia of technical market indicators*. McGraw-Hill, pp. 8-9.

Devlin, J., 2018. *bert.* [Online]. Available at: https://github.com/google-research/bert [Accessed 22 November 2021].

Devlin, J. & Chang, M.-W., 2018. *Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing.* [Online]. Available at: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html [Accessed 02 November 2021].

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Du, J., Huang, Y. and Moilanen, K., 2019, August. AIG Investments. AI at the FinSBD task: Sentence boundary detection through sequence labelling and BERT fine-tuning. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing* (pp. 81-87).

Feldman, R., 2013. Techniques and applications for sentiment analysis. *Communications of the ACM*, Volume **56**.

Frew, C., 2021. *Reddit Traders Celebrate As Hedge Fund That Shorted GameStop Forced To Shut Down.* [Online]. Available at: https://www.unilad.co.uk/news/reddit-traders-celebrate-as-hedge-fund-that-shorted-gamestop-forced-to-shut-down/ [Accessed 24 06 2021].

Gao, T. & Chai, Y., 2018. Improving stock closing price prediction using recurrent neural network and technical indicators. *Neural computation*, **30**(10), pp. 2833-2854.

Ghosh, S., 2021. *Reddit group WallStreetBets hits 6 million users overnight after a wild week of trading antics.* [Online]. Available at: https://www.businessinsider.com/wallstreetbets-fastest-growing-subreddit-hits-58-million-users-2021-1?r=US&IR=T [Accessed 10 October 2021].

Go, A., Bhayani, R. & Huang, L., n.d. *Sentiment140.* [Online]. Available at: http://help.sentiment140.com/for-students [Accessed 22 November 2021].

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y., 2016. *Deep learning.* 1 ed. Cambridge: MIT press Cambridge, pp. 330-345.

Guo, X. and Li, J., 2019, October. A novel twitter sentiment analysis model with baseline correlation for financial market prediction with improved efficiency. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 472-477). IEEE.

Hajizadeh, E., Seifi, A., Zarandi, M. F. & Turksen, I., 2012. A hybrid modeling approach for forecasting the volatility of S&P 500 index return. *Expert Systems with Applications*, **39**(1), pp. 431-436.

Hamner, B., 2019. *Twitter US Airline Sentiment.* [Online]. Available at: https://www.kaggle.com/crowdflower/twitter-airline-sentiment/discussion/18283 [Accessed 20 November 2021].

Hartwig, J., 2021. *Reddit's Top Investing and Trading Communities.* [Online]. Available at: https://www.investopedia.com/reddit-top-investing-and-trading-communities-5189322 [Accessed 20 November 2021].

Hiew, J.Z.G., Huang, X., Mou, H., Li, D., Wu, Q. and Xu, Y., 2019. BERT-based financial sentiment index and LSTM-based stock return predictability. *arXiv preprint arXiv:1906.09024*.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, ***9***(8), pp.1735-1780.

Hoseinzade, E. & Haratizadeh, S., 2018. CNNPred: CNN-based stock market prediction using several data sources. *arXiv preprint arXiv:1810.08923.*

Hu, B., Lu, Z., Li, H. and Chen, Q., 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems* (pp. 2042-2050).

Hu, Z., Zhao, Y. & Khushi, M., 2021. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, **4**(1), p. 9.

HY Chiu, I., 2021. Social Disruptions in Securities Markets-What Regulatory Response Do We Need?. *Richmond Journal of Law & Technology*, *28*(1).

Jung, S.H. and Jeong, Y.J., 2021. Examining stock markets and societal mood using Internet memes. *Journal of Behavioral and Experimental Finance*, ***32***.

Kaggle, 2019. *Twitter US Airline Sentiment.* [Online]. Available at: https://www.kaggle.com/crowdflower/twitter-airline-sentiment?select=Tweets.csv [Accessed 23 November 2021].

Leitch, G. & Tanner, J. E., 1991. Economic forecast evaluation: profits versus the conventional error measures. *The American Economic Review,* pp. 580-590.

Li, M., Li, W., Wang, F., Jia, X. and Rui, G., 2021. Applying BERT to analyze investor sentiment in stock market. *Neural Computing and Applications*, ***33***(10), pp.4663-4676.

Liu, Z., Huang, D., Huang, K., Li, Z. and Zhao, J., 2021, January. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence* (pp. 4513-4519).

Livieris, I. E., Pintelas, E. & Pintelas, P., 2020. A CNN-LSTM model for gold price time-series forecasting. *Neural computing and applications,* **32**(23), pp. 17351-17360.

Li, X., Xie, H., Chen, L., Wang, J. and Deng, X., 2014. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, ***69***, pp.14-23.

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*, p. 31.

Lubitz, M., 2017. Who drives the market? Sentiment analysis of financial news posted on Reddit and Financial Times. *University of Freiburg: http://ad-publications. informatik. uni-freiburg. de/theses/Bachelor_ Michael_Lubitz_2018. pdf.*

Lu, W., Li, J., Li, Y., Sun, A. and Wang, J., 2020. A CNN-LSTM-based model to forecast stock prices. *Complexity*, *2020*.

Lu, W., Li, J., Wang, J. & Qin, L., 2020. A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications,* pp. 1-13.

Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M. and Balahur, A., 2018, April. Www'18 open challenge: financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018* (pp. 1941-1942).

Malo, P., Sinha, A., Korhonen, P., Wallenius, J. and Takala, P., 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, ***65***(4), pp.782-796.

Munikar, M., Shakya, S. and Shrestha, A., 2019, November. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)* (Vol. **1**, pp. 1-5). IEEE.

McGurk, Z., Nowak, A. & Hall, J. C., 2020. Stock returns and investor sentiment: textual analysis and social media. *Journal of Economics and Finance,* Volume **44**, pp. 458--485.

Mehtab, S. and Sen, J., 2020, November. Stock price prediction using CNN and LSTM-based deep learning models. In *2020 International Conference on Decision Aid Sciences and Application (DASA)* (pp. 447-453). IEEE.

Moghaddam, A. H., Moghaddam, M. H. & Esfandyari, M., 2016. Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science,* **21**(41), pp. 89-93.

Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P. and Anastasiu, D.C., 2019, April. Stock price prediction using news sentiment analysis. In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)* (pp. 205-208). IEEE.

Naik, N. & Mohan, B. R., 2020. Intraday Stock Prediction Based on Deep Neural Network. *National Academy Science Letters,* **43**(3), pp. 241-246.

Nasdaq, n.d. *Stock Screener.* [Online]. Available at: https://www.nasdaq.com/market-activity/stocks/screener [Accessed 30 October 2021].

Nguyen, T. H., Shirai, K. & Velcin, J., 2015. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications,* **42**(24), pp. 9603-9611.

Ou, P. & Wang, H., 2009. Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science,* Volume 3, pp. 28-42.

Platen, P. v., n.d. *Bert-base-uncased.* [Online]. Available at: https://huggingface.co/bert-base-uncased [Accessed 22 September 2021].

ProsusAI, n.d. *Finbert.* [Online]. Available at: https://huggingface.co/ProsusAI/finbert [Accessed 01 November 2021].

Pytorch, n.d.a *Datasets & Dataloaders.* [Online]. Available at: https://pytorch.org/tutorials/beginner/basics/data_tutorial.html [Accessed 01 November 2021].

Pytorch, n.d.b *Torch.* [Online]. Available at: https://pytorch.org/tutorials/beginner/blitz/tensor_tutorial.html#sphx-glr-beginner-blitz-tensor-tutorial-py [Accessed 10 october 2021].

Qi, L., Khushi, M. and Poon, J., 2020, December. Event-driven LSTM for forex price prediction. In *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)* (pp. 1-6). IEEE.

Rafail, P. & Freitas, I., 2020. Natural Language Processing. *SAGE Research Methods Foundations.*

Roh, T. H., 2007. Forecasting the volatility of stock price index. *Expert Systems with Applications,* **33**(4), pp. 916-922.

Saunders, M. N., Lewis, P., Thornhill, A. & Bristow, A., 2015. *Understanding research philosophy and approaches to theory development.* Harlow: Pearson Education.

Sebastian, W. & Isa, S. M., 2020. Stock Price Prediction Using BERT and Word2Vec Sentiment Analysis. *International Journal of Emerging Trends in Engineering Research.*

Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., 2017, September. Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 1643-1647). IEEE.

Sezer, O.B., Gudelek, M.U. and Ozbayoglu, A.M., 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, ***90***, p.106181.

Shynkevich, Y., McGinnity, T.M., Coleman, S.A., Belatreche, A. and Li, Y., 2017. Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, ***264***, pp.71-88.

Sirois, A., 2021. *The Top 10 Meme Stocks on Reddit: Should You Buy, Sell or Hold?.* [Online]. Available at: https://finance.yahoo.com/news/top-10-meme-stocks-reddit-180846931.html?guccounter=1 [Accessed 20 November 2021].

Smith, I. & Wigglesworth, R., 2021. *GameStop's wild ride: how Reddit traders sparked a 'short squeeze'.* [Online]. Available at: https://www.ft.com/content/47e3eaad-e087-4250-97fd-e428bac4b5e9 [Accessed 05 06 2021].

Sundermeyer, M., Schlüter, R. and Ney, H., 2012. LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

ThinkMarkets, n.d. *Using the Bill Williams Accelerator Oscillator.* [Online] Available at: https://www.thinkmarkets.com/en/indicators/bill-williams-accelerator/ [Accessed 01 November 2021].

Thomsett, M. C., 2017. *Candlestick Charting : Profiting from Effective Stock Chart Analysis.* Boston: Walter de Gruyter GmbH, pp. 175-176.

Tian, K. and Peng, Z.J., 2019, August. aiai at finsbd task: Sentence boundary detection in noisy texts from financial documents using deep attention model. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing* (pp. 88-92).

Usmani, S. and Shamsi, J.A., 2021. News sensitive stock market prediction: literature review and suggestions. *PeerJ Computer Science*, ***7***, p.e490.

Uszkoreit, J., 2017. *Transformer: A Novel Neural Network Architecture for Language Understanding.* [Online]. Available at: https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html [Accessed 30 October 2021].

Utthammajai, K. and Leesutthipornchai, P., 2015. Association mining on stock index indicators. *International Journal of Computer and Communication Engineering*, ***4***(1), p.46.

Valencia, F., Gómez-Espinosa, A. and Valdés-Aguirre, B., 2019. Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. *Entropy*, ***21***(6), p.589.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Wu, J.L., Yu, L.C. and Chang, P.C., 2014. An intelligent stock trading system using comprehensive features. *Applied Soft Computing*, ***23***, pp.39-50.

Yong, B.X., Rahim, M.R.A. and Abdullah, A.S., 2017, August. A stock market trading system using deep neural network. In *Asian Simulation Conference* (pp. 356-364). Springer, Singapore.

Yu, P. and Yan, X., 2020. Stock price prediction based on deep neural networks. *Neural Computing and Applications*, ***32***(6), pp.1609-1628.

Zhang, L., Wang, F., Xu, B., Chi, W., Wang, Q. and Sun, T., 2018. Prediction of stock prices based on LM-BP neural network and the estimation of overfitting point by RDCI. *Neural Computing and Applications*, ***30***(5), pp.1425-1444.

APPENDIX A: STRUCTURE OF CNN AND LSTM NETWORKS

A.1 CNN

CNN or Convolutional Neural Networks were initially used for image processing problems but have since then evolved for other applications. A CNN network can be comprised of one or more convolutional layer.

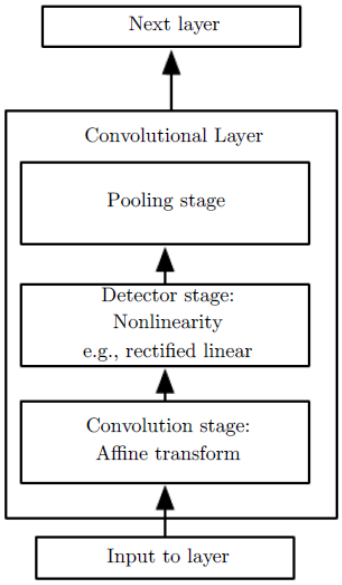


Figure A.1: Structure Of A Convolutional Layer

Each convolutional layer has three parts to it (Goodfellow et al 2016) namely,

1. Convolution stage

   The convolution stage is a linear transformation for feature extraction. Convolution is performed on the input dataset using a kernel (matrix). Figure A.2 outlines the convolution operation. The kernel is passed over the input data and produces the output via convolution. The output is referred to as the feature map. The below image shows how a kernel is passed over an input to produce feature maps.

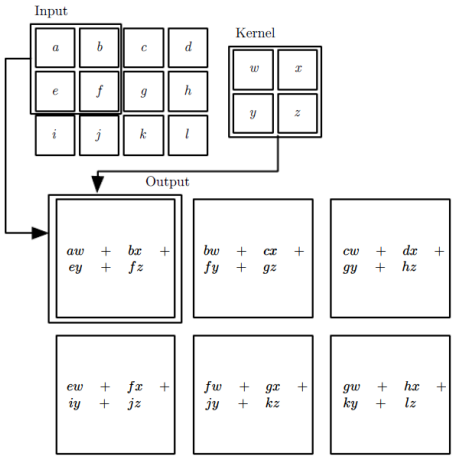


Figure A.2:Simple Example Of Convolution Operation

If we have a 2D input I and a kernel K, then the feature maps can be denoted as,

$S(I,j) = (K * I)(i,j) = \Sigma_m \Sigma_n I(m,n)K(i-m, j-n)$

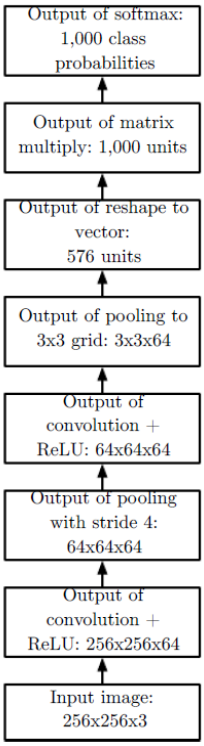


Figure A.3: Sample Representation Of Dimensionality Reduction In A CNN

2.Detector stage

The second stage introduces non-linearity in the network using a function such as ReLu. Each linear activation from the previous stage is run through a non-linear activation in this stage.

3.Pooling stage

The output from the detector stage is passed through a pooling function (such a max pooling). This reduces the dimensionality of the feature maps. The pooling layer mostly uses a square grid (say, 2x2, 3x3, etc) and the output depends on the function used. For example, if maxpooling function is used, then the output would be the highest value within the grid, or if it is average, then it would be the average of all values within a grid. Stride represents the number of steps to move the grid on the detector stage output.

By stacking several convolution layers, the end output is only the high-level features which are easy to work with. Figure A.3 on the left shows how the entire CNN network reduces the dimensionality of an input. A 256 x 256 x 3 input is reduced to 64 x 64 x 64 after the first convolutional layer and then to 3 x 3 x 64 by the end of second convolutional layer.

A.2 LSTM

LSTM or Long Short-Term Memory networks were first introduced by (Hochreiter & Schmidhuber 1997). They help overcome some shortcomings of a regular RNN (Recurrent Neural Network). LSTM network overcomes the gradient vanish and gradient exploding problems in a regular RNN. They do this by giving additional weights to long-term interactions compared to a regular RNN where long-term interactions are given exponentially low weights (Goodfellow et al 2016).

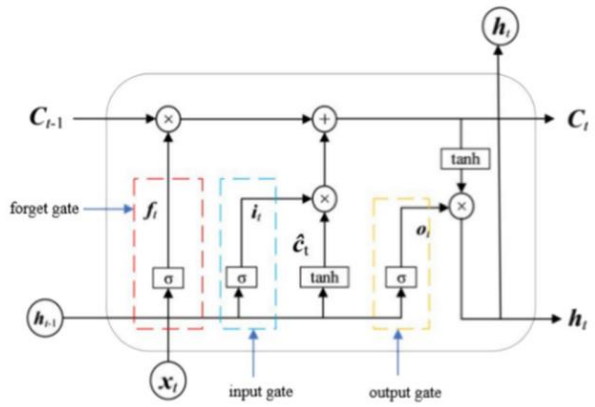


Figure A.4: Structure Of A LSTM Memory Cell With Input, Output And Forget Gates

LSTM allows for long-term interactions in inputs by using three gates in its cell namely input, output, and forget gates. The gates have a sigmoid function which forces the output to a value between 0 to 1. Each of the gates also have their own weights and biases. Apart from this, outside of the gates, there are tanh functions which force output between -1 to 1.

The above diagram shows the architecture of a single cell in a LSTM network (Lu et al 2020) where,

- C_{t-1}, C_t represent cell state of previous input and current cell state respectively
- x_t represents current input
- h_{t-1}, h_t represents output value for the previous input and output value of current input respectively
- f_t, i_t, and o_t represent the outputs from forget, input and output gates respectively
- ⓧ represents multiplication operation

- $\oplus$ represents addition operation

- σ represents sigmoid function

The below section presents equations for the LSTM architecture presented above. w and b denote the weights and biases at each gate of the architecture.

The forget gate uses the current input and output from previous input to decide how much of previous cell state to retain. The value of f_t is between 0 and 1. 0 means completely forget/ discard previous cell state while 1 means keep the previous cell state as is.

$$f_t = \sigma\,(w_f\,[h_{t-1},x_t] + b_f)$$

Next, the input gate decides what additional information is to be added to the previous cell state. i_t denotes which values to update while $\hat{c}_t$ denotes a vector of new candidate values.

$$i_t = \sigma\,(w_i\,[h_{t-1},x_t] + b_i)$$

$$\hat{c}_t = tanh(w_c[h_{t-1},x_t] + b_c)$$

Both these are multiplied, and the resulting information is added to the processed previous cell state after its interaction with the forget gate output f_t. The complete equation for the current cell state is given by,

$$C_t = (f_t * c_{t-1}) + (i_t * \hat{c}_t)$$

In the final part, the output gate with a sigmoid function decides the part of cell state to pass on as the output. The current cell state C_t is passed through a tanh function and then multiplies with the output of output gate, O_t to form the current output h_t.

$$o_t = \sigma\,(w_o\,[h_{t-1},x_t] + b_o)$$

$$h_t = o_t * tanh(c_t)$$

APPENDIX B: DETAILED VIEW OF RMSE VALUES

Table B.1: Detailed View Of RMSE Values

Ticker	Model	Reddit RMSE	Twitter RMSE	Combined RMSE	Best Result
AMC	**BERT Model 1**	1.082775	1.141935	1.403975	reddit
	BERT Model 2	1.156365	1.119235	1.09659	combined
	BERT Model 3	1.17346	1.006615	1.283715	twitter
	BERT Model 4	1.057085	1.11772	1.175995	reddit
AMD	**BERT Model 1**	0.22687	0.244915	0.26993	reddit
	BERT Model 2	0.27441	0.297645	0.331655	reddit
	BERT Model 3	0.278035	0.23824	0.317545	twitter
	BERT Model 4	0.282345	0.285465	0.30565	reddit
BABA	**BERT Model 1**	0.11004	0.10062	0.090785	combined
	BERT Model 2	0.109665	0.139625	0.173765	reddit
	BERT Model 3	0.10935	0.111745	0.111025	reddit
	BERT Model 4	0.095275	0.11131	0.13032	reddit
DKNG	**BERT Model 1**	0.09296	0.11455	0.092665	combined
	BERT Model 2	0.07938	0.09927	0.11486	reddit
	BERT Model 3	0.12534	0.11873	0.133585	twitter
	BERT Model 4	0.088075	0.112435	0.08749	combined
TSLA	**BERT Model 1**	0.07904	0.09003	0.080415	reddit
	BERT Model 2	0.078995	0.080395	0.085895	reddit
	BERT Model 3	0.0967	0.07153	0.06962	combined
	BERT Model 4	0.070605	0.093305	0.07915	reddit

APPENDIX C: ANALYSIS OF POOR PREDICTION FOR $AMC

Figure C.1 shows the plot of $AMC prices. It could be seen that there is a sharp spike from about $15 to around $60.

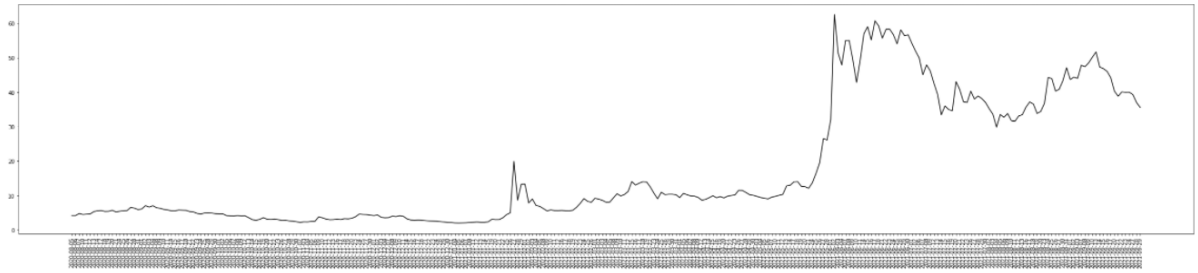


Figure C.1: Plot Of $AMC Prices

Figure C.2 shows the plots of training, validation and testing dataset with the split that was used in the study. It could be seen that the training data stops immediately after the spike and the validation and testing data are in price ranges which the model would not have learned with the training data.

Figure C.2: Datasets Used For Training, Validation And Testing Respectively

Figure C.3 shows the bad prediction performance with the above input datasets.

BERT model1

dataset	MAE	RMSE	MAPE	MSE	R2
Train	0.04755	0.14725	28.73069	0.02168	0.68635
Validation	0.65957	0.66634	627.62708	0.44401	-11.12447
Test	0.82661	0.84431	301.86722	0.71286	-7.08557

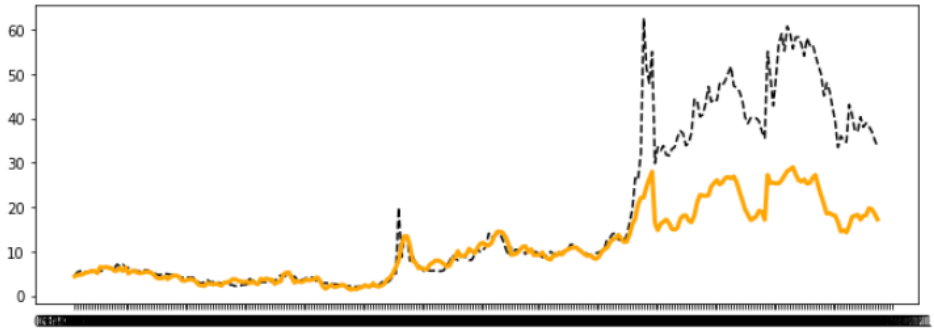


Figure C.3: Predictions With No Changes To Training Data

To confirm if the above analysis was the cause of poor prediction performance, the training dataset was updated to have some data before and after the spike as shown in Figure C.4.

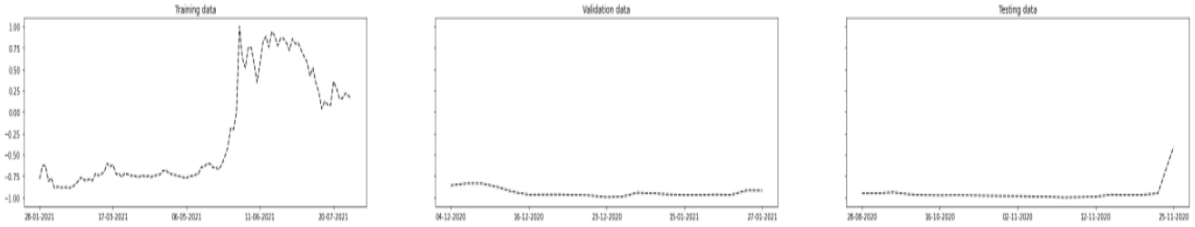


Figure C.4: Training, Validation And Testing Data After Updating The Timelines For Training Data

When trained and predicted with the updated training dataset, the prediction performance improved irrespective of Reddit (Figure C.5) and Twitter (Figure C.6) sentiment. This proves that the poor performance if due to the nature of the increase in stock prices which due to the timeline and data split were not learned by the CNN-Model. Since we are working with a time series prediction model, it is not ideal to have training dataset in the middle of a time-

series. But even though $AMC has a poor prediction performance; the directional accuracy of prediction could be measured using the two assumed scenarios.

BERT model1

dataset	MAE	RMSE	MAPE	MSE	R2
Train	0.08265	0.15731	45.03326	0.02475	0.93919
Validation	0.16848	0.18534	17.97851	0.03435	-14.23641
Test	0.20713	0.23763	25.49358	0.05647	-2.85419

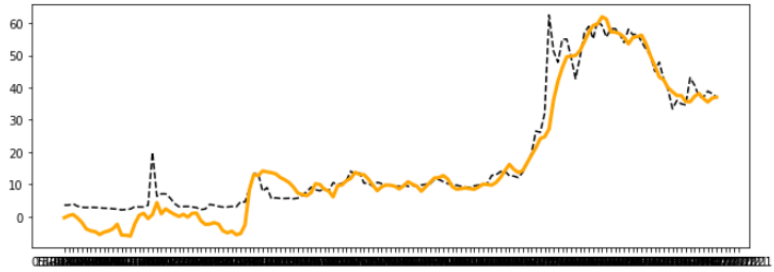


Figure C.5: Prediction Plot Using Updated Training Data With Reddit Sentiments

BERT model1

dataset	MAE	RMSE	MAPE	MSE	R2
Train	0.06939	0.12459	19.06471	0.01552	0.96185
Validation	0.19017	0.19739	20.33113	0.03896	-16.28276
Test	0.16656	0.20449	21.31116	0.04181	-1.85399

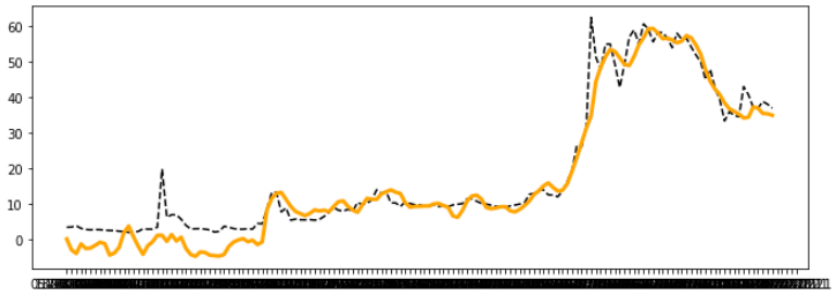


Figure C.6: Prediction Plot With Updated Training Data Using Twitter Sentiments

APPENDIX D: LINKS TO PROJECT FILES ON ONEDRIVE

All the code and intermediary outputs from the Python codes used to perform the study can be found in the link,

https://outlookuwicac-my.sharepoint.com/:f:/g/personal/st20183898_outlook_cardiffmet_ac_uk/EixJ0CKguNhOr4L-M8YF--oBiCb1hxIBOJo7rA0OT76eQw?e=NDHwze

The above link is the main folder which has all the data. The below mentioned specific links can be used to look at specific data,

- Jupyter Notebook Files: These are the python code on which the entire study was implemented.

  https://outlookuwicac-my.sharepoint.com/:f:/g/personal/st20183898_outlook_cardiffmet_ac_uk/EsDTdVqv_BZNqLMEBWvJUB8BFVabebqdu5p62v9ARstRaA?e=eoEhlM

  The below table D.1 lists all the python code files and gives a brief description of what the code in each file does.

Table D.1: Python Code And Description

Jupyter Notebook File	**Description**
F1A_Extracting_data_from_reddit.ipynb	Code used to extract Reddit data from Pushshift api
F1B_Extracting_data_from_twitter.ipynb	code used to extract Twitter data using snscrape library
F2A_Prepare_reddit_data_for_BERT.ipynb	Data-pre-processing activities on Reddit data
F2B_Prepare_twitter_data_for_BERT.ipynb	Data-pre-processing activities on Twitter data

F3A_Bert_sentiment_analysis_TwiiterAirline_Even.ipynb	Fine-tuning, hyper-parameter tuning and sentiment predictions using BERT Model 1
F3B_Bert_sentiment_analysis_TwiiterAirline_Uneven.ipynb	Fine-tuning, hyper-parameter tuning and sentiment predictions using BERT Model 2
F3C_Bert_sentiment_analysis_Sentiment140.ipynb	Fine-tuning, hyper-parameter tuning and sentiment predictions using BERT Model 3
F3D_finBERT_sentiment_analysis.ipynb	sentiment predictions using BERT Model 4 (finBERT)
F4A_Stock_technical_data.ipynb	code to retrieve price data from yfinance for all stocks and to calculate the 15 technical indicators
F5A_Hyperparameter tuning for each stock.ipynb	Hyper-parameter tuning of CNN-LSTM models
F6A_CNN_LSTM_Predictions_for_BERT1_data.ipynb	Stock price prediction using sentiments from BERT Model 1
F6B_CNN_LSTM_Predictions_for_BERT2_data.ipynb	Stock price prediction using sentiments from BERT Model 2
F6C_CNN_LSTM_Predictions_for_BERT3_data.ipynb	Stock price prediction using sentiments from BERT Model 3
F6D_CNN_LSTM_Predictions_for_BERT4_data.ipynb	Stock price prediction using sentiments from BERT Model 4
F7A_metrics_comparison.ipynb	Computation of average RMSE values
F7B_Dataset_Statistics.ipynb	Confusion Matrix, histograms and correlation plots for Reddit and Twitter sentiments
F7C_Dataset_Statistics_2.ipynb	Descriptive statistics of all datasets
F7D_Dataset_Counts_Per_Month.ipynb	Plots to show the normalization of dates of Reddit and Twitter data
F7E_Profit_Loss_Calculations.ipynb	Calculation of profits in Scenario1 and Scenario 2

Z8A_Analysis of poor prediction performance of AMC 1.ipynb	Analysis on the poor prediction performance of $AMC stock
Z8B_Analysis of poor prediction performance of AMC 2.ipynb	
Z8C_Analysis of poor prediction performance of AMC 3.ipynb	

- Saved BERT Models: The best performing BERT Models after the hyper-parameter tuning process for each BERT model was saved here

  https://outlookuwicac-my.sharepoint.com/:f:/g/personal/st20183898_outlook_cardiffmet_ac_uk/Engu9BrFB3FIgbHAVknCTIMBbPKEwOhermI6rdfa2hpkRw?e=hUt45A

- Final Dfs: Intermediary results are stored in the below folder
  https://outlookuwicac-my.sharepoint.com/:f:/g/personal/st20183898_outlook_cardiffmet_ac_uk/EmTztXW0T4hJkjb5zKCc0BYBsrN_7OGEC7YXhK8gpgTRMA?e=BhYBBD

- Plots and Metrics: All plots and subplots along with tabular data which were saved from python code are stored in the below folder,

  https://outlookuwicac-my.sharepoint.com/:f:/g/personal/st20183898_outlook_cardiffmet_ac_uk/Er6lqooAHmxMu4XXsZ8QfpABmT4F-UtsxZyQBsnemMbSJg?e=ymlHUf

- Extracted data from Reddit and Twitter are stored in the below locations,

  Reddit:
  https://outlookuwicac-my.sharepoint.com/:f:/g/personal/st20183898_outlook_cardiffmet_ac_uk/EpSHuGtnUHFFpEL8d-IkUSABC5MnGZEDWkoM0WZTXXZwmw?e=wc73wJ

Twitter:

https://outlookuwicac-my.sharepoint.com/:f:/g/personal/st20183898_outlook_cardiffmet_ac_uk/Et_Znl8C56JGrNf1gRm-RukBJyLwQWKkQYlHTHoL7-XzgQ?e=OPdPab