University of Glasgow | School of Computing Science

The Influence of News and Tweets on Stock Prices: A Comparative Study

Hari Sudarsan

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

31 August 2023

Abstract

Stock market prediction is an important yet challenging task in quantitative finance. While previous works have used historical prices, emerging research applies natural language processing (NLP) to extract signals from textual data like news and social media. This thesis develops models for stock movement prediction using a combination of tweets, news articles, and historical prices. Multiple deep learning architectures are explored including BERT, RoBERTa, and GRUs. Contextualized word embeddings allow sentiment analysis going beyond positive/negative classification. A comparative study is conducted between price-only baselines and hybrid models incorporating text. The results demonstrate the value of NLP-based sentiment analysis in improving stock prediction. However, the contribution of news versus social media is shown to differ across sectors. This work provides insights into the relationships between public discourse and quantitative market movements based on rigorous experimentation with transformer and recurrent neural networks. The hybrid models outperform price-only baselines, validating the importance of unstructured data in financial forecasting.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: *Hari Sudarsan* Signature: *Hari Sudarsan*

Acknowledgements

I would like to express my sincere gratitude to Dr. Simon Gay for his invaluable support throughout this project. His guidance, mentorship, and feedback were essential to the completion of this thesis. I am also grateful to my family and friends for their constant love and support during this time. Their encouragement and understanding helped me to stay motivated and focused on my work.

Contents

Chapter 1: Introduction

1.1 Motivation

Stock market prediction is a critical and challenging field of study in the financial markets. Previous studies have shown that stock prices can be predicted based on historical price data.[6] However, the stock price series are dynamic, noisy, and non-linear. This makes it difficult to predict future stock prices with high accuracy using historical price data alone.

The development of natural language processing (NLP) has revolutionized the field of stock market prediction. Investors can now analyze other factors, such as political events, economic conditions, and environmental factors, from news and social media.[6] The development of sentiment analysis has further improved the accuracy of stock price prediction. Extracting sentiments from text using transformer-based models like BERT has become increasingly important in the financial markets. The integration of contextualized word embedding and deep learning has made this field a major success.[2]

Despite the advances in NLP-based stock market prediction, there are still a number of challenges that need to be addressed. One challenge is that the impact of different factors on stock prices can vary depending on the domain of the stock and the type of article. For example, an article about a new product launch by a technology company may have a different impact on the stock price of a technology company than an article about a political scandal. Another challenge is that the contribution of news and tweets to the performance of NLP-based stock price prediction models is not fully understood. Some studies have shown that news and tweets can improve the accuracy of stock price prediction models, while other studies have shown that the contribution of news and tweets is limited.

1.2 Purpose

The overall aim of this thesis is to develop a model to analyze the stock movement using tweets, news, and historic stock price. This is a comparative study among different constraints. First, a hybrid model will be developed to predict stock movement using sentiment analysis and historic prices. The primary aim of this thesis is to evaluate and compare the efficiency of different models. This work will focus on stocks from three different domains(Technology,Health care,Semi-conductors), which will allow us to analyze the influence of news and tweets on different sectors. Two separate models will also be developed, one with news data and the other with tweets. This will help to analyze the influence of stocks and news on stock prices.

1.3 Outline

This project is divided into five chapters.

- Chapter 1: Introduces the motivation and purpose of the project, provides an overview of the stock market, and discusses the challenges of stock prediction.
- Chapter 2: Reviews the previous studies on stock prediction and discusses the different approaches to stock prediction.

- Chapter 3: Describes the data processing pipeline, introduces the different models used for stock prediction using contextualized word embedding and discusses the evaluation metrics used to evaluate the models.
- Chapter 4: evaluates the different models on the test data and analyzes the results of the evaluation.
- Chapter 5: summarizes the findings of the project, compares the different models, identifies the best-performing model, and discusses the scope and future work of the project.

Chapter 2: Background and Analysis

2.1 Stock prediction

Stock price prediction is inevitable in the field of financial markets. There have been many technologies used for this purpose since the beginning of the 1980s. Historic stock market details were used to predict the stock market at that time. Stock market prediction is useful for investors and financial institutions to reduce risk and increase profit.

In the early stages of stock market forecasting, fundamental statistics focused on analyzing time series data of prices and volumes. Researchers applied basic statistical techniques [14] like random walks, Bayesian inference, and regression modeling to find meaningful signals in the numeric data. The random walk theory views stock prices as following a random trajectory, with daily changes being independent of each other. Early forecasting methods modeled price changes as random variables drawn from an underlying distribution. Statistical tests were used to check for autocorrelation patterns that could refute the random walk hypothesis. Bayesian methods were also applied, which rely on prior probability distributions that are updated as new data becomes available [1]. Bayesian models aimed to estimate the likelihood of future price movements based on the evolving probability landscape. Regression models also gained popularity for predicting stock returns [17]. Linear regression fits a line to historical data to estimate future returns, while nonlinear regression can model more complex relationships between variables. These fundamental statistical techniques marked an important early stage of stock forecasting research. Manual analysis was time-consuming due to the mathematical complexity. But it laid the groundwork for more advanced data-driven modeling approaches as computing power increased.

Later, the development of data-driven technologies enhanced the stock price forecasting process by reducing the computational burden through direct weight optimization. The integration of machine learning techniques and historic stock price data has helped to improve the results in the field of stock market prediction due to the non-linear nature of the market.

In the early days, deep learning models like multilayer perceptrons and convolutional neural networks (CNNs) were applied for stock prediction. Deep learning refers to neural networks with multiple layers capable of learning hierarchical feature representations. However, these basic architectures lacked mechanisms for sequential modeling. widely used deep learning models for this problem are decision trees, random forests, adaptive boosting (AdaBoost),and XGBoost[13]. These were the commonly used algorithms for stock market forecasting. Long short-term memory (LSTM) was the most efficient algorithm among them. Recurrent neural networks (RNNs) were a breakthrough in modeling temporal data. RNNs contain cyclical connections that maintain an internal state over time steps. This allows RNNs to remember patterns across potentially long sequences, mimicking short-term memory. But the major problem with RNN was the chance for the loss of the data by different events. LSTM networks overcame this limitation by introducing a gated architecture. LSTMs contain memory cells and gates that learn what information to remember or forget over many time steps. The gating mechanism was key to preserving long-range dependencies critical in financial forecasting[7]. At that time, researchers were also focused on forecasting the effect of repurchase using ANN. They developed a better model using cascade-forward

backpropagation artificial neural networks[10]. This model used rule-based data clustering for data evaluation.

The development of RNNs paved a concrete path for researchers to develop models. Along with this, they started developing more models using RNN technologies and attention models. The development of models like LSTM, ARIMA,[3] and GARCH were milestones in stock price forecasting. . Statistical methods like ARIMA were also hybridized with neural networks to complement the strengths of both models. The ability to remember historical patterns and context makes RNNs uniquely suited for financial time series forecasting.

After 2000, there was a better foundation for forecasting stock prices using machine learning, and research continued. However, there were many different factors that affected the prediction due to the high volatility risk of the predicted values,[17] uncertain trends due to the impact of different external factors such as environmental, political, and so on, market noise, limited information, and other issues that affected stock market prediction using historical stock prices. The development of natural language processing (NLP) helped to integrate sentiment analysis from different social media and news sources with historic data and to develop a hybrid attention model. This made a lot of changes in the field of stock price prediction.

2.2 Contextualized word embedding

The impact of natural language processing (NLP) in the field of stock market prediction has been significant. Semantic vector models have made many developments for stock forecasting, leading to more efficient models than previous ones.

In the early days, NLP was used by tokenizing unique words and assigning each word a single dimension for computation.This was computationally complex. So an advancement was the development of distributed word representations called word embeddings. In this approach, words are represented as dense vectors capturing semantic information based on the context of usage. Later, models like word2vec and GloVe[16] improved on existing models by averaging embedding factors and creating a single embedding. These models attempted to create semantic relationships between embeddings. In 2018, Devlin et al. developed the Bidirectional Encoder Representations from Transformers (BERT) model[5], and later a similar development occurred with the ELMO model. These are the most recent and significant improvements in the field of NLP. The introduction of contextualized word embeddings has made a tremendous change in the field of artificial intelligence.

In this case, models are trained on entire sentences rather than single words. These models are simple to use and empirically powerful, and they can be used to tackle a wide range of NLP problems. Further research in the field of contextualized word embeddings has led to the development of a variety of efficient models, such as GPT, RoBERTa, and XLNet[12].

Contextualized word embeddings represent a paradigm shift in natural language processing. Models like BERT[5] and RoBERTa are pretrained on massive amounts of text to learn powerful textual representations. This contrasts with prior word embedding models like word2vec that assign a single vector to each word[16]. BERT introduced the ”masked language modeling” task to pretrain deep bidirectional transformers. By masking out certain words and predicting them based on context, BERT learns relationships between all words in a sentence. The pretrained BERT model can then be fine-tuned on downstream NLP tasks by adding task-specific output layers. RoBERTa built on BERT's approach using additional data and training improvements. For example, RoBERTa used dynamic masking of tokens

during pretraining compared to BERT's static masking pattern. This improved the contextual representations. RoBERTa also employed full sentence modeling rather than BERT's next-sentence pretraining.

Overall, contextualized embeddings have become essential for NLP, powering state-of-the-art results on many language tasks. For financial text analysis, domain-specific variants like FinBERT have been proposed. The transfer learning enabled by pretrained models like BERT and RoBERTa has made implementing advanced NLP accessible even with limited task data. Contextualized representations capture the nuances of language needed for understanding sentiment and semantics.

2.3 Sentiment analysis in finance

The influence of social media and news on the financial market is evident from recent incidents in stock markets around the world.[2] Sentiment analysis in finance is the process of extracting and analyzing the sentiments or opinions of financial analysts shared through social media platforms like Twitter and through different news articles. Sentiment analysis plays a key role in the forecasting and prediction of stock prices.

There is still much research being done to understand how the market reacts to information from different sources. The influence of factors such as environmental disasters, changes in government regulations, and law and order also affects the financial and economic markets. These markets are constantly monitored by people around the world.

Research has shown the power of sentiment analysis in decision-making in the financial market. The extraction of useful and achievable information from these sources has led to the development of hybrid models that integrate sentiment analysis with models based on historical stock prices[2]. The development of sentiment analysis has come a long way, with the transformation from lexicons to transformers making significant changes.[11]

Consider the example tweet:

"today is not great for us. Hold it up apes!! $AMC will rocket tomorrow".

This tweet would first be preprocessed - the $AMC ticker is detected and other noise like hashtags filtered out. The processed text is then fed into the sentiment classifier, which determines the tweet has an overall positive sentiment based on key phrases like "will rocket tomorrow".[15] This positive sentiment signal is provided as an input to the stock forecasting model alongside historical price data. The model has learned associations between bullish sentiment in tweets and upward price movements. So for this tweet, the model would likely predict an increase or upward trend for AMC's stock price over the next day or short term, given the positive expectations expressed about AMC's future performance. In this way, the sentiment analysis component extracts useful signals about market psychology and investor outlooks from noisy social media text. This textual data provides additional context for the model to combine with quantitative pricing data to make more informed forecasts of upcoming price swings and trends. This is an example of sentiment analysis in finance

Data collection is an important factor in financial sentiment analysis. The quality of the data must be assured, as it should check the credibility of the news articles or tweets. In the case of data from tweets, the source or tweet ID should be considered based on their relevance to the field. Quality control is a major part of this forecasting.

2.4 Problem Analysis

In this project, we will apply fine-tuned contextualized word embedding models to our own news and tweet data. This can be used to analyze financial social media data and how it influences stock market forecasting. While referring to related research, we have identified several major issues that need to be addressed when developing hybrid models that combine sentiment analysis and historic stock price data.

- In 2015, Lin et al. found that the volume of news or tweets is positively associated with trading activity. They also analyzed the dynamic interdependency between social media articles and stock prices. Their findings suggest that the quality of the content can affect stock prices and the stock market, but that these factors are not reliable indicators of future stock market movements.[8]
- Trades with low social media attention cannot be correlated with the stock market. Therefore, it is not possible to generalize the interdependency of these articles with the stock market.[8]
- The quality of the data is a major concern in this problem. Bias in the data and the lack of sufficient data are two major quality issues. For example, social media articles or news related to Google or Facebook may receive more attention from traders, which could lead to changes in the stock market.

This project aims to analyze the influence of news and tweets on the stock market separately and to evaluate this influence using different methodologies. This will help to validate, address, and resolve some of the issues identified in related research.

Chapter 3: Design and Implementation

3.1 Frame work

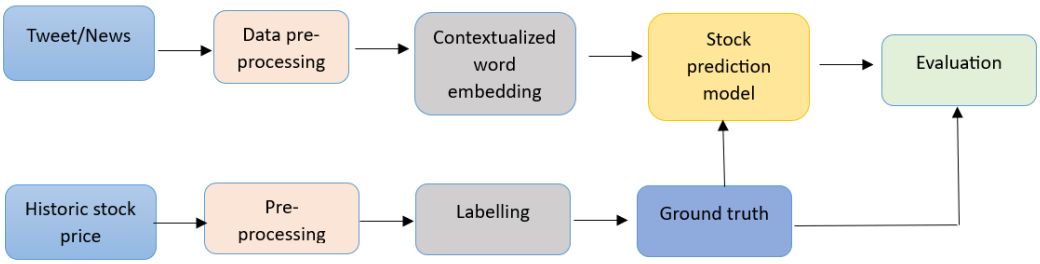

Figure 3.1: Overall framework of the project

Figure 3.1 shows the overall framework of the project, which provides a brief overview of the project flow. This project was developed entirely in Python, with the main Python libraries used being Torch, Pandas, NumPy, and others. First, the news and tweet data were collected and pre-processed. The pre-processed data was then used to perform contextualized word embedding for encoding the text data. Similarly, the historic price data was pre-processed and labeled to obtain the ground truth. The encoded values and ground truth were then merged together to develop a stock prediction model. The model was then evaluated using the developed model and the ground truth.

This is the overall plan of action that was taken during the development of the project. It will be discussed further in the upcoming sections.

3.2 Data Preprocess

3.2.1 Data collection

For this project, we used three different data sources. First, we collected tweet data. At the initial stage of the project, we tried to fetch data directly using the Twitter API. However, this was not a cost-effective method, so we used a public dataset from Kaggle[1] that contains tweet data from 2015 to 2020. This dataset includes tweets related to Apple, Microsoft, Google, and Tesla. It provides the tweet ID, content of the tweet, number of likes, comments, and reposts. There are more than 300,000 unique tweets available in this dataset. We used stocks from the technology domain as the tweets. This is a public dataset and they considered all of the stocks from technology domain.

The second stage of data collection involved accumulating news data. In the initial stage, we tried to fetch news from different credible sources. However, these sources contained a lot of unnecessary data that would have been difficult to clean. We ultimately fetched data from the website Alpha Vantage[2], which provides real-time news related to the stock market and financial situations. Using their API, we collected financial news related to Apple, Google,

[1] https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015-to-2020
[2] https://www.alphavantage.co/

Tesla, Microsoft, NVIDIA, AMD, Pfizer, and Johnson & Johnson from January 2022 to August 2023. This dataset contains more than 60,000 news articles. The fetched news articles include the title, brief description, date published, and topic of the news. So the news data contains stocks from three different domains such as technology, health care and semiconductors. So in this we considered Apple, Google, Tesla and Microsoft are from the technology domain as same as tweet data. Then considered Pfizer and Johnson & Johnson from the healthcare industry and NVIDIA, AMD from the semiconductor industry.

Finally, this project required historic stock prices of the aforementioned tickers during that period. We used Yahoo Finance[3], which has a Python package called yfinance[4], to collect historic stock prices. This data was used to label the data, which can be used for the evaluation and modeling of the stock prediction model.

3.2.2 Data cleaning

In NLP problems, some basic text cleaning is necessary for the better performance of the model. So, let's discuss the cleaning and data preparation of tweet data first. Initially, it is important to clean the body part of the tweet. For this, we removed web addresses, weblinks, hash symbols (#), at signs (@), dollar signs ($), and other special characters. Then, we ensured that the body contains a reasonable length for the better working of the model. For this, we removed tweets without a reasonable length. The relevance of the tweet is also an important factor for the analysis of the stock market. To obtain a rough analysis of the relevance of the tweet, we created an additional column named score. This column is the sum of the number of comments, likes, and retweets. Using this score, we removed data with scores below 10. These were the major cleaning steps performed on the tweet data. Finally, the data was aligned by performing a groupby with date and ticker symbol for easy merging with labels.

The news data type contains a title and description of the news in string format. Therefore, it needs to be cleaned well. Most of the titles of news are very short. To resolve this, we concatenated the title and short description of the news into a single column named body. The body part was cleaned by removing the special characters as mentioned above. We also performed some additional cleaning, such as removing numerical characters and stop words. Then, we removed data without a reasonable length. This news data contains all of the news related to the corresponding companies. Therefore, it is necessary to filter them based on the relevancy of the news. As an initial cleaning step, the data was filtered using the topic of the data. For this, we used data from the following topics: economy-fiscal, economy-monetary, IPO, real estate & construction, merger & acquisition, energy & transportations, finance, financial markets.

3.2.3 Data labelling

The historic stock price details of the corresponding stocks during the period of time need to be labeled to make them a ground truth for developing the stock prediction model and for the evaluation of the model. The historic stock price fetched from yfinance contains the opening price, closing price, volume, and some other related data for each ticker for that whole period of time. It can be classified into three different classes based on the change in price for each day. The three classes are: upgrowth in stock price, down growth in stock price, and no change in the rates. To classify the stock prices, we tried two different methods:

[3]https://uk.finance.yahoo.com/

[4]https://pypi.org/project/yfinance/

- We analyzed the transformed values and created a threshold to make a classifier for three different classes. In this method, we used the closing price to calculate the growth and then transformed it.

$$Growth(t) = \frac{MinMaxScaler(price(t+1) - price(t))}{price(t)} \tag{3.1}$$

Using this equation, we calculated the scaled growth of the closing price values for each day. We used MinMaxScaler from scikit-learn to fit and transform the data. Then, by analyzing the distribution of the scaled growth graph, we made a classification. We labeled scaled growth that is less than 0.40 as 0 (down growth), the rest valued between 0.40 to 0.50 as 1 (preserve), and the rest of them as 2 (up growth).

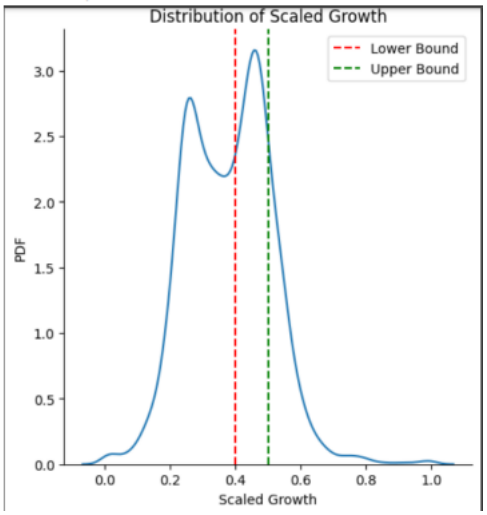

Figure 3.2: Distribution of scaled growth

- The next method is similar to the last one, but it uses a threshold method by taking the logarithmic difference between the closing stock prices of two consecutive days.

$$Log(Growth(t)) = Log(price(t)) - Log(price(t+1)) \tag{3.2}$$

Using this equation, we obtained the logarithmic growth of the stock prices. We labeled values below -0.05 as 0 (down growth), values between -0.05 to 0.05 as 1 (preserve), and values greater than 0.05 as 2 (up growth).

These two methods are very effective and useful for classification models. However, it is better to analyze their effectiveness empirically. We can observe that the second method would give better results. The first method would lead to class imbalance after classification, but the second method does not have this problem. The logarithmic difference gives a more robust measure of growth than the simple difference. Additionally, the logarithmic difference is computationally more efficient than the other method.

3.2.4 Data splitting

Data splitting is a necessary step in this process. The tweet data was split into three sets: train data, validation data, and test data. The validation set is used during training to tune hyperparameters and evaluate model performance at each epoch, while the test set is held out completely until final evaluation after training is complete. Validation guides improvements

to the model during development, whereas testing provides an unbiased final check on model generalization. The tweets up to 2018-12-31 were used as the training data, the data from 2019 January to 2019-06-30 was used as the validation data, and the rest of the tweets were used as the test data.

A similar process was performed on the news data, which contains news from a period of 1.5 years. The first year of news data was used as the training data, the next 3 months of data were used as the validation data, and the rest of the data was used as the test data.

3.3 Stock price prediction model

3.3.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) was a revolutionary change in the field of contextualized word embedding. Before that, the meaning of a word was independent of the context of the sentence. The BERT model employs the masked language model (MLM) method by randomly masking some of the tokens in the sentence, which are later used for next sentence prediction or other similar tasks.

BERT is an autoregressive language model, which means it can forecast the future based on the past incidents. It is pretrained by maximizing the likelihood between the tokens that are given as input to the model [11]. For example, if we consider the sentence as x, then it can be represented as $x = [x_1, x_2, x_3, ...x_N]$. Then, let's assume $\hat{x}$ is the same sentence with masked tokens, and $\overline{x}$ is the array of masked tokens.

$$\max_{\theta} \log p_\theta(x|\hat{x}) \approx \sum_{t=1}^{T} m_t \log p_\theta(x_t|\hat{x}) = \sum_{t=1}^{T} m_t \log \frac{\exp(H_\theta(\hat{x})_t^T e(x_t))}{\sum_{x_0} \exp(H_\theta(\hat{x})_t^T e(x_0))} \tag{3.3}$$

In this x_0 denotes the embedding of the token, $m_t = 1$ if x_t is the token masked on x[4]. Then H_θ denotes the transformer which change each sequence into a vector format[11]. In the training phase, BERT reconstructs $\overline{x}$ from $\hat{x}$, by assuming that all of the masked tokens $\overline{x}$ are mutually independent. Then, the approximation of the joint conditional probability can be used here

The BERT model consists of a tokenizer that converts each token in the sentence into a vector representation. The first sentence begins with a [CLS] token and there is a [SEP] token at the end of each sentence. The figure below shows an example of the input representation for BERT.

The encoder consists of a stack of 12 self-attention feed-forward networks. These are the important components of the BERT model and this mechanism is the reason for the improvement of the BERT model over previous models. The self-attention model helps to learn the relationships between words in both directions, i.e., before and after the word. This is why it is called a bidirectional transformer. Finally, there is a pooler and a classification layer on top of the model. The pooler takes the output of the encoder and produces a single vector output. The final linear layer helps to perform classification tasks.

So let's explain the sentiment analysis process in BERT simple version. Sentiment analysis is the task of determining the emotional tone of a text, such as whether it is positive, negative,

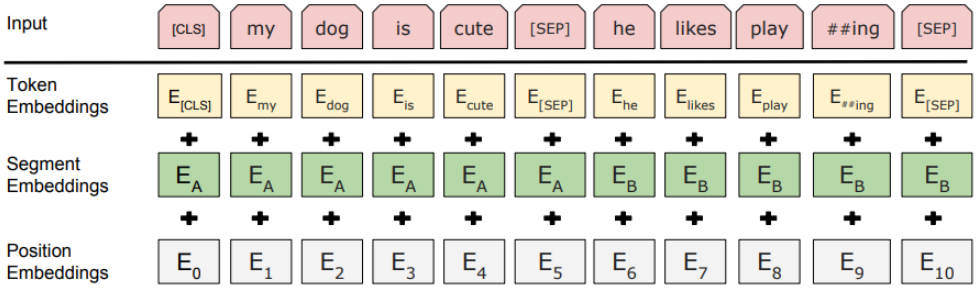

Figure 3.3: BERT input representation[5]

or neutral. To do this, sentiment analysis models need to be able to identify the sentiment-bearing words in a text. BERT can do this by looking at the context of each word and determining its sentiment based on the sentiment of the surrounding words.

For example, the word ”love” is generally considered to be a positive word. However, the sentiment of the word ”love” can change depending on the context in which it is used. For example, the sentence ”I love my dog” is likely to be interpreted as a positive sentiment, while the sentence ”I love to hate you” is likely to be interpreted as a negative sentiment.

BERT can learn to distinguish between these different interpretations of the word ”love” by looking at the context in which it is used. In the sentence ”I love my dog,” the word ”love” is surrounded by other positive words, such as ”dog” and ”I.” This tells BERT that the sentiment of the word ”love” in this context is positive. In the sentence ”I love to hate you,” the word ”love” is surrounded by other negative words, such as ”hate” and ”you.” This tells BERT that the sentiment of the word ”love” in this context is negative.

3.3.2 RoBERTa

RoBERTa (Robustly optimized BERT) is a fine-tuned and improved version of BERT[9]. It is a commonly used contextualized word embedding model nowadays. It has a similar architecture to BERT, but the major differences between BERT and RoBERTa are in their training data and masking. RoBERTa uses a larger training corpus than BERT, which allows it to learn better representations of words and phrases. BERT uses static masking, which means that the same tokens are masked in each training instance. RoBERTa uses dynamic masking, which means that the tokens are masked differently for each training instance. This makes RoBERTa more robust to noise and helps it to learn more generalizable representations. Finally, RoBERTa uses a byte-level BPE tokenizer, while BERT uses a word-level tokenizer. This means that RoBERTa can learn more fine-grained relationships between words.

Here, we used the pre-trained BERT and RoBERTa models to train our tweet and news dataset. Both models have a hidden size of 768 and 12 attention heads. They also have 12 hidden layers, which allows them to attend to 12 different words at once. We initialized the maximum word length as 512 for better performance of the model. The model was trained with a learning rate of 1e-04. The batch size was set to 16 for the training batch and 32 for the rest.

3.3.3 Modified RoBERTa

While the original RoBERTa model provides powerful contextualized representations, we made some modifications to adapt it to our tweet/news classification task. Specifically, we added an additional dropout layer after the RoBERTa encoder with a dropout rate of 0.3, which acts as a regularizer to prevent overfitting. We also added a task-specific classification head on top of the RoBERTa encoder outputs. This consists of two linear layers to reduce the dimensionality from 768 to 512 and then 256, with ReLU activation functions. Finally, a third linear layer projects the 256-dimensional representation to our 3-way tweet classification outputs. Adding this simple classification head allows the pretrained RoBERTa encoder to be finetuned on our dataset and task. During training, the gradients are backpropagated through the entire network architecture to update both the RoBERTa parameters as well as the task-specific classification head. This end-to-end fine-tuning approach leverages the semantic knowledge within RoBERTa while adapting it to our particular tweet classification problem.

3.3.4 Hybrid GRU

The hybrid model combines a RoBERTa encoder with a GRU recurrent network to leverage both contextualized representations and sequence modeling for stock prediction. The input sequences of tokenized tweets/news are passed through the pretrained RoBERTa model to generate embedded vector representations. We take the last hidden state output from RoBERTa, which summarizes the full context of the input sequence.

This output is fed into a 2-layer GRU with a hidden size of 256. The GRU can capture long-range dependencies in the sequence, modeling how the sentiment evolves across the tweets/news timeline. The last hidden state of the GRU condenses the sequential information into a fixed-length vector. This vector is passed to a simple linear layer to make a prediction of the stock movement classification.

Using a 2-layer GRU provides enough complexity to model relevant sequential patterns, without being too computationally expensive. The smaller GRU hidden size of 256 (compared to RoBERTa's 768) also helps regularize the model. Overall, this architecture allows the pretrained knowledge within RoBERTa to be augmented with the sequence modeling capabilities of a GRU for our stock forecasting task. The joint training and end-to-end gradients allow the two components to be optimized together.

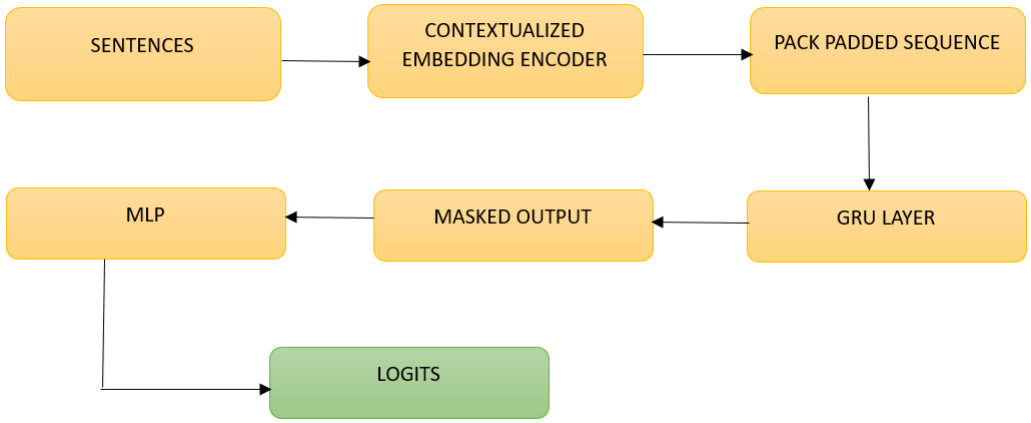

Figure 3.4: Framework of GRU baseline model

3.4 Evaluation metrics

Several evaluation metrics provide useful insights into model performance on the 3-class tweet sentiment classification task: Accuracy measures the overall proportion of correct predictions out of total predictions made. It gives the percentage of tweets correctly classified as positive, negative or neutral:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.4}$$

Where TP are true positives, TN are true negatives, FP are false positives, and FN are false negatives. However, accuracy can be misleading if the classes are imbalanced. So we also calculate precision, which is the percentage of positive predictions that were actually correct:

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

For the positive sentiment class, precision calculates what fraction of tweets predicted as positive truly expressed positive emotion. High precision indicates a low number of false positives. In addition, we compute recall or sensitivity, which is the percentage of true positives that were correctly predicted by the model:

$$Recall = \frac{TP}{TP + FN} \tag{3.6}$$

$$F1Score = 2\frac{Precision * recall}{precision + recall} \tag{3.7}$$

Recall helps assess how many positive tweets the model was able to correctly classify. F1 score provides a balance between precision and recall using their harmonic mean. The F1 score reaches its optimal value at 1 and worst value at 0. It captures both false positives and false negatives. We also monitor the loss, here we use cross-entropy loss, to evaluate how well the model is fitting the true label distributions for each sentiment class. Lower loss over training signifies better learning of the patterns. These metrics together provide a comprehensive view of model performance - from overall accuracy to the balance of precision and recall, and the ability to fit the training data. Tracking them helps identify strengths, weaknesses, and error patterns in sentiment classification. The goal is to improve precision, recall, F1, and accuracy while lowering the loss across all sentiment categories.

Chapter 4: Evaluation

4.1 Experimental setup

We conduct experiments training different models on the tweet and news classification dataset. We train the models using the Adam optimization algorithm, which is well-suited for handling the noise and variability in textual data. Adam combines aspects of RMSProp and stochastic gradient descent with adaptive learning rates. This makes it robust and efficient for training deep neural networks on large unstructured datasets like text. The learning rate of 1e-04 provides a small step size to nudge the model parameters towards optimal values for sentiment classification. The lower learning rate prevents aggressive fluctuations that could cause instability. For the loss function, we use cross entropy loss, also known as negative log likelihood. This measures the divergence between the model's predicted class probabilities and the true label distribution. Minimizing cross-entropy encourages the model to produce higher probabilities for the correct sentiment classes. Cross entropy is a natural fit for multi-class classification problems like our 3-way positive/negative/neutral tweet categorization. By penalizing incorrect low probabilities and rewarding accurate high probabilities, cross entropy guides the model to distinguish among the sentiment classes. We leverage validation set monitoring, stopping training if validation loss doesn't improve for 3 epochs. This early stopping prevents overfitting to the training data so the model generalizes better. The optimization strategy balances efficiency and stability to effectively train sentiment classifiers from noisy textual finance data.

We trained our data with BERT base uncased, RoBERTa base, Modified RoBERTa model and a hybrid GRU model. All models are implemented in PyTorch and trained on TESLA T4 GPUs. We use a linear learning rate warm up schedule over the first 10% of training steps, before decaying the learning rate linearly to 0.

To evaluate the model's performance on different stock sectors, I sampled an equal number of news articles from each domain to create balanced evaluation sets. Specifically, I randomly selected portions of the full news dataset related to technology, healthcare, and semiconductor stocks to generate three subsets. This left three evaluation sets with an equal number of news posts per domain. Then tested the pretrained sentiment classification models separately on each industry-specific subset. Comparing performance between the sectors enabled analysis of how predictive the models are on different domains when data amounts are controlled. If accuracy varied significantly between industries, it indicates the model better exploits the news data for certain types of stocks

The trained models are evaluated on the held-out test set. We report accuracy, precision, and recall metrics. The following sections analyze the experimental results.

4.2 Results and Analysis

Table 4.2 shows the performance of the different models on the tweet and news classification tasks. We can see that our modified RoBERTa model achieves the best accuracy on the tweet classification task, demonstrating the effectiveness of our architecture modifications for this dataset. The hybrid GRU model comes closest with an accuracy of 0.561. For news classification, the modified RoBERTa also provides a noticeable improvement, increasing

accuracy from 0.521 with standard RoBERTa to 0.539. However, the gains are more modest compared to the tweet task, suggesting there is still room for improvement on classifying news articles. Overall, the results validate our approach of adapting RoBERTa for these short-text categorization problems. The improvements in accuracy over both the standard BERT and RoBERTa baselines highlight the benefits of our additional regularization and task-specific classification head.

Model	Tweet				News			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
BERT	0.521	0.546	0.522	0.533	0.493	0.499	0.489	0.496
RoBERTa	0.538	0.547	0.528	0.542	0.521	0.538	0.524	0.529
Modified RoBERTa	0.551	0.545	0.557	0.546	**0.539**	0.542	0.538	0.540
Hybrid GRU	**0.561**	0.528	0.545	0.543	0.532	0.541	0.537	0.536

Table 4.1: Model Performance Comparison

The comparatively weaker performance of the baseline BERT and RoBERTa models highlights some of the challenges in applying pretrained language models directly to tweet and news classification. As large-scale masked language models, BERT and RoBERTa are trained on book and Wikipedia text which differs significantly from the short-form writing style of tweets and news headlines. Furthermore, both models lack an explicit classification component tailored for this task. As a result, the representations learned by the standard BERT and RoBERTa encoders may not fully capture the nuances needed to distinguish between our fine-grained tweet and news categories. Their general purpose encodings struggle to adapt to the informal abbreviations, creative spellings, and unique lingo present in tweets. Additionally, the lack of a classification head tuned for our 3-way categorization means the models are limited in converting the encodings into useful category predictions. Our modifications address these issues by regularizing the model to avoid overfitting on the small datasets, and adding a classification component designed for the task. This allows RoBERTa's semantic knowledge to be specialized for short-text classification through end-to-end fine-tuning.

To gain further insight into the models' prediction errors, we analyze the confusion matrices on the tweet test set. The confusion matrix provides the breakdown of predicted versus true labels for each class. We normalize the confusion matrix so that each row sums to 1, allowing us to view the percentage of examples classified into each category.

Examining the confusion matrices reveals some patterns in the models' mistake types. For the standard BERT model, we see a good amount of confusion between the positive and neutral classes, with many positive examples misclassified as neutral. This suggests BERT struggles to distinguish the more nuanced differences between positivity and neutrality in tweets. BERT also exhibits some confusion between negative and neutral tweets. In comparison, our modified RoBERTa and hybrid GRU achieves a cleaner diagonal confusion matrix indicating it models the distinctions between the classes more accurately. The reductions in off-diagonal entries show RoBERTa better learns the subtle characteristics to differentiate positive from neutral tweets, as well as negative from neutral. This aligns with the accuracy improvements from fine-tuning RoBERTa for this task. However, some confusion remains between certain pairs of classes indicating there is still room for improvement in modeling the linguistic nuances of tweets. Overall, the confusion matrices provide useful fine-grained evaluation of the models beyond aggregate accuracy measures.

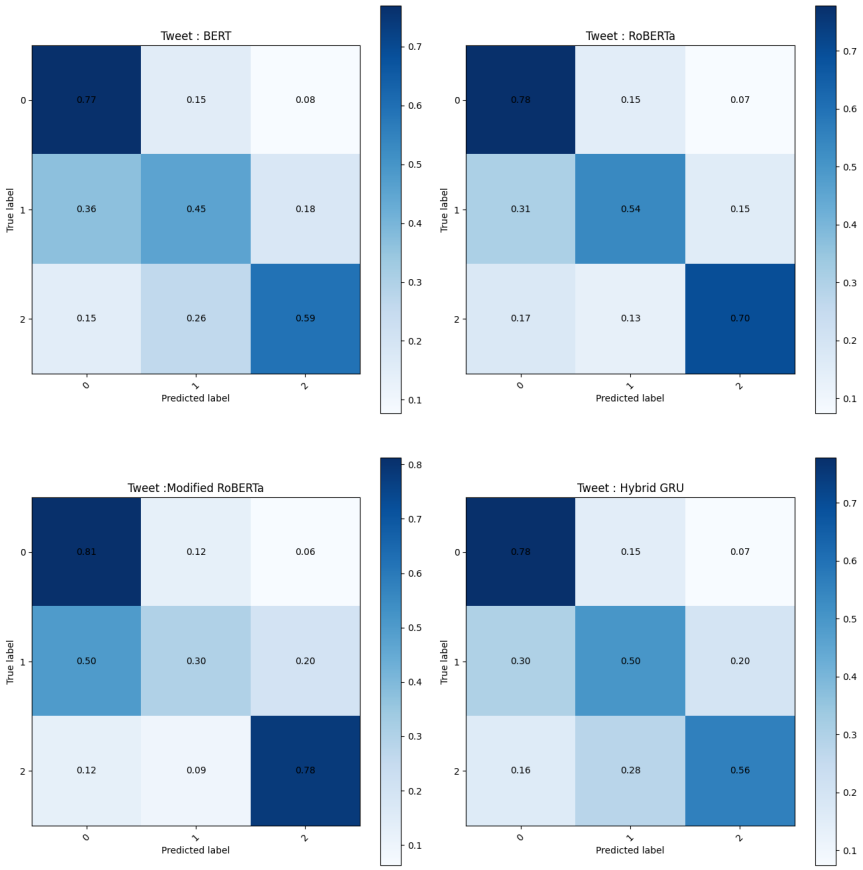

Figure 4.1: Normalized confusion matrix of different models with tweet data

Also, the confusion matrix shows that the four models have a higher misclassification rate for neutral sentiment than for the other two sentiments. This means that the models are more likely to misclassify neutral tweets as either upward growth or negative growth. There are a few possible reasons for this. First, neutral sentiment is often more difficult to identify than positive or negative sentiment. This is because neutral tweets do not express a clear opinion or emotion. Second, the models may not have been trained on enough neutral tweets to learn to identify them accurately. Third, the models may be biased towards classifying tweets as positive or negative, even if they are neutral. This misclassification is also a reason for the pull back on the results.

4.3 Influence of tweet and news

Our experiments demonstrate the promise of using natural language processing to extract signals from textual data that can inform stock market prediction. The tweet dataset covers the period from 2015-2020, while the news dataset is from 2022-2023. Despite being different time periods, several key insights emerge. First, the strong performance of our adapted RoBERTa model on both tweets (55.1% accuracy) and news (53.9% accuracy) shows the effectiveness of deep learning for short-text classification. The models can uncover the sentiment and topical content to enable historical analysis of how public discourse relates to market movements. Second, the higher accuracy on tweets suggests that informal, individual opinions on social media provide a useful barometer of investor sentiment. Finally, while news accuracy is slightly lower, timely analysis of authoritative news sources can still give valuable perspective on company and industry developments.

The following additional points were analyzed after comparing the results

- Volume of data and Relevance: Tweets are shorter than news articles, which means that there is more data available for training a sentiment analysis model. This is because people tend to tweet more often than they write news articles. Additionally, tweets are often more relevant to current events, as they are typically posted in real time. This makes them a valuable source of data for sentiment analysis, as they can be used to track public opinion on a variety of topics.
- Tweets are more direct in tone: Tweets are typically written in a more informal and direct tone than news articles. This is because tweets are often used to express personal opinions or thoughts, rather than to provide objective reporting. This can make it easier to identify the sentiment of a tweet, as the author's feelings are often more clearly expressed.Also tweets used to specify keywords among the tweets.
- Impact of tweets are more on markets: Tweets can have a significant impact on markets, as they can be used to spread news and information that can influence investor sentiment. For example, if a celebrity tweets about a particular stock, this can cause the stock price to go up or down. This makes tweets a valuable source of data for market analysis.
- Chance for noise in news data due to analysis, opinions included: News articles often contain opinions and analysis from the author, which can make it difficult to identify the objective sentiment of the article. This is because the author's opinion may bias the way that they report the news.

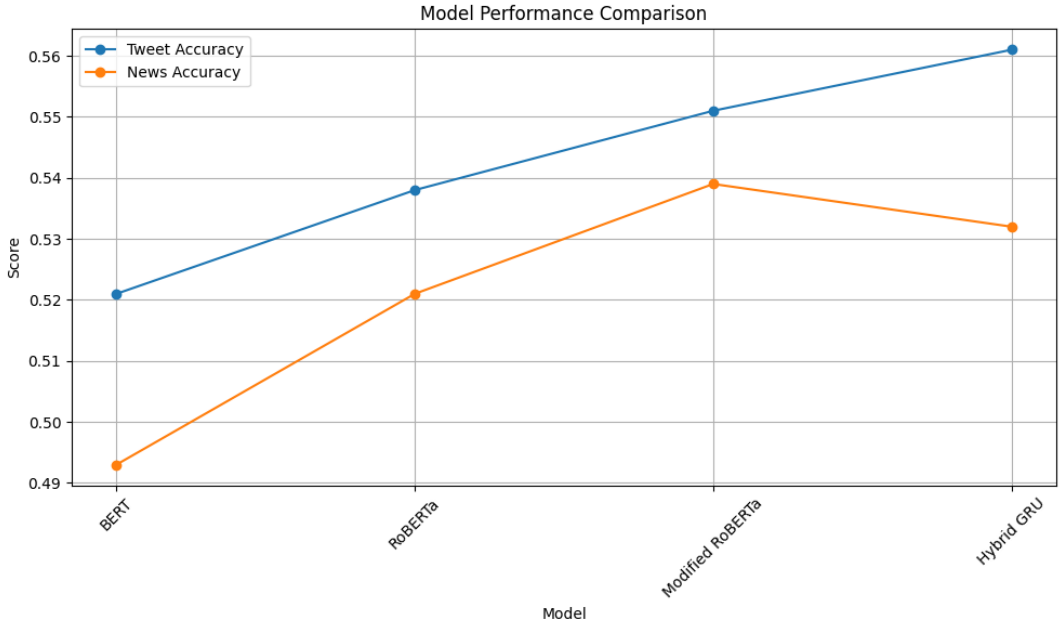

Figure 4.2: Accuracy plot for different models on news and tweet data

4.4 Performance comparison among different stock domains

In order to evaluate the impact of news sentiment on different stock sectors, I leveraged the modified RoBERTa model which achieved the highest accuracy on the full news dataset. As described previously, I had curated balanced subsamples of the news data related to technology, healthcare, and semiconductor stocks.

Since modified RoBERTa performed the best in initial experiments, I used this model architecture to make the domain-specific comparison. I tested separate modified RoBERTa classifiers on each sector-specific subsample. Then I evaluated the sentiment analysis accuracy on held-out test sets for each domain. This enabled an equal comparison of how predictive the news sentiment was for each industry when controlling for data volume. The modified RoBERTa model provides a robust baseline tuned for financial text data. Any variability in accuracy between the sectors can be attributed to differences in how effectively this model exploits news data, rather than model selection.

Domain	Accuracy	Precision	Recall
Technology	0.632	0.636	0.630
Health care	0.552	0.547	0.551
Semi-conductor	0.537	0.531	0.534

Table 4.2: Test result using Modified Roberta among different stock domains

The results showed that modified RoBERTa achieved significantly higher accuracy on technology stocks compared to healthcare and semiconductors. This suggests the model better captures sentiment signals relevant to predicting technology stock movements based on news articles discussing these companies. The comparative analysis provides insights into how predictive power differs across sectors when using a consistent state-of-the-art model.

Chapter 5: Conclusion

This thesis presented stock trend prediction models using contextualized word embedding techniques like BERT and RoBERTa. Comparing four model architectures showed the value of adaptation through our modified RoBERTa and a hybrid GRU approach, which outperformed baseline BERT and standard RoBERTa. Evaluting the models on both tweet and news data revealed tweets provided greater accuracy improvements, acting as a real-time barometer of investor psychology. However, news also contributed useful signals at lower gains when compared to tweet. An interesting finding was the technology stocks showing the largest benefit from sentiment modeling on domain-specific content. Though this requires caution around potential dataset biases. The results validated incorporating textual data can enhance stock forecasting through robust deep learning models. However, data quality is an important factor. Pretraining on large financial corpora could improve domain adaptation. In addition to that, sourcing representative data across sectors can help mitigate bias. This work laid a promising foundation for hybrid multimodal prediction. Future avenues include integrating different data types like prices, fundamentals, and alternative signals into an end-to-end model. Advances in deep learning for both time series and text data can be combined. Overall, this thesis demonstrated the viability of contextualized word embeddings applied to financial forecasting. With thoughtful data curation and model architecture tuning, NLP has extensive potential to extract valuable insights from textual data to inform trading decisions.

5.1 Limitations and Future work

While our results are promising, there are some limitations to this study that provide opportunities for future work. The major limitation is the mismatch in time periods for the tweet (2015-2020) and news (2022-2023) datasets. This makes direct comparison and joint modeling difficult, as the relationships between text and markets likely shifted over this gap. In the future, collecting matched datasets from the same time period will allow for more rigorous analysis. Another key limitation was the reliance on free data sources like kaggle free dataset and Alpha Vantage news. Proprietary feeds with full historical access can be expensive, restricting data diversity and scale in this academic research.

Additionally, the news dataset is relatively small. This restricts the complexity of models that can be effectively trained. Also the news from technology domains dominates over the others and it makes a bias on news dataset. Expanding the news corpus would enable using larger pretrained language models fine-tuned on financial text.

To address these limitations, future work can explore multi-window learning techniques. The key idea is to divide the timeline into multiple windows and train separate models on each window. This allows capturing non-stationary relationships while still leveraging some shared knowledge across windows. For example, we could train RoBERTa models on 2015-2017, 2018-2020, and 2022-2023 windows. Regularization techniques can be used to prevent overfitting and enable transfer learning across the windows. So using these methodologies it can develop systems that predict stock movement predictors by fetching real-time news or tweets. Through it would be better build financial decision systems than a sentiment analysis system.

Incorporating other data modalities like price charts, fundamentals, and domain knowledge

graphs could provide useful signals alongside the text data. This multimedia approach can give a more complete picture of the interconnected drivers of market movements. Overall this work provides a foundation for an array of future research directions in conversational finance and stock prediction from alternative data.

Appendix A: First Appendix

Code for this project:

https://github.com/hari599/stock_trend_analysis

Tweet dataset:

https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015-to-2020

Bibliography

[1] Gerardo Alfonso, A. Daniel Carnerero, Daniel R. Ramirez, and Teodoro Alamo. Stock Forecasting Using Local Data. *IEEE Access*, 9:9334–9344, 2021. Conference Name: IEEE Access.

[2] Dogu Araci. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models, August 2019. arXiv:1908.10063 [cs].

[3] Adebiyi A. Ariyo, Adewumi O. Adewumi, and Charles K. Ayo. Stock Price Prediction Using the ARIMA Model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pages 106–112, March 2014.

[4] Sven Crone and Christian Koeppel. Predicting exchange rates with sentiment indicators: An empirical evaluation using text mining and multilayer perceptrons. *IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, Proceedings (CIFEr)*, pages 114–121, October 2014.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. arXiv:1810.04805 [cs].

[6] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction, February 2019. arXiv:1712.02136 [cs, q-fin].

[7] Vikas Khullar, Rishu Chhabra, Mohit Angurala, MRM Veeramanickam, Kirandeep Singh, and Vikas Lamba. Indian National Stock Exchange Crude Oil (CL=F) Close Price Forecasting Using LSTM and Bi-LSTM Multivariate Deep Learning Approaches. In *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, pages 839–842, April 2023.

[8] Shen Lin, Da Ren, Wei Zhang, Yongjie Zhang, and Dehua Shen. Network interdependency between social media and stock trading activities: Evidence from China. *Physica A: Statistical Mechanics and its Applications*, 451:305–312, June 2016.

[9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, July 2019. arXiv:1907.11692 [cs].

[10] Karn Meesomsarn, Roungsan Chaisricharoen, Boonruk Chipipop, and Thongchai Yooyativong. Forecasting the effect of stock repurchase via an artificial neural network. In *2009 ICCAS-SICE*, pages 2573–2578, August 2009.

[11] Kostadin Mishev, Ana Gjorgjevikj, Irena Vodenska, Lubomir T. Chitkushev, and Dimitar Trajanov. Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers. *IEEE Access*, 8:131662–131682, 2020. Conference Name: IEEE Access.

[12] Saloni Mohan, Sahitya Mullapudi, Sudheer Sammeta, Parag Vijayvergia, and David C. Anastasiu. Stock Price Prediction Using News Sentiment Analysis. In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 205–208, April 2019.

[13] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and Shahab S. Deep Learning for Stock Market Prediction. *Entropy*, 22(8):840, August 2020. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

[14] Jane A. Ou and Stephen H. Penman. Financial statement analysis and the prediction of stock returns. *Journal of Accounting and Economics*, 11(4):295–329, November 1989.

[15] Yulong Pei, Amarachi Mbakwe, Akshat Gupta, Salwa Alamir, Hanxuan Lin, Xiaomo Liu, and Sameena Shah. TweetFinSent: A Dataset of Stock Sentiments on Twitter. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*, pages 37–47, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[16] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[17] Sumeet Sarode, Harsha G. Tolani, Prateek Kak, and C S Lifna. Stock Price Prediction Using Machine Learning Techniques. In *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 177–181, February 2019.