Vol. 34, No. 2 (2024) | DOI: 10.37190/ord240206

OPEN ACCESS

Operations Research and Decisions

www.ord.pwr.edu.pl

OPERATIONS RESEARCH AND DECISIONS QUARTERLY

Predicting stock market by sentiment analysis and deep learning

Süreyya Özöğür Akyüz[1] Pınar Karadayı Ataş[1*] Aymane Benkhaldoun[2]

[1] *Department of Mathematics, Faculty of Engineering and Natural Sciences, Bahçeşehir University, Istanbul, Turkey*

[2] *Department of Computer Engineering, Faculty of Engineering and Architecture, Arel University, Istanbul, Turkey*

[3] *Department of Big Data Analytics and Management, Faculty of Engineering and Natural Sciences, Bahçeşehir University, Istanbul, Turkey*

[*] *Corresponding author, email address: pinaratas@arel.edu.tr*

Abstract

The stock market may be unpredictable; understanding when to purchase and sell can greatly assist businesses and individuals in maximizing profits and minimizing losses. Many companies have previously modified time-series analysis, a data mining technique, to forecast stock price movement. The idea of textual data mining has recently come up in debates about stock market forecasts. In this study, five of the largest firms' historical stock prices were used to train two deep learning models—long short-term memory (LSTM) and one-dimensional convolutional neural network (1D CNN), then the results of all the models were compared. To connect price value fluctuations with the general public, sentiment scores were offered in addition to stock price values by employing natural language processing techniques (TextBlob) to tweets.

Keywords: *stock market, Twitter, deep learning, sentiment analysis*

1. Introduction

Predicting stock market prices has long been a crucial subject in the financial world. The rule of supply and demand drives the stock market; to make significant profits, investors must understand when to purchase and sell. Fundamental analysis stock valuation is the oldest and most used method. It seeks to forecast future earnings, dividends, and risk to determine the companies' actual worth and, consequently, forecast the volatility of that stock, as explained by Sinisa et al. [2]. Machine learning has made it possible to predict the stock market in new and more precise ways. For short-term forecasting, the support vector machine (SVM) method is the most widely employed by Lie et al.[8] and Long et al [26].

Sometimes it is difficult to predict changes in the stock market using only stock price data because big news and events have a significant impact on the market. According to Ghanem and Rosvall's [13], big global events have varying effects on stock markets. Finding methods to comprehend and take advantage

Received 26 October 2022, accepted 28 April 2024, published online 8 July 2024
ISSN 2391-6060 (Online)/© 2024 Authors

The costs of publishing this issue have been co-financed by the program *Development of Academic Journals* of the Polish Ministry of Education and Science under agreement No. RCN/SP/0241/2021/1

of those industry adjustments is crucial. Similar to stock market investing selections, their choices can be strongly influenced by their emotional condition. While people are uncertain or gloomy about the future, they are less willing to invest and are more cautious when trading on the stock market. When people are pessimistic about the future, the reverse occurs. Breaban and Noussair [3] investigated how emotions affect stock market behavior. They discovered that although good emotions tend to predict purchases, negative emotions tend to predict price declines and selling. Therefore, taking into account people's emotions when making stock market predictions is likely to produce excellent outcomes.

The sentiment analysis has been used by numerous scholars to create techniques for stock market forecasting. According to general agreement, deep learning algorithms can forecast the stock market using textual information. In this investigation, we separately applied LSTM and 1D-CNN to forecast stock market movement using tweets. Every study used a unique method for sentiment analysis, and several of them made use of pre-made sentiment libraries. The pre-defined sentiment library was selected by determining which of the five most popular sentiment libraries has the highest association with historical stock market price data, which distinguishes this study from earlier studies in two ways. The second involves creating and contrasting LSTM and 1D-CNN models using the same sentiment from Twitter. The second involves creating and contrasting LSTM and 1D-CNN models using Twitter's identical stock price and sentiment data.

The purpose of this paper was to determine which sentiment library is best for anticipating changes in stock market pricing, followed by which current deep learning algorithms are most effective for stock market forecasting. This study will forecast stock market change by turning textual information gathered from Twitter into measurable statistics. It is assumed that Twitter's news, commentary, and emotions have a substantial impact on the stock market. The selection and analysis of several NLP libraries will be used to derive sentiment scores from the Twitter data. Two deep learning models (LSTM and 1D-CNN) that forecast stock price changes and compare will be developed using the sentiment scores along with historical price data.

2. Literature review

The relationship between the stock market and social media has been researched extensively. Using a quantitative approach, Abu-Taleb and Nilsson [1] investigated how social media affects investment choices and discovered a positive correlation between the two. Gilbert and Karahalios [14] discovered that an increase in anxiety, dread, and worry, as seen in a dataset of 20 million LiveJournal postings, forecasts the S&P500 index's downward trend. These studies demonstrate a clear connection between social media sentiment and stock market action. According to Kemp [40], a typical person spends two hours and twenty-four minutes on social media, with usage in first-world nations being substantially greater. As a result, social media is a fantastic area to gather public sentiment and opinions. Sentiment analysis is a fantastic tool for identifying and measuring the emotions expressed on social media.

Numerous academics have recently begun examining the connection between market sentiment and changes in stock prices. To extract sentiments from the text, researchers utilize sentiment analysis. Some people gather textual data from social media posts, particularly those on Twitter, to discover this connection. French tweets were utilized by Martin [27] to forecast the closing values of the CAC40, a benchmark

French stock market index that is a capitalization-weighted assessment of the 40 most important stocks listed on Euronext Paris. To estimate the price of five distinct stocks on any given day, Khatri and Srivastava [22] analyzed tweets that referenced the equities. For sentiment analysis, they divided tweets into four parts by them, which are up, down, happy, and rejected. Xu and Keelj [46] took advantage of StockTwits, a website similar to Twitter. They used a manually annotated corpus, an SVM, and 16 randomly selected equities that were often discussed on Stocktwits to generate sentiment scores for the postings. Porshnev et al. [32] took a different approach and determined the emotions of the general public by counting the number of emoticons used in tweets. The authors then combined this information with historical market data to forecast movements of the DJIA and S&P 500 indices.

Tinku Singh et al. [39] examined and compared various state-of-the-art machine learning models, namely ARIMA, SVR, LSTM, and XGBoost, as well as their ensembles using weighted averaging and boosting techniques, on a 1-year and 5-year timeline. The study utilizes data collected from two companies, HDFC and Sun Pharma, listed in the Nifty 50 stocks on the National Stock Exchange (NSE) in India. The findings of this paper may contribute to the existing literature on the machine learning model's performance in predicting stock prices and aiding investors in making informed decisions. Mukhtar [17] observed that posts with higher engagement are a good indicator of other people's sentiment agreeability. He used sentiment analysis of tweets to create stock market forecasts. In particular, a post with more likes, comments, and reactions is preferable to one with fewer because it might reveal the opinions of many users as a whole. Another recent paper [6] examined several related sources that employed SM data and SA approaches to predict elections. One discovery was that, as compared to less well-liked choices like the Twitter Search API, Twitter's Streaming API was by far the most used to extract data from SM. This study also discovered that the majority of the examined sources combined SA methods with volumetric data to predict elections.

More information may be found in the work of Catelli et al. [5, 30] which contrasts an NN created specifically for SA (BERT) versus lexicon-based techniques in identifying the sentiment in Italian e-commerce evaluations and admits the difficulties in utilizing SA in languages other than English and Chinese. They focus on the fact that popular SA NNs require sizable datasets for efficient model training, which are frequently unavailable for non-English/Chinese languages. The present study [47] introduces a novel sequential three-way stock prediction model within the multilevel granular structure of hybrid data. Three multi-granularity methods are constructed to integrate price and textual data, specifically, high-frequency data with features extracted by sentiment analysis, high-frequency data with features extracted by BERT, and low-frequency data with features extracted by BERT. Sequential three-way decisions are then employed to predict the future trend of each test sample at appropriate granularity. Furthermore, an autonomous attention mechanism is integrated into the CNN and RNN to handle large embeddings and long-time series dynamically. Comparative experiments are conducted on 5-year stock data to demonstrate that the proposed models outperform the non-sequential two-way predictions.

Since financial news has less noise than general news, other academics use it to collect textual data. They choose a single passage from the news article or the headline to feed into their algorithm. To forecast the stock market, Xie and Jiang [45] merged stock price data from the largest 20 Chinese equities based on trade volume with sentiment data from financial news items acquired using a bag of words approach. The sentiment dictionary Harvard IV-4 and the Loughran and McDonald financial lexicon were utilized by Li et al. [24] to obtain the sentiment score while using financial news as textual data. However, Shastri et al. [37] only analyzed market data that is related to the history of Apple stock along with the emotion of financial news headlines to forecast the trend of the aforementioned stock. Vargaset al. [41] used news about financial sentiment to predict the S&P index. The data was taken from Reuters and Bloomberg, two extremely well-known financial journals, and merged with textual data from the financial time series. Other researchers recommended mixing posts on social media and financial news for the textual data. For example, Zhang et al. [48] used posts taken from the Chinese trading social media platform Xueqiu to extract user sentiments and financial data taken from websites in China to extract events.

Creating a model that can predict stock market movement based on tweets or financial news is exciting but challenging. Long et al. [26] limited their analysis to the SZ002424 stock's financial news. They constructed a model to predict stock price with 1-, 2-, and 3-day lags. To examine the data's structure, they utilized a semantic and structural Kernel based on SVM. The model's prediction of the price changes was 73% accurate. Using postings from Stocktwits alone, Xu and Keelj [46] developed a model for forecasting up and down change. To calculate the sentiment score, they employed an SVM and a manually annotated corpus. They were 58.9% accurate in anticipating the movement of the stocks (up or down). Although this model does an excellent job of forecasting whether it will move up or down, it is less useful than the prior study because it cannot forecast stock prices.

Both studies employed SVM for prediction and had unsatisfactory results; other researchers adopted a deep learning strategy which produced outcomes superior to those of conventional machine learning techniques. To model both short- and long-term influences of events on stock price fluctuations, Ding et al. [10] proposed a deep convolutional neural network. The results showed that their neural network exhibited about 6% improvement in stock market prediction compared to the usual baseline approaches. They retrieved the events from the text and trained the neural network using a novel neural tensor network. Wehrmann et al. [43] also applied a deep convolutional neural network to forecast stock market movement simply based on emotion on Twitter. They put forth a method for sentiment analysis that is language-independent and translation-free, processing tweets to identify their sentiment in any language without the necessity for translation. The outcomes demonstrated that this strategy outperformed the SVM model. Although employing only textual information for stock market prediction produced respectable results, adding numerical elements, such as the change in stock price, would greatly increase the predictions' accuracy.

SVM is the most popular algorithm used in conventional machine learning methods for stock market forecasting. Both Xie and Jiang [45] and Li et al. [24] suggested a model that forecasts the stock market based on the sentiment of financial news and historical stock price data using SVM, and despite employing various sentiment dictionaries, their prediction accuracy results were extremely comparable, with Xie and Jiang [45] having a prediction accuracy of 59.17% and Lie et al. having a prediction accuracy of 54.6%. Using social media messages as textual data instead of personal information, Nguyen et al.[29] similarly developed a prediction model using SVM and achieved comparable results in 54.41%. From the earlier research, it can be seen that SVM produces findings that are quite comparable regardless of the type of textual data or the sentiment analysis employed. In addition to SVM, other researchers also employed various conventional machine learning methods. On Twitter and Indonesian company price data, Naive Bayes classifier and Random Forest were used by Cakra and Trisedya [4]; their respective prediction accuracy rates were 67.37% and 66.37%. Using a Decision tree to model Twitter posts and four S&P500 businesses, Vu et al. [42] achieved excellent results with an accuracy of 82.93%. A model developed by Bing et al. in 2014 makes movement predictions for 30 NASDAQ and New York Stock Exchange companies. The proposed model produced an overall accuracy of 76.12% when the authors applied association rule learning to classify Twitter tweets using natural language processing.

Martin [27] developed an 80% direction accuracy using a straightforward feed-forward neural network to examine historical stock market prices and tweets for the CAC40 index. Martin was the first to combine textual and numerical data with deep learning to employ neural networks for stock market prediction. Other researchers who also utilized a conventional feed-forward neural network include Picasso, Merello, Ma, Oneto, and Cambria [31]. The lexicon of Loughran and McDonald, a sentiment dictionary with pre-labeled terms, and the affective space dictionary were used by the authors to first extract the sentiment using two feature selection approaches. The neural network model created by Loughran and McDonald demonstrated higher prediction accuracy compared to the affective space dictionary model.

Long short-term memory is the most used deep learning method for stock market forecasting. Li et al. [23] used forum postings and CSI300 index data in their study to anticipate market movements by the LSTM approach. The authors employed the Naive Bayes algorithm to analyze the textual data, manually labeled the sentiments by experts, and then combined the sentiment with the index data using LSTM, which had a merged layer, a ReLU layer, and a softmax layer. This model produced an 87.86% directional accuracy.In their investigation of the coronavirus's favorable and unfavorable effects on stocks in important stock sectors, Jabeen et al. [19] extracted sentiment scores from tweets about the virus and used them to build an LSTM model that can forecast changes in the stock market price. Another LSTM user is Liu [25], who employed a different strategy than the two studies previously mentioned; he suggested an attention-based LSTM (AT-LSTM) to forecast S&P500 stock data. The focus is split into two groups: the first group gives weight to news stories with a positive tone, while the second group concentrates on giving weight to stories involving S&P500 firms. This approach had a 66% accuracy rate, demonstrating how well the attention-based model forecasts the stock market. Karlemstrand and Leckstrom [21] conducted another study that employed LSTM; the tweet sentiment analysis was extracted using the VADER sentiment library, and the study's goal was to determine whether or not including additional attributes, such as the number of favorites, retweets, and followers, would improve the model's accuracy. Their findings indicated that this was the case. Instead of employing long short-term memory, Ho and Huang [48] employed a convolutional neural network. They began by collecting sentiment scores from tweets using the VADER sentiment library, and then they used a one-dimensional convolutional neural network to predict five important high-demand stocks.

The COVID-19 epidemic had a tremendous influence on the entire world, and its impacts on financial markets have drawn a lot of attention. They look into how news about COVID-19 affects financial markets in study [7]. They use a BERT model that has been modified for the financial market to extract the sentiment of the news by looking at a sizable dataset of 203,886 web articles from reliable sources including MarketWatch, the New York Times, and Reuters. These findings imply that market expectations have been influenced by the flow of COVID-19 news. In the study [9], they use a large amount of data from microblogging websites like Twitter to forecast stock values based on public opinion. The dynamics of the stock market are significantly influenced by the sentiment analysis of user opinions and feelings. This research provides insights into several approaches, their benefits, and drawbacks for stock market prediction by a thorough assessment of current studies. The findings show how crucial sentiment analysis is to accurately forecast changes in stock prices, which leads to greater profitability. This study demonstrates the capability to forecast stock costs with a high level of accuracy using contemporary machine learning and deep learning approaches. Additionally, it highlights the most recent and efficient forecasting methodologies and looks at how stock expectations have changed over time.

Accurate prediction is difficult due to the noise and volatility inherently included in stock market data. In the study [49], they offer SA-DLSTM model which combines an emotion-enhanced convolutional neural network (ECNN), a denoising autoencoder (DAE), and an LSTM model, is a novel hybrid model that addresses this issue. To supplement stock market data, the model takes advantage of user-generated comments from the web and extracts sentiment using ECNN. By employing DAE to extract important elements from stock market data, it further improves prediction accuracy. In order to create more trustworthy sentiment indexes, the model also takes into account the timely influence of emotions on the stock market. According to experimental findings, SA-DLSTM beats other models in terms of prediction accuracy and also performs well in terms of return and risk metrics. The study offers beneficial information to investors. The research [28], which uses deep learning approaches to optimize stock market prediction, produces outstanding outcomes. The CNN model's accuracy was 98.92%, which was greater than the ANN model's accuracy of 97.66%. The CNN model introduced a fresh technique to analysis by using 2-D histograms produced from quantized samples within certain time frames. A case study involving the recent COVID-19 epidemic, which had a considerable impact on the stock market, is also included in the paper. This study's findings revealed a decent accuracy rate of 91%.

Another study [20] explores the predictive power of sentiments during the COVID-19 pandemic, health impacts, and macroeconomic indicators on stock indices. Using long-short-term memory (LSTM) networks, the research focuses on predicting the influence of COVID-19-induced sentiments on stock values in the United States and India. A comparison is made with traditional time series statistical models like autoregressive moving averages and linear regression models, revealing that the LSTM machine learning model outperforms in terms of accuracy. The performance of these models across sectors and countries is analyzed to draw meaningful economic insights.

The goal of the other study [35] is to combine model predictions from textual corporate disclosures with trading strategies while taking into account practical elements like transaction costs, order clearance times, post-publication returns, and liquidity filtering. The suggested trading techniques show annualized returns of up to 7.81% and 9.34% out-of-sample when analyzing a dataset of 354,992 form 8-K filings and 10,204 ad hoc announcements. The results emphasize the significance of training machine learning models on the ternary prediction problem and giving them extra data on earlier disclosures to forecast neutral market reactions. The impact of transaction costs, ensemble sizes, return neutrality thresholds, and liquidity filtering on profitability is investigated using sensitivity analysis. Another study [16] looks at how news sentiment—which includes both general and ESG-related news—predicts stock returns in European markets. Tone, polarity, and activity density, three sentiment indices derived from GDELT-supplied news, are discovered to display strong connections with stock price returns during seven years from 2015 to 2022. Even in a straightforward manner, these connections can be used as leverage to develop effective trading methods that exceed the market. Additionally, these sentiment indicators can be used as inputs for more complex machine learning algorithms which may result in trading methods that are even more successful. Especially encouraging results are shown with indicators derived from ESG-related news, whether used arbitrarily or as inputs for machine learning algorithms. In the study [36], they propose a Lexicon Enhanced Collaborative Network (LECN) for targeted sentiment analysis (TSA) which responds to the need for fine-grained sentiment analysis in financial documents. To capture the many sentiment polarity of particular targets (such as company entities) within sentences, LECN adopts a target-level perspective as opposed to earlier studies that mainly focused on coarse-grained sentiment analysis. The model offers a message selective-passing mechanism to improve collaborative effects regulate information flow between tasks and incorporate sentiment lexicons to facilitate sentiment classification. With increases ranging from 1.47 percentage points to 1.94 percentage points, experimental results on a variety of financial datasets show that LECN beats state-of-the-art baselines in terms of F1-score. The ability of LECN to comprehend domain-specific expressions and achieve favorable interactions between tasks is further revealed by further research.

3. Methodology

3.1. Background of the study

3.1.1. Natural language processing

Understanding spoken and written human language is the main goal of the artificial intelligence field of natural language processing (NLP). To achieve this, specific computer programs are developed. NLP, according to Shruthi and Swamy [38], is a technique that helps a computer by simulating a human's ability to understand language. The interface between human language and computer systems is known as NLP. Only structured, clear programming languages may be used to communicate with a computer verbally. Standard computers cannot understand human-spoken languages, notwithstanding how hazy and unclear they are. As a result, algorithms that can analyze and structure words and then build a computer-understandable language based on those structures are required for software to identify spoken or written language.

The three main processes of natural language processing, according to Fabien [12], are as follows:

- Pre-processing. This step standardizes textual data so that it can be used more readily.
- To represent text as a vector in this stage, you can use the bag of words or term frequency-inverse document frequency (Tf-IdF) approaches. Deep learning can also be used to learn vector representations (embedding).
- Classification, to identify the most similar phrases.

Many pre-built, trained sentiment dictionaries with good text sentiment representation accuracy already use NLP. The five general sentiment dictionaries TextBlob, Flair, Vader, Loughran & McDonald, and Harvard IV-4 will be investigated and contrasted in this study.

An overview of each sentiment package will be provided first, with TextBlob, a Python library used to evaluate textual data, serving as the starting point. This package includes several NLP features such as sentiment analysis, part-of-speech tagging, translation, tokenization, classification, and many others. TextBlob was created on top of NLTK, a different Python-based NLP toolset. TextBlob assigns a score to each word using a pre-defined, human-annotated sentiment vocabulary and then computes the score for the entire phrase using a weighted average. The output consists of three scores: intensity, subjectivity (objective or subjective), and polarity (positive or negative). The polarity score is equivalent to the polarity score and is expressed in the range [–1 1], with -1 being the most adverse and 1 being the most favorable. Another Python library created and made available for use by the public by Zalando Research is Flair. The distinguishing feature of the embedding-based Flair technique is that it may combine many word embeddings with itself, including GloVe, BERT, ELMo, and character embeddings. Another built-in model in Flair is called TextClassifier; it uses word embedding and RNN to create a textual representation. The next step is to predict the sentiment of the text to determine the sentiment score, which has a range from 0 to 1.

As rule-based libraries that employ a pre-defined and human-annotated sentiment lexicon for each word, Vader and TextBlob are quite similar to one another. The fact that VADER was created and trained primarily for social media, however, makes a difference and suggests that VADER places a high weight on rules that reflect the spirit of language frequently used on social media. A compound score with a range of [–1 1] is returned by VADER as positive, negative, and neutral intensity scores that are then normalized. Another natural language processing library, Loughran and McDonald, was developed using the most probable interpretation of a term in an economic and business context. There are various word lists in this library (negative, positive, uncertainty, litigious, strong modal, and weak modal). A Python module called pysentiment2 was employed for this study. The LM function in this library uses Loughran and McDonald's terms list to categorize sentiment. After tokenizing the text, this function provides a score in a range of [–1 1].

The Harvard IV-4 psychological lexicon offered by General Inquirer Software is divided into a number of word categories such as positive, negative, affiliation, antagonism, power, submission, active, and passive. The Loughran and McDonald library was replaced with the pysentiment2 Python library, which also has a function called HIV4 that uses the Harvard IV-4 psychological vocabulary for sentiment categorization. This function tokenizes the text and outputs a score with a [–1 1] range.

3.1.2. Long short-term memory

Hochreiter and Schmidhuber [18] define long short-term memory as a recurrent neural network coupled with an appropriate gradient-based learning approach. To handle error back-flow issues, the LSTM method was developed. It was made to address the vanishing and exploding gradient issue with a regular recurrent neural network. The only difference between the LSTM design and the Vanilla recurrent neural network is that the LSTM cell is present in place of the hidden units. The architecture of long short-term memory is depicted in Figure 1.

Each LSTM cell also keeps track of a cell state vector in addition to the hidden state vector. Using an explicit gating mechanism, each unit has three gates with the same form, and the LSTM can select whether to read from, write to, or reset the cell at any time step.

The three gates shown in Figure 2 inside the cell are:

- the input equation for the input gate, which determines whether the memory cell is updated

$$i^{(t)} = 2^2\sigma(W^i[h^{(t-1)}, x^{(t)}] + b^i)$$

- the forget equation which determines whether the memory is set to 0

$$f^{(t)} = \sigma(W^f[h^{(t-1)}, x^{(t)}] + b^f)$$

- output gate determines whether the information about the state of the current cell is made visible

$$o^{(t)} = \sigma(W^o[h^{(t-1)}, x^{(t)}] + b^o)$$

The model is still differentiable despite the smooth curves shared sigmoid activation ranging from 0 to 1.

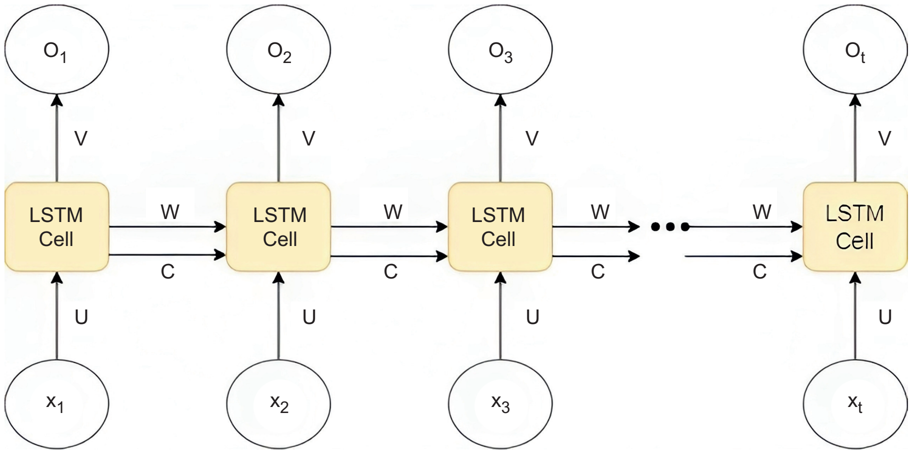

Figure 1. Long short-term memory architecture [15]

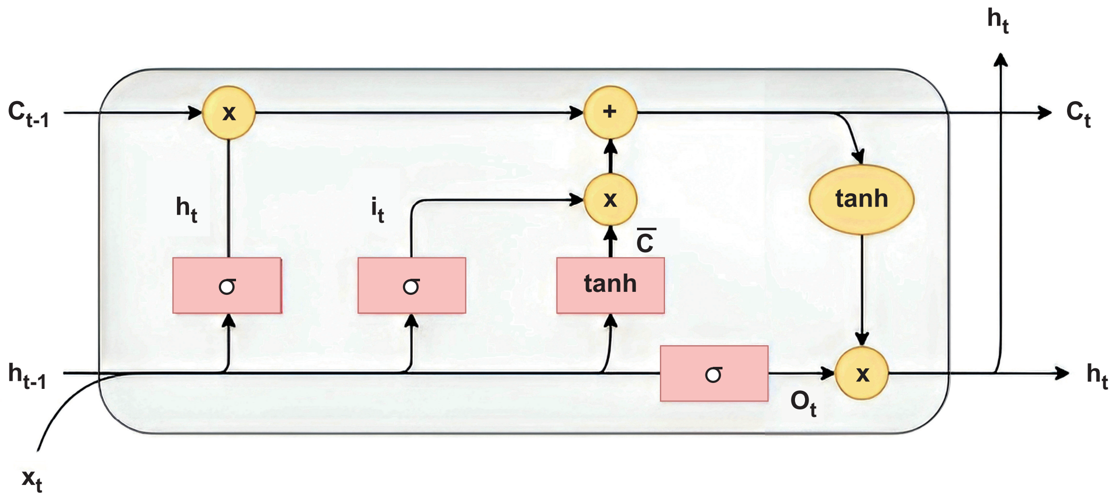

Figure 2. Diagram of an LSTM cell showing the flow of information through its gates and cell states [15]

Another vector C that alters the cell state is present in addition to these gates. This vector possesses tanh activation which makes it possible for cell state information to flow for a longer time without dissipating or exploding. The following is the equation for cell input activation:

$$\overline{C}^{(t)} = \tanh(W^c[h^{(t-1)}, x^{(t)}] + b^c)$$

Each gate accepts the hidden state and the current input x as inputs before concatenating the vectors and applying a sigmoid. $\overline{C}$ stands for a brand-new potential value that could be used to change the state of the cell.

As was previously said, the input gate determines whether a memory cell is updated, therefore it is applied to $\overline{C}$, the sole vector that can change the state of the cell. How much of the previous state $C(t-1)$ to be forgotten is decided by the forget gate $f(t)$. The following is the cell state equation:

$$C^{(t)} = f^{(t)}C^{(t-1)} + i^{(t)}\overline{C}^{(t)}$$

where h^t is the hidden state vector, which is obtained by applying this state to the output gate. This state's equation is as follows:

$$h^{(t)} = \tanh(C^{(t)})0^{(t)}$$

3.1.3. Convolutional neural network

The only method for processing data with grid-like topologies is a convolutional neural network such as one-dimensional time series data, which is a grid of samples across space, or two-dimensional picture data, which is a grid of pixels in space. The first and most significant layer in a convolutional neural network—which typically contains four different types—is the convolution layer. This subcase uses filters or kernels, which are tiny units that are applied over the data using a sliding window, to convolve the data. The element-wise product of filters is used in this operation, along with casting those values for each sliding movement.

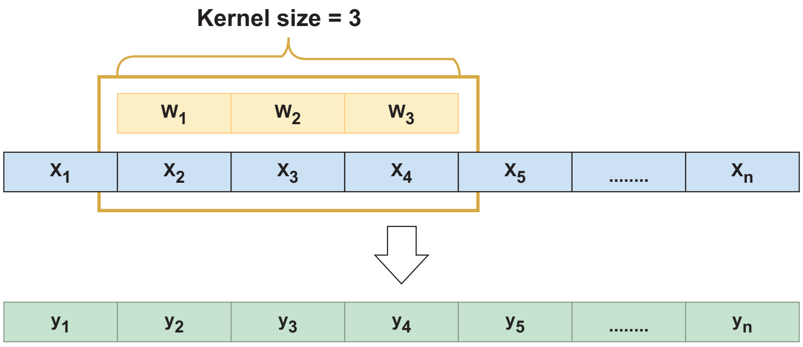

Figure 3. One-dimensional convolution operation (Ragab et al. [33])

Figure 3 shows how convolving a one-dimensional time series data works. In this example, the kernel size will be set to 3 with the weights being (w_1, w_2, w_3). To obtain the feature map values meaning the output y, the values from the input layer are multiplied by the weights and summed up. For example, to obtain y_3 the following equation is used

$$y_3 = w_1x_2 + w_2x_3 + w_3x_4$$

The activation layer is a further layer of the convolutional neural network. Following the convolution layer, a non-linear activation function is utilized to produce an activation map as an output by taking the feature map produced by the convolution layer as input. Most frequently, leaky reLU and reLU are utilized as activation functions.

The pooling layer, which downsamples the data so that fewer parameters need to be learned during training, is another key layer in the CNN. The spatial extent dimension, or, to put it another way, the value of n for which a representation of $n \times n$ features can be taken and mapped to a single value, and the stride, or, to put it another way, the number of features the sliding window skips, are commonly introduced by the pooling layer. Because there are fewer parameters, pooling reduces the risk of overfitting.

The output of the convolutional layers, which may also be flattened and linked to the output layer, represents high-level features in the data. Convolutional layers successfully produce a low-dimensional, largely invariant feature space; adding a fully linked layer is often a quick way to learn non-linear combinations of these layers. The fully linked layer is learning in that space a potentially non-linear function.

3.1.4. Coefficient of determination

The coefficient of determination R^2 serves as a statistic that explains the amount of variance accounted for in the link between two (or more) variables [34]. The formula for this statistic is as follows:

$$R^2 = \frac{RSS}{TSS}$$

The R^2 score value falls between $[0, 1]$, where RSS stands for the sum of squares of residuals and TSS for the total sum of squares. In supervised machine learning issues, the accuracy of a model is typically assessed using R squared. One being the most precise and 0 being the least.

3.2. Data collection

Two types of data will be used in this study: historical 2015 stock price data for five distinct stocks and Twitter data that contains various tweets connected to the topic at hand. A dataset developed by Dogan et al. [11] and released on the website Kaggle for their study on speculator and influencer evaluation in the stock market by using Social media was used for the Twitter data. The yfinance package on Python was utilized for the stock data; this package enables users to obtain historical stock market data from Yahoo! Finance using a ticker model; for instance, the ticker for Apple stock is AAPL.

Selenium was used to obtain the tweets. Dogan et al. [11] used several keywords to find the tweets; in the case of the Apple stock, for example, they used tags (apples), stocks (AAPL), and phrases (apple). The same was carried out with the equities of Microsoft, Tesla, Amazon, and Google Inc. The Dataset contains information about each tweet, including its tweet ID, author, posting date, text content, and the number of comments, likes, and retweets. It spans five years, from 2015 to 2020, and contains more than 3 million tweets. Each tweet includes a unique ID and a ticker symbol, making it easier to use and distinguish it from other tweets (AAPL, GOOG, AMZN, MSFT, and TSLA). We will only discuss statistics from 2015 in this article.

For the historical stock data, the history function of the yfinance package was used to retrieve the historical stock prices for each working day of 2015. The following columns are present in the data:

- Date. The date of the business day.
- Open the stock price at the opening of the business day.
- Close. The stock adjusted for splits at the business day's closing.
- High. The highest stock price that day.

- Low. The lowest stock price that day.
- Volume. The volume of the trades that day.
- Dividends. The amount paid to shareholders.

3.2.1. Data analysis

The Twitter data being used is unprocessed and noisy. The data needs to be thoroughly cleaned and pre-processed because it contains words and symbols that do not fully provide meaningful information, such as hashtags, retweet tags, and links, which the natural language processing libraries being utilized can not interpret well. Before cleaning the tweets, all rows with null values or duplicate tweets must be removed.

The tweets are then cleansed. To begin, all characters are first converted to lowercase. Regular expressions, a Python package, substitutes undesired words and symbols with more evocative language. Since all Twitter usernames begin with the @ sign, all words beginning with that symbol are converted to the word USER using the function re. In the same way, any word beginning with "https?://" is changed to URL for links. The hashtag, punctuation, and dollar signs that are often used before the name of stocks are then deleted for greater clarity. Pronouns, coordinating conjunctions, and auxiliary verbs are a few examples of words that are typically filtered out before processing natural languages and comprise stopwords. For defining the stopwords that were to be dropped, the natural language toolkit (NLTK) was employed. Since the stock market is closed on weekends and holidays, all of the tweets must be deleted after they have been cleaned. The data was changed to only include dates in this library; for this, the library Bday, a Python library that contains all the business days of the year, was utilized.

3.2.2. Applying sentiment analysis

Five various sentiment analysis libraries will be examined, as was previously described, and the best one to utilize for predicting the stock market price will be selected.

A sample of the data was made, and the sample solely consisted of January. The Apple stock was chosen as the experiment's test stock using the ticker symbol column. The data was then subjected to the five sentiment libraries after the sample had been created.

Using a weighted average to get the score for the entire sentence, TextBlob uses a pre-defined sentiment lexicon that provides a score for each word. The output is then provided as three scores. The TextBlob Python module is used in this instance to analyze each tweet and forecast the results. Only the polarity scores provided in the [–1, 1] interval are taken, and a new column called sentiment is added to the data. The same procedure used with TextBlob is followed, and VADER forecasts the positive, negative, neutral, and compound scores. Only the compound, which combines the other three values into a single score, will be used, and the data will be placed in a new column called sentiment. Similar to how it was done for the first two, each tweet's sentiment is predicted using the NLP library flair. Flair, however, returns the score in a [0, 1] range along with a positive or negative value. The tweet is multiplied by –1 if it is classified as negative, and it is left alone if it is positive. In the new column sentiment, all of the final results are kept.

The Python library sentiment pysentiment2 is used to predict the sentiment score of each tweet. Its built-in function named LM was used to tokenize the textual data and return a score of an interval [–1 1], which is then added to the data in a column named sentiment similar to what was done before. The identical procedure used for Loughran and McDonald was applied to Harvard IV-4, and the pysentiment2 Python library was used. However, HIV4 was used as opposed to the built-in function LM.

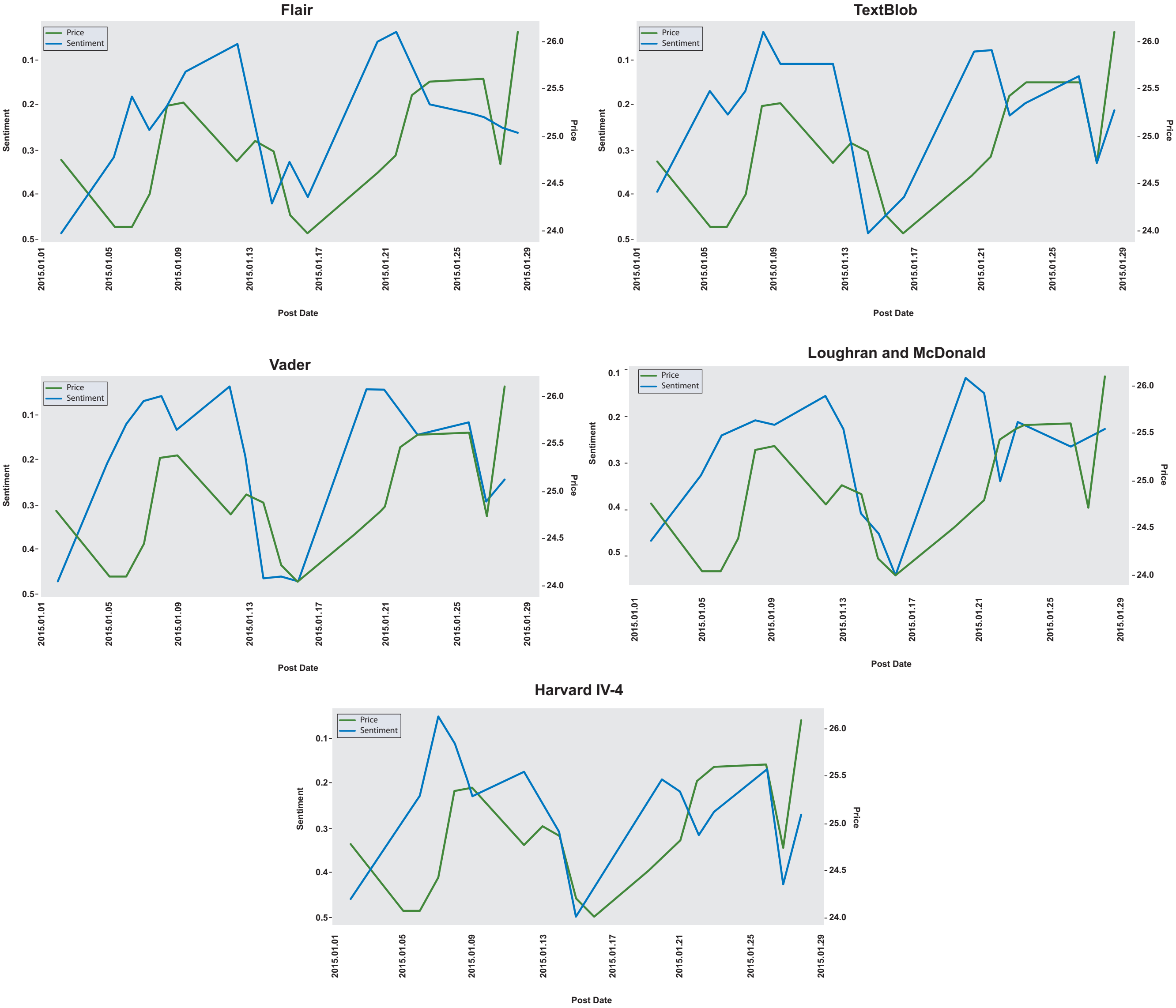

Figure 4. The plots of the change of the sentiment scores with the change of the stock market price over January

It is necessary to compare the change in the sentiment score of the tweets mentioning the Apple stock over January 2015 with the change in the actual Apple stock for each library to determine which one of the five deployed libraries is performing the best. For each day of the month, one sentiment score is obtained by first taking all the tweets from that day and averaging their sentiment scores. The data from the sentiment scores is then combined with the Apple stock price data from the yfinance Python package that was previously discussed.

The correlation between the stock market price data and the sentiment scores is required now that the datasets are complete. Corr, a Python data frame built-in function, is used for this purpose; it returns a pairwise correlation of columns. When the function is used, the outcomes shown in Table 1 are produced. Additionally, the following graphs are produced by plotting the change in sentiment ratings over January against the change in stock market price.

Table 1. Correlation results between stock market price and sentiment score

Natural language processing library	Correlation results
TextBlob	0.312
Vader	0.28
Flair	0.253
Loughran & McDonald	0.305
Harvard IV-4	0.265

Table 1 shows that TextBlob and Loughran & McDonald have correlations higher than 30% between the change in sentiment score and the change in stock price, making them the two NLP libraries with the strongest correlations. Figure 4 confirms the correlation findings since it is clear that, in contrast to the other three, TextBlob and Loughran & McDonald's change in sentiment score closely mirrors changes in stock price. These findings show that TextBlob and Loughran & McDonald are the most effective sentiment libraries for the investigation. Since TextBlob produced the best results of the two, it will be the only one used.

3.3. Research design

The goal of the research is to develop a deep learning model that properly forecasts stock market movement using textual data, including historical stock price time-series data and tweets regarding certain stocks. The tweets must undergo sentiment analysis to be used in the study's models. As a result, as indicated in Figure 5, the research will be divided into two sections: pre-processing, which involves cleaning the data and applying sentiment analysis; and training two deep learning models (LSTM and 1D-CNN) on the combined data (Twitter data and historical stock price data).

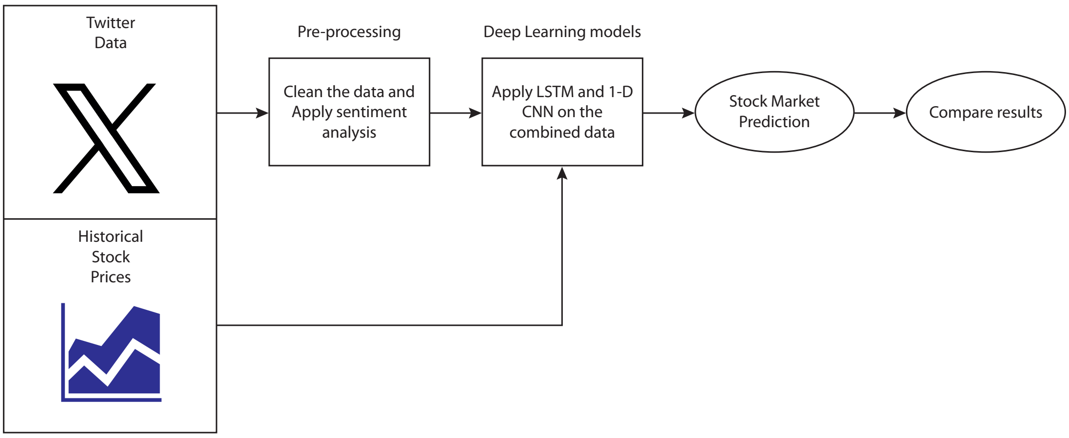

Figure 5. Research Framework

3.3.1. Developing the models

The Loughran & McDonald library and TextBlob were found to be the finest natural language processing libraries for the research based on the sentiment analysis performed; as a result, these libraries were utilized to obtain the sentiment score for each tweet of the 2015 data. For the five different stocks that were tested in this study, five distinct datasets were produced for the modes. The stock that each tweet refers to is indicated in a column in the emotion scores data called the ticker symbol. The sentiment data is divided into five datasets including tweets regarding the five distinct equities using the ticker symbol column. To use the models, the sentiment score must then be converted into time-series data. To do this, all of the tweets from a particular day are taken for each dataset and their sentiment ratings are averaged, yielding one sentiment score for each day of the year. The historical stock price data is then combined with the sentiment score data using the yfinance Python package, which was previously detailed.

Setting up each dataset for the LSTM is the initial step. The datasets will need to be normalized and framed as a supervised learning problem to be ready. Using a MinMaxScaler, which first divides by the range and then subtracts the minimal value, the features are first normalized.

Given the sentiment score and stock price from the previous day, the supervised learning task will be phrased as a prediction of the stock price on the current day. To accomplish this, the Dataset is changed using a function that adds a new column with the stock price data but with a one-day lag. subsequently, all pre-processing was applied.

The prepared data is divided into two sets: a train set that has 172 rows and a test set that contains 74 rows. The sentiment score column and the stock price from the previous day are then divided into input and output, with the newly generated column containing the stock price but with a one-day lag being the output. Because LSTM anticipates inputs in that shape, the inputs are molded into a 3D structure similar to the following [samples, timesteps, features].

To predict the stock price, the LSTM is defined with 230 neurons in the first hidden layer and one in the output layer. One time step with two features will make up the input shape. A straightforward formula that is typically employed for less complicated problems was utilized to determine the number of neurons:

$$N_h = \frac{2(N_i + N_0)}{3}$$

where N_i is the number of input neurons, N_0 is the number of output neurons, and N_h is the number of neurons in the first hidden layers.

30% dropout was applied as a layer. By ignoring randomly chosen neurons during training and lowering the sensitivity to the particular weights of individual neurons, the dropout layer guards against overfitting. Three dense layers were added after the dropout layer; the first one reduced the dimension from 230 to 50 and included the activation function ReLu. The second shrinks the dimension from 50 to 3 and includes a weight initializer called HeNormal that addresses the disappearing and exploding gradient descent issues. The output from the previous Dense Layer will be used by the final Dense Layer, which will then multiply by a matrix and a vector to produce an output of size 1.

Wilson et al. [44] demonstrated that the efficient Adam version of stochastic gradient descent performs well for LSTM, hence this version was chosen for the optimizer. This model was evaluated using the mean absolute percentage error. Finally, a batch size of 72 will allow the model to fit 110 training epochs. The number of epochs and the batch size were discovered using a basic grid search.

The datasets will be framed as a supervised learning problem for CNN, and one way to achieve this is to structure the problem as a prediction of the stock price on a particular day given the sentiment score and stock price on the previous three days. To produce a new column with the stock price data but with a one-day lag, the same function series to supervise used for LSTM will be utilized. The data must then be organized into samples for the input and output. The three days before the current day are used because the 1D CNN model needs enough information to learn the mapping from an input sequence to an output value. To retain the order of observations across the two input sequences, the data must be divided into samples. The data must be divided into sequences, with each sequence containing the input data from the three previous days and the output is just the stock price of the current day, to be able to predict the stock price of a given day using the stock price and sentiment score of the three previous days.

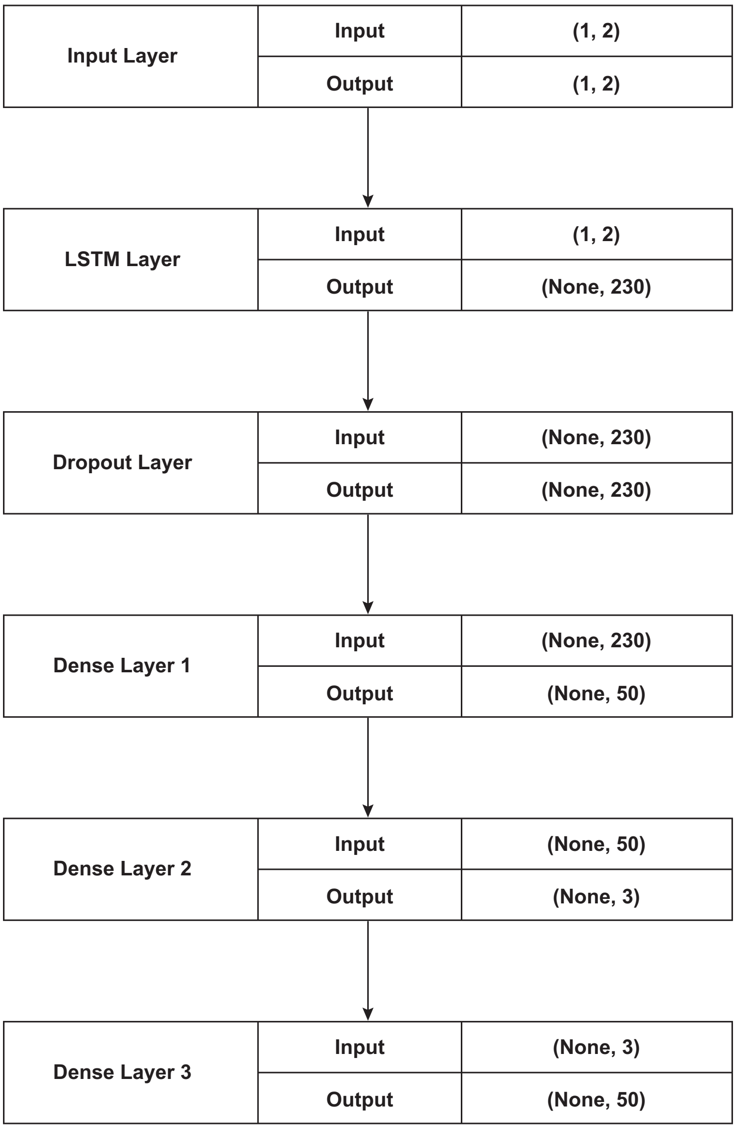

Figure 6. A plot of the LSTM model

The data is initially divided into train and test data sets, with the training set consisting of 174 rows and the test set containing 72 rows, just like with LSTM. After that, the model is defined by building a sequential model and including layers. The first layer is a one-dimensional convolution layer, with a kernel size of 2 and 64 neurons serving as its filters. A straightforward grid search was used to determine the number of filters. Because Chazareix defines the kernel size as the product of the number of outputs and the number of inputs, the kernel size was set to two. The kernel size should be 2, as there are only two inputs (sentiment score and stock price of the prior day) and one output (stock price of the present day). Additionally included was the ReLu activation function, which takes the feature map produced by the convolution layer as input and produces an activation map as the result.

The next step was to add a pooling layer, which is used for downsampling. The pooling size and stride were both set to 2, which implies that each feature map will be cut in half by the pooling layer. There is now a flattened layer that reduces the pooling layer's multidimensional input to a single dimension. Finally, a dense layer was built with a unit of 1. All of the neurons in the Flatten layer will input into each of these layer neurons, which will then execute a matrix-vector multiplication and produce an output of size 1. The study's LSTM model's layer architecture is depicted in Figure 7.

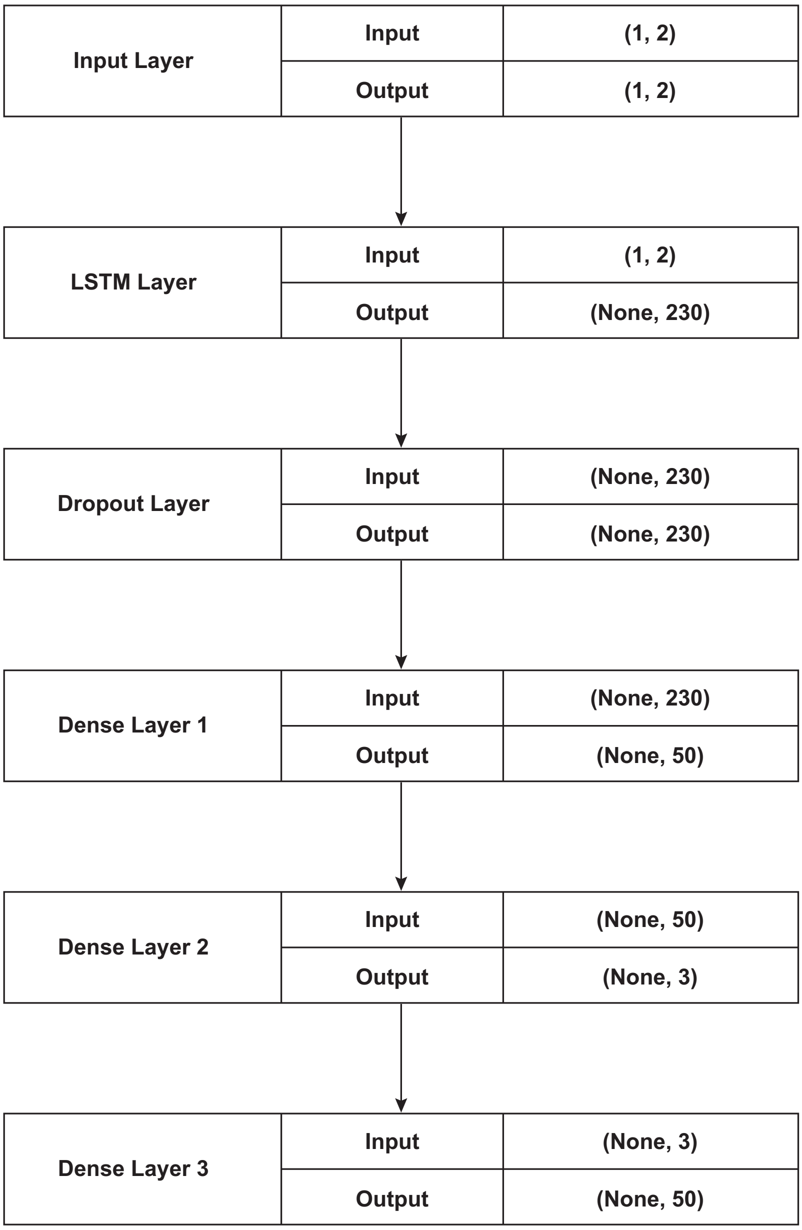

Figure 7. A scheme of the LSTM model

It is important to note that this study has several restrictions. The algorithm might not perform well for fewer well-known equities because there won't be enough Twitter data. After all, individuals are less likely to tweet or discuss these stocks. Because this study only takes into account tweets in English, it may not be able to accurately portray the sentiment of the general audience. Additionally, the models only employ time-series data with a one-day increment, which may not be optimal for businesses and traders since the models only estimate the closing stock price for the following day using sentiment and stock price from the previous days.

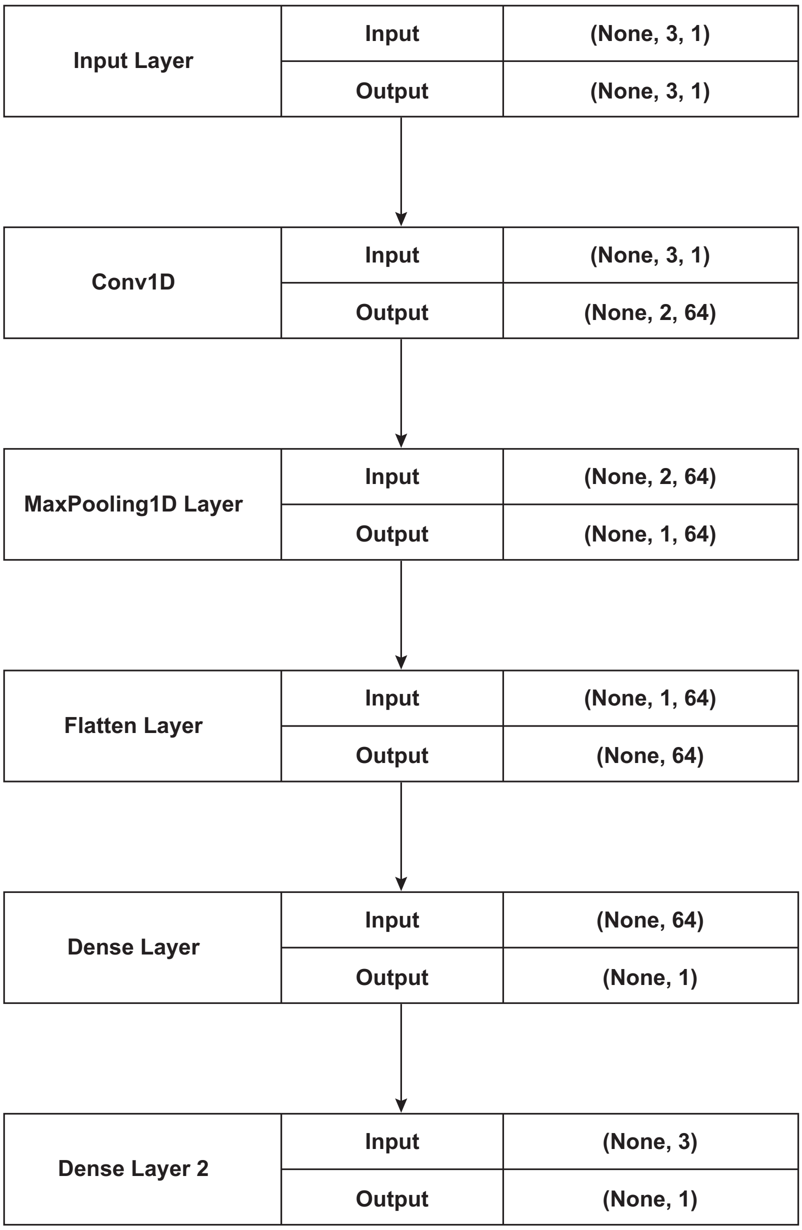

Figure 8. The scheme of the developed 1D CNN model

3.4. Results

Our models may be tested on the five different stocks, Apple, Amazon, Google, Microsoft, and Tesla. Both 1D CNN and LSTM models have been defined and trained. Each stock's change in stock price is predicted using the models, and the predicted values are then contrasted with the actual values. The coefficient of determination, R^2 is used to calculate the accuracy results. It provides the accuracy of the predicted values compared to the actual values as a number between 0 and 1, and the result is then multiplied by 100 (Table 2).

Table 2. R^2 for LSTM and 1D CNN for each stock and average R^2 [%]

Model	LSTM (%)	1D CNN (%)
Stocks Apple	82.3	74.9
Amazon	82.3	94.5
Google	96.0	93.4
Microsoft	95.3	91.8
Tesla	83.9	59.9
Average accuracy	90.7	82.9

The charts for individual stocks were produced after plotting the anticipated stock prices and the actual values for each stock vs. the number of days in the test data set.

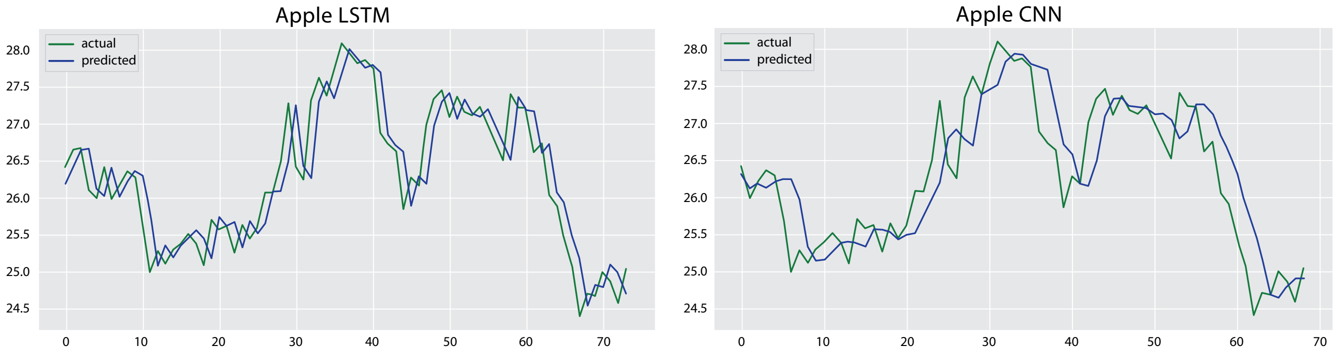

Figure 9. The plots of the LSTM and CNN model prediction and the actual Apple stock market price change

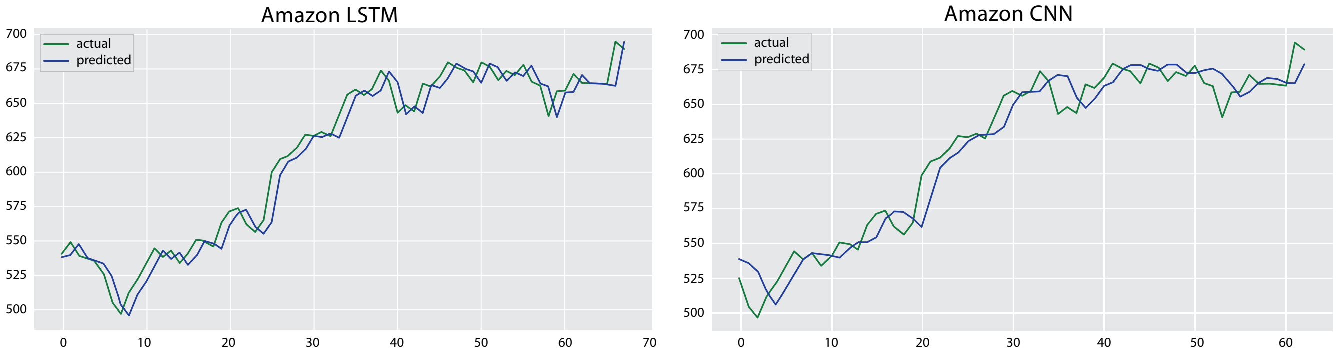

Figure 10. The plots of the LSTM and CNN model prediction and the actual Amazon stock market price change

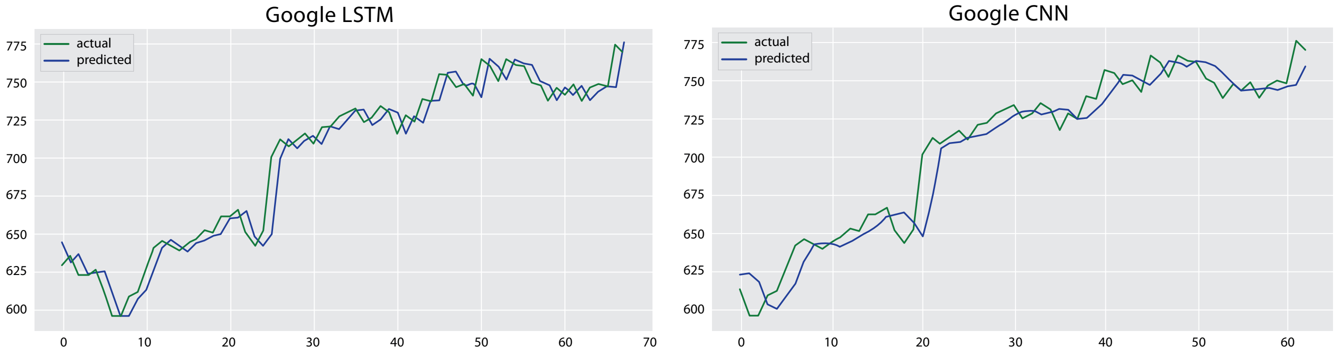

Figure 11. The plots of the LSTM and CNN model prediction and the actual Google stock market price change

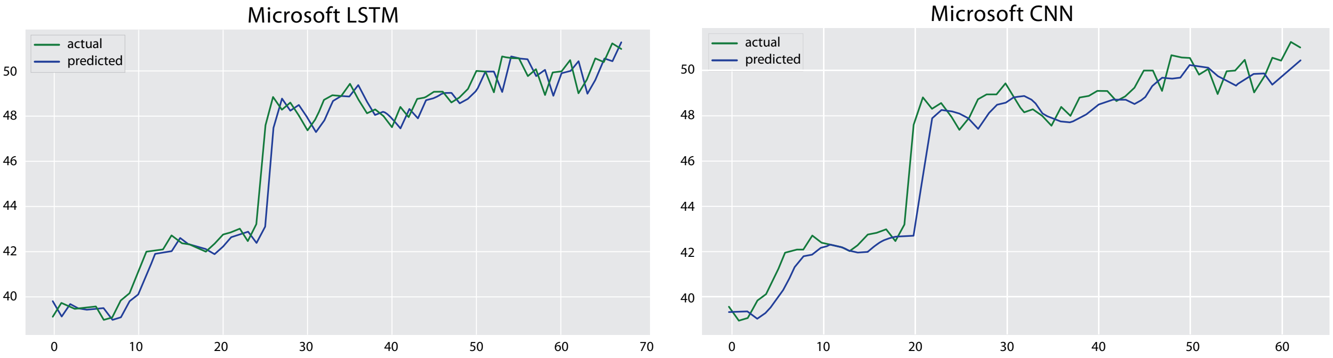

Figure 12. The plots of the LSTM and CNN model prediction and the actual Microsoft stock market price change

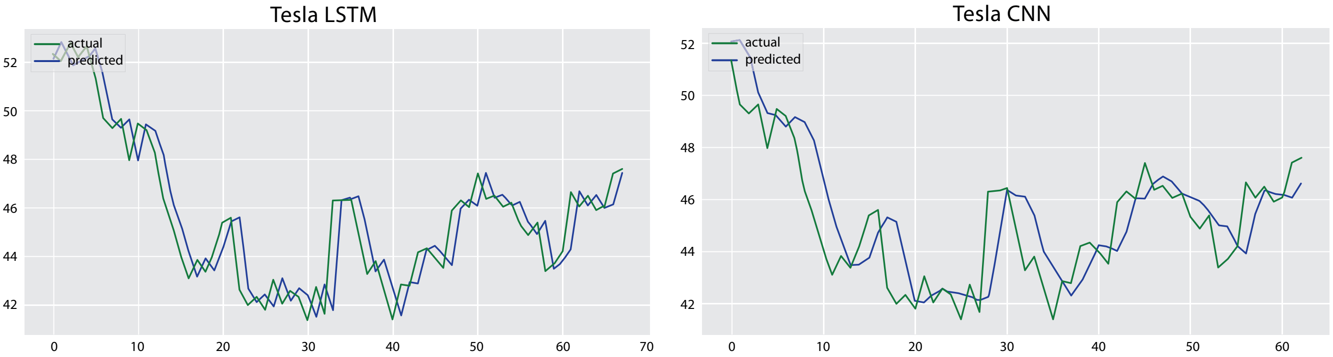

Figure 13. The plots of the LSTM and CNN model prediction and the actual Tesla stock market price change

3.5. Discussion

Which deep learning model has a higher forecasting accuracy is the first question that the study aims to address. The LSTM model outperforms the 1D CNN model. Despite the strong testing results of both models, the LSTM model produced greater R^2 across all of the stocks it was tested on.

The greatest R^2 for the LSTM is 96.1% for Microsoft stock and 96.0% for Amazon, with an average R^2 of 90.7% for both. The shares of Apple and Tesla have the lowest yields, at 82.3% and 83.9%, respectively. The 1D CNN model, in contrast, had an average R^2 of 82.9%. The equities of Amazon and Google had the greatest and lowest R^2 values, respectively, of 94.5% and 93.4%, and 59.9% and 74.9%.

The findings demonstrate that while the R^2 results for Google, Amazon, and Microsoft stock for both LSTM and 1D CNN models all surpass 90%, the LSTM model performs significantly better in both Apple and Tesla stock with an R^2 exceeding 80%. The 1D CNN, in contrast, cannot barely reach 75%. Figures 9 and 12, which show the plot of the actual market price test data for Apple and Tesla, respectively, can be used to demonstrate this. As demonstrated in Figures 10–12, the two stocks are far more volatile than Amazon, Google, and Microsoft, which have a more linear and simple shift in the actual stock price. Since LSTM retains long-term memory and context of the data within its neurons, it can handle sequences with volatile changes better than 1D CNN, which only considers the three previous sequences to predict the next day's price. LSTM also uses forget gates to remove old values since it is unlikely that the subsequent outcome will depend on them. Which of the pre-defined sentiment analysis libraries is best for foretelling changes in stock market price is the second study topic. Since TextBlob and Loughran & McDonald's sentiment shift had the highest correlation with the stock market movement, according to the analysis in Section 3.2, they are the best choices for our task. Since the first sentiment library was developed based on the most probable interpretation of a word in an economic and business context, it was anticipated that Loughran & McDonald and VADER would do the research best. The latter, on the other hand, was created and trained primarily for social media. However, TextBlob had the best performance, while VADER fared badly. It is claimed that this is because tweets and postings on money and stocks are more official than normal social media posts. The outcomes made sense given that TextBlob is more suited for formal language usage, whereas VADER works better with slang, emojis, and simpler phrases used in social media.

3.6. Conclusions

LSTM can handle time-series problems well because it can maintain the context of long-term memory, most research in the literature that used deep learning for forecasting the stock market price change using Twitter sentiment preferred the use of an LSTM model. Our research expands on that and demonstrates that LSTM can handle time-series problems well.

The experiment, according to this study, also sheds new light on the best sentiment library to employ when conducting Twitter sentiment analysis to solve stock market prediction issues even though various studies in the literature employed VADER and Loughran & McDonald under the guise that the former is appropriate for social media posts. The latter, however, is suitable for language used in finance. According to the results, TextBlob performs better than the two libraries mentioned, Flair, Harvard IV-4, and only Loughran & McDonald has results that are almost identical to TextBlob's.

Using Twitter sentiment research to anticipate stock market movement, the results demonstrate that the LSTM is superior to 1D CNN, especially for stocks with dramatic price changes. Additionally, it demonstrates that the TextBlob sentiment library is the most effective choice for sentiment analysis on tweets data connected to stocks.

Since most stock market decisions are made by the minute and waiting until the end of the day to make decisions can be expensive for investors, the two models used in this study LSTM and 1D CNN--predicted the stock market change for the five stocks by each day using the sum of sentiment scores for an entire day. In future research, a model can be created to predict the stock market price change by the hour or minutes instead. After evaluating it against four other NLPs, the preset natural language processing package TextBlob was chosen for the Twitter sentiment analysis in this study.

More natural language libraries can be looked into for Twitter sentiment analysis in future studies, or we can even build our own from scratch. Another suggestion that can be explored is including additional model parameters such as the number of retweets, followers, and comments. These parameters can be used to weight tweets, with tweets that receive more engagement receiving higher weights. This will enable the model to concentrate on tweets with greater significance and weed out spam tweets. Using solely English-language tweets can be restrictive; analyzing the sentiment of tweets in other languages can be highly helpful to our model and assist in predicting local stocks in nations where English is not the primary language.

References

[1] Abu-Taleb, S. K., and Nilsson, F. Impact of social media on investment decision: A quantitative study which considers information online, online community behaviour, and firm image. Bachelor's thesis, Umeå University, 2021.

[2] Baresa, S., Bogdan, S., and Ivanovic, Z. Strategy of stock valuation by fundamental analysis. *UTMS Journal of Economics 4*, 1 (2013), 45–51.

[3] Breaban, A., and Noussair, C. N. Emotional state and market behavior. *Review of Finance 22*, 1 (2018), 279–309.

[4] Cakra, Y. E., and Trisedya, B. D. Stock price prediction using linear regression based on sentiment analysis. In *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (Depok, Indonesia, 2015), IEEE, pp. 147–154.

[5] Catelli, R., Fujita, H., De Pietro, G., and Esposito, M. Deceptive reviews and sentiment polarity: Effective link by exploiting BERT. *Expert Systems with Applications 209* (2022), 118290.

[6] Chauhan, P., Sharma, N., and Sikka, G. The emergence of social media data and sentiment analysis in election prediction. *Journal of Ambient Intelligence and Humanized Computing 12*, 2 (2021), 2601–2627.

[7] Costola, M., Hinz, O., Nofer, M., and Pelizzon, L. Machine learning sentiment analysis, Covid-19 news and stock market reactions. *Research in International Business and Finance 64* (2023), 101881.

[8] Deng, L., and Yu, D. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing 7*, 3–4 (2014), 197–387.

[9] Dhanasekaren, K., Aluri, S. T., Karthikeyan, N., Baskaran, S. H., and Selvanambi, R. A study on the impact of sentiment analysis on stock market prediction. *Recent Advances in Computer Science and Communications 16* (formerly: *Recent Patents on Computer Science*), 1 (2023), 73–93.

[10] Ding, X., Zhang, Y., Liu, T., and Duan, J. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1415–1425.

[11] Doğan, M., Metin, Ö., Tek, E., Yumuşak, S., and Öztoprak, K. Speculator and influencer evaluation in stock market by using social media. In *2020 IEEE International Conference on Big Data (Big Data)* (Atlanta, GA, USA, 2020), IEEE, pp. 4559–4566.

[12] Fabien, M. (3 November 2019). Natural language processing in French (TAL/NLP) (accessed on 29 November 2019) (in French).

[13] Ghanem, D., and Rosvall, D. Major world events impact on stock market prices: An event study. Bachelor's thesis, Department of Business Studies, Uppsala University, 2014.

[14] Gilbert, E., and Karahalios, K. Widespread worry and the stock market. In *Proceedings of the Fourth International AAAI Conference on web and Social Media*, Vol. 4, No. 1 (2010) pp. 58-65.

[15] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*, The MIT Press, 2016.

[16] Goutte, S., Liu, F., Le, H. V., and von Mettenheim, H.-J. (5 January 2023). ESG investing: A sentiment analysis approach (accessed on 28 December 2023).

[17] Hasselgren, B., Chrysoulas, C., Pitropakis, N., and Buchanan, W. J. Using social media & sentiment analysis to make investment decisions. *Future Internet 15*, 1 (2023), 5.

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

[19] Jabeen, A., Afzal, S., Maqsood, M., Mehmood, I., Yasmin, S., Niaz, M. T., and Nam, Y. An LSTM based forecasting for major stock sectors using Covid sentiment. *Computers, Materials and Continua 67*, 1 (2021), 1191–1206.

[20] Jena, P. R., and Majhi, R. Are Twitter sentiments during Covid-19 pandemic a critical determinant to predict stock market movements? A machine learning approach. *Scientific African 19* (2023), e01480.

[21] Karlemstrand, R., and Leckström, E. Using Twitter attribute information to predict stock prices, 2021. Working paper version available from arXiv: https://doi.org/10.48550/arXiv.2105.01402.

[22] Khatri, S. K., and Srivastava, A. Using sentimental analysis in prediction of stock market investment. In *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (Noida, India, 2016), IEEE, pp. 566–569.

[23] Li, J., Bu, H., and Wu, J. Sentiment-aware stock market prediction: A deep learning method. In *International Conference on Service Systems and Service Management* (Dalian, 2017), IEEE, pp. 1–6.

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

[25] Liu, H. Leveraging financial news for stock trend prediction with attention-based recurrent neural network, 2018. Working paper version available from arXiv: https://doi.org/10.48550/arXiv.1811.06173.

[26] Long, W., Song, L., and Tian, Y. A new graphic kernel method of stock price trend prediction based on financial news semantic and structural similarity. *Expert Systems with Applications 118* (2019), 411–424.

[27] Martin, V. Predicting the French stock market using social media analysis. In *2013 8th International Workshop on Semantic and Social Media Adaptation and Personalization* (Bayonne, France, 2013), IEEE, pp. 3–7.

[28] Mukherjee, S., Sadhukhan, B., Sarkar, N., Roy, D., and De, S. Stock market prediction using deep learning algorithms. *CAAI Transactions on Intelligence Technology 8*, 1 (2023), 82–94.

[29] Nguyen, T. H., Shirai, K., and Velcin, J. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications 42*, 24 (2015), 9603–9611.

[30] Pathak, A. R., Pandey, M., and Rautaray, S. Topic-level sentiment analysis of social media data using deep learning. *Applied Soft Computing 108* (2021), 107440.

[31] Picasso, A., Merello, S., Ma, Y., Oneto, L., and Cambria, E. Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications 135* (2019), 60–70.

[32] Porshnev, A., Redkin, I., and Shevchenko, A. Machine learning in prediction of stock market indicators based on historical data and data from Twitter sentiment analysis. In *2013 IEEE 13th International Conference on Data Mining Workshops* (Dallas, TX, USA, 2013), IEEE, pp. 440–444.

[33] Ragab, M. G., Abdulkadir, S. J., Aziz, N., Al-Tashi, Q., Alyousifi, Y., Alhussian, H., and Alqushaibi, A. A novel one-dimensional CNN with exponential adaptive gradients for air pollution index prediction. *Sustainability 12*, 23 (2020), 10090.

[34] Salkind, N. J. *Encyclopedia of Research Design*, vol. 1. Sage, 2010.

[35] Schmitz, H. C., Lutz, B., Wolff, D., and Neumann, D. When machines trade on corporate disclosures: Using text analytics for investment strategies. *Decision Support Systems 165* (2023), 113892.

[36] Shang, L., Xi, H., Hua, J., Tang, H., and Zhou, J. A lexicon enhanced collaborative network for targeted financial sentiment analysis. *Information Processing & Management 60*, 2 (2023), 103187.

[37] Shastri, M., Roy, S., and Mittal, M. Stock price prediction using artificial neural model: an application of big data. *EAI Endorsed Transactions on Scalable Information Systems 6*, 20 (2019), e1.

[38] Shruthi, J., and Swamy, S. A prior case study of natural language processing on different domain. *International Journal of Electrical and Computer Engineering 10*, 5 (2020), 4928–4936.

[39] Singh, T., Bhisikar, S. B., and Kumar, M. Stock market prediction using ensemble learning and sentimental analysis. In *Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021* (Singapore, 2023), R. Doriya, B. Soni, A. Shukla and X.-Z. Gao, Eds., vol. 946 of Lecture Notes in Electrical Engineering, Springer, pp. 429–441.

[40] Turri, A. M., Smith, K. H., and Kemp, E. Developing affective brand commitment through social media. *Journal of Electronic Commerce Research 14*, 3 (2013), 201–214.

[41] Vargas, M. R., de Lima, B. S. L. P., and Evsukoff, A. G. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)* (Annecy, France, 2017), IEEE, pp. 60–65.

[42] Vu, T. T., Chang, S., Ha, Q. T., and Collier, N. An experiment in integrating sentiment features for tech stock prediction in Twitter. In *Proceedings of the Workshop on Information Extraction and Entity Analytics on Social Media Data* (Mumbai, India, 2012), The COLING 2012 Organizing Committee, pp. 23–38.

[43] Wehrmann, J., Becker, W., Cagnini, H. E., and Barros, R. C. A character-based convolutional neural network for language-agnostic Twitter sentiment analysis. In *2017 International Joint Conference on Neural Networks (IJCNN)* (Anchorage, AK, USA, 2017), IEEE, pp. 2384–2391.

[44] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems 30: 31st Annual Conference on Neural Information Processing Systems (NIPS 2017)* (Long Beach, CA, USA, 2017), U. von Luxburg, S. Bengio, R. Fergus, R. Garnett, I. Guyon, H. Wallach and S.V.N. Vishwanathan, Eds., Curran Associates, Inc., pp. 4149–4159.

[45] Xie, Y., and Jiang, H. Stock market forecasting based on text mining technology: A support vector machine method, 2019. Working paper version available from arXiv: https://doi.org/10.48550/arXiv.1909.12789.

[46] Xu, F., and Keelj, V. Collective sentiment mining of microblogs in 24-hour stock price movement prediction. In *2014 IEEE 16th Conference on Business Informatics* (Geneva, Switzerland, 2014), vol. 2, IEEE, pp. 60–67.

[47] Yang, X., Loua, M. A., Wu, M., Huang, L., and Gao, Q. Multi-granularity stock prediction with sequential three-way decisions. *Information Sciences 621* (2023), 524–544.

[48] Zhang, X., Qu, S., Huang, J., Fang, B., and Yu, P. Stock market prediction via multi-source multiple instance learning. *IEEE Access 6* (2018), 50720–50728.

[49] Zhao, Y., and Yang, G. Deep learning-based integrated framework for stock price movement prediction. *Applied Soft Computing 133* (2023), 109921.