sensors

Article

Stock Price Movement Prediction Using Sentiment Analysis and CandleStick Chart Representation

Trang-Thi Ho * and Yennun Huang

Research Center for Information Technology Innovation, Academia Sinica, Taipei 10607, Taiwan; yennunhuang@citi.sinica.edu.tw
* Correspondence: hothitrang@citi.sinica.edu.tw

Abstract: Determining the price movement of stocks is a challenging problem to solve because of factors such as industry performance, economic variables, investor sentiment, company news, company performance, and social media sentiment. People can predict the price movement of stocks by applying machine learning algorithms on information contained in historical data, stock candlestick-chart data, and social-media data. However, it is hard to predict stock movement based on a single classifier. In this study, we proposed a multichannel collaborative network by incorporating candlestick-chart and social-media data for stock trend predictions. We first extracted the social media sentiment features using the Natural Language Toolkit and sentiment analysis data from Twitter. We then transformed the stock's historical time series data into a candlestick chart to elucidate patterns in the stock's movement. Finally, we integrated the stock's sentiment features and its candlestick chart to predict the stock price movement over 4-, 6-, 8-, and 10-day time periods. Our collaborative network consisted of two branches: the first branch contained a one-dimensional convolutional neural network (CNN) performing sentiment classification. The second branch included a two-dimensional (2D) CNN performing image classifications based on 2D candlestick chart data. We evaluated our model for five high-demand stocks (Apple, Tesla, IBM, Amazon, and Google) and determined that our collaborative network achieved promising results and compared favorably against single-network models using either sentiment data or candlestick charts alone. The proposed method obtained the most favorable performance with 75.38% accuracy for Apple stock. We also found that the stock price prediction achieved more favorable performance over longer periods of time compared with shorter periods of time.

Keywords: stock market; sentiment analysis; candlestick chart; convolution neural network

check for updates

Citation: Ho, T.-T.; Huang, Y. Stock Price Movement Prediction Using Sentiment Analysis and CandleStick Chart Representation. *Sensors* **2021**, *21*, 7957. https://doi.org/10.3390/s21237957

Academic Editors: Friedhelm Schwenker and Mariofanna Milanova

Received: 27 October 2021
Accepted: 25 November 2021
Published: 29 November 2021

1. Introduction

The stock market is a major topic in modern life. Investors can easily purchase new stocks and gain significant profits from dividends provided in the company's bonus program for shareholders. Investors can also trade their own stocks with the other traders in the stock market via stock brokerages and electronic trading platforms. Stock traders wish to buy stocks with values that are expected to increase and sell stocks with values that are expected to decrease. Therefore, stock traders must be able to predict the trends of a stock's behavior before making a trading decision to buy or sell a stock. The more accurate their prediction about the behavior of a stock is, the more profit they earn. Therefore, developing an automatic algorithm that can accurately predict market trends to help traders maximize profit is essential. However, determining stock market trends is a challenging problem because of factors such as industry performance, company news, company performance, investor sentiment, economic variables, and social media sentiment.

According to Fama's efficient market hypothesis [1], investment in the stock market involves risk, and the possibility of investors gaining an advantage by buying underrated stocks or selling stocks for an inflated price does not exist. To avoid high-risk stocks and secure higher profits, a stock trader has only one means of evaluating a company's performance before purchasing its stock. With the development of technological advances, deep learning—especially convolutional neural networks (CNNs)—has exhibited favorable performance in a range of research fields [2–15]. Many researchers have applied deep learning to the question of stock market prediction. There are several approaches for stock market prediction, such as analyzing indicators of historical time series data [16–20], or using candlestick chart converted from historical data [21–26], or analyzing the social media [27–32], or analyzing the financial news [33–36]. However, using a single classifier may not achieve maximum performance compared with using combined classifiers. Different classifiers can be combined using ensemble methods in machine learning to improve the prediction accuracy of individual classifiers. In this study, we proposed a novel framework for stock market prediction that incorporates both candlestick charts and sentiment analysis of social-media data as internal and external factors. To our knowledge, this is the first study to integrate candlestick charts and sentiment data from social media for the purpose of stock prediction.

Because social-media data are text data from tweets and candlestick chart data comprise color images, the information requires extraction and processing before being incorporated into a collaborative network. For social-media data, the Natural Language Toolkit (NLTK) was used for sentiment analysis of the content of tweets. We removed spam tweets and irrelevant information to help our model achieve accurate sentiment score features. The candlestick chart data were generated by mpl_finance module [37] by using four features (open, high, low, and close) of historical time series data. Data from candlestick charts were normalized before being input into a CNN for extraction of internal features. A multibranch network was designed to incorporate features from social-media and candlestick-chart data. The output of these branches was concatenated in a dense layer. The last layer of the multibranch network contained two neurons to classify the stock trend as increasing or decreasing in the near future.

In summary, this study makes the following contributions:

- We employed a one-dimensional (1D) CNN (1D-CNN) that performed more favorably than traditional machine learning and long short-term memory models for stock trend prediction based on social-media data.
- We proposed a novel collaborative deep learning network for stock-price-movement prediction. The main features were incorporated internal (candlestick chart) and external (social media sentiment) stock features that helped to improve the accuracy of the prediction of stock price movement.
- We formulated a new accuracy-level metric for comparing the accuracy ranges between classification models for stock trend prediction.
- We observed that price movement prediction for a "high-demand stock" achieved more favorable performance in the long term.

In the remainder of this article, we first review related literature in Section 2. We then explain our methodology in Section 3. Section 4 presents the experimental evaluations, and we finally conclude the article in Section 5.

2. Related Works

Many researchers have developed various prediction models for stock market prediction to help traders make the correct decision on their stocks. In 1990, Schneburg [38] conducted a study on the German stock market. A machine learning architecture with back-propagation was built based on historical time-series data. In 2007, Khoa [39] proposed a simply recurrent neural network for stock prediction and concluded that this simply recurrent neural network achieves better results compared to back-propagation neural networks due to its "capturing capabilities." After that, Ou and Wang [20] utilized the Hong Kong stock market to build a stock market prediction model with ten machine learning algorithms. They demonstrated that SVM and LS-SVM achieve better performance than the other methods. In 2016, Chen proposed planar feature representation methods and deep convolutional neural networks to improve the algorithmic trading framework [40].

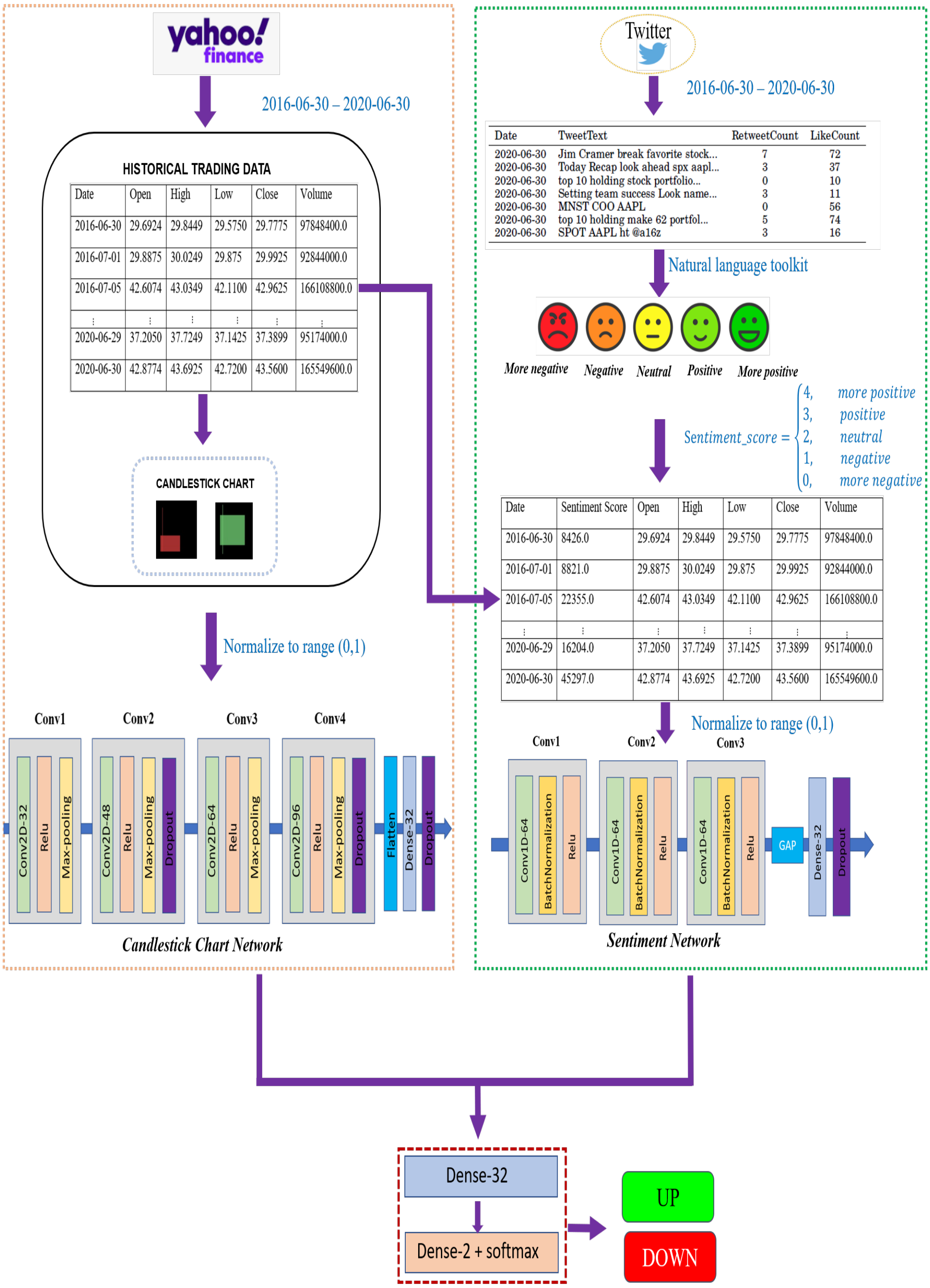
In the last decade, sentiment analysis has carried great importance because of the huge amount of textual data on news and social media platforms. There is a significant amount of research on mining opinions of users for different application areas [41–45]. Ali performed sentiment analysis to detect transportation entities in the large corpus [42] and to detect traffic accidents to reduce serious injuries [43]. Basiri proposed a sentiment analysis method on the Twitter dataset using the attention-based bidirectional CNN-RNN deep model [44]. Li [45] proposed a bidirectional emotional recurrent unit for exploring the emotion in conversation. Using the same trend for stock prediction, some researchers have started to utilize sentiment analysis to analyze the stock market's movements. In 2011, J. Bollen proposed a sentiment analysis method to predict the movement of the Dow Jones Industrial Average (DJIA) stock market based on the famous microblogging site Twitter. Khatri and Srivastava [30] explored the relationship between text sentiment from tweets, comments on the Stock Twist website, and the stock price of Facebook, Apple, Google, Oracle, and Microsoft stocks from Yahoo Finance. In 2017, Urolagin used an SVM classifier to perform sentiment classification and predict stock market status based on stock price data from Yahoo Finance and social-media data from Twitter [27]. He explored an association between tweet features and the stock prices of a company. Not long after that, Chakraborty proposed a stock-market-movement prediction model based on tweets' sentiment of Appple Inc stock [28]. The SVM classifier was used for sentiment classification, while the stock movement prediction model was trained by applying the boosted regression tree. In 2020, Khan [46] utilized algorithms on social media and financial news data to discover the impact of both data on stock movement.

Many researchers have found different methods to approach stock data; they transformed the historical time series data into other forms to learn more patterns inside the stock movement. Candlestick charts have been used to visualize the daily price and stock market movement. In 2013, Prado utilized sixteen candlestick patterns to forecast stock movement for Brazilian stocks [26]. Tsai combined candlestick charts with seven different wavelet-based textures to predict stock movement [22]. In 2017, Hu employed a convolutional encoder to learn the candlestick chart patterns to build a decision-making system for the stock market. Our previous work [21] proposed a deep convolutional neural network for stock market prediction based on candlestick charts for two different stock markets (Taiwan50 and Indo10). In 2020, Birogul [25] employed a real-time object detection system (YOLO) to recognize buy–sell objects inside 2D candlestick charts; from these buy–sell objects, the trader can make their decision on the stock. Not long after that, Hung [24] proposed a deep predictor framework for price movement based on candlestick charts. He explored a CNN-autoencoder to acquire the best sub-chart representation and applied recurrent neural networks to predict the stock price movement.

Unlike existing methods, we did not solely rely on sentiment analysis of social media posts or explore patterns in the candlestick charts. In this study, we proposed a collaborative network incorporating candlestick charts and social-media data for stock movement prediction. To our knowledge, this is the first study to use both candlestick charts and social sentiment data to predict stock prices. Using both types of data is more effective in predicting stock trends because both types of data can change and affect movements in a stock's price and in traders' decisions.

3. Approach

In this section, we present a joint multichannel framework for stock movement prediction. This section is organized as follows: first, we introduce the sentiment analysis for stock trend prediction in Section 3.1. We then describe the candlestick chart generation and its branch network for stock price movement in Section 3.2. Finally, we present a collaborative network incorporating all the features from sentiment analysis and candlestick charts that helps to improve prediction performance in Section 3.3. Section 3.4 introduces performance measures utilized to evaluate our classification model performance. Figure 1 exhibits the architecture of our stock prediction framework.



Date	Open	High	Low	Close	Volume
2016-06-30	29.6924	29.8449	29.5750	29.7775	97848400.0
2016-07-01	29.8875	30.0249	29.875	29.9925	92844000.0
2016-07-05	42.6074	43.0349	42.1100	42.9625	166108800.0
⋮	⋮	⋮	⋮	⋮	⋮
2020-06-29	37.2050	37.7249	37.1425	37.3899	95174000.0
2020-06-30	42.8774	43.6925	42.7200	43.5600	165549600.0

Date	TweetText	RetweetCount	LikeCount
2020-06-30	Jim Cramer break favorite stock...	7	72
2020-06-30	Today Recap look ahead spx aapl...	3	37
2020-06-30	top 10 holding stock portfolio...	0	10
2020-06-30	Setting team success Look name...	3	11
2020-06-30	MNST COO AAPL	0	56
2020-06-30	top 10 holding make 62 portfol...	5	74
2020-06-30	SPOT AAPL ht @a16z	3	16

$$Sentiment\_score = \begin{cases} 4, & more\ positive \\ 3, & positive \\ 2, & neutral \\ 1, & negative \\ 0, & more\ negative \end{cases}$$

Date	Sentiment Score	Open	High	Low	Close	Volume
2016-06-30	8426.0	29.6924	29.8449	29.5750	29.7775	97848400.0
2016-07-01	8821.0	29.8875	30.0249	29.875	29.9925	92844000.0
2016-07-05	22355.0	42.6074	43.0349	42.1100	42.9625	166108800.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2020-06-29	16204.0	37.2050	37.7249	37.1425	37.3899	95174000.0
2020-06-30	45297.0	42.8774	43.6925	42.7200	43.5600	165549600.0

Figure 1. Architecture of the multibranch network for stock trend prediction.

3.1. Sentiment Analysis for Stock Trend Prediction

3.1.1. Sentiment Analysis Using Social Media Data

Twitter is a social media platform on which users can post and interact with messages known as "tweets." Registered users can post, like, and re-tweet the tweets from followed friends, while unregistered users can only see those that are in public status. Like the other social media platforms, Twitter has become a popular tool to connect people and transfer information and personal feelings. It also provides a convenient method to catch market sentiment. Due to its popularity, many investment communities have adopted the sentiment analysis from Twitter to predict the stock market trend. If Twitter users want to show that they share financial information, they can write a tweet including a stock ticker with a dollar sign. In this work, we performed sentiment analysis by analyzing each word in a tweet to obtain insights into users' sentiment about a particular stock. The Natural Language Toolkit (NLTK) [47] contains various utilities that allowed us to manipulate and analyze linguistic data effectively. Among its advanced features are text classifiers; we can employ a pre-trained sentiment analyzer called valence aware dictionary and sentiment reasoner (VADER) to classify the tweet data into overall positive, neutral, and negative categories. The VADER library will return four values: (1) pos: the probability of the sentiment to be positive, (2) neu: the probability of sentiment to be neutral, (3) neg: the probability of the sentiment to be negative, and (4) compound: the normalized compound score, which calculates the sum of all lexicon ratings and takes values from −1 to 1. We classified the sentiment of the tweet into five classes (more positive, positive, more negative, negative, and neutral) by the following equation:

$$Sentiment = \begin{cases} more\ positive & if\ compound > 0.5 \\ positive & if\ compound \in [0.5, 0) \\ neutral & if\ compound = 0 \\ negative & if\ compound \in (0, -0.5] \\ more\ negative & if\ compound < -0.5 \end{cases} \tag{1}$$

Some research on stock trends has shown that sometimes positive sentiment has an impact [48], while other times negative sentiment has an impact [49]. Therefore, it is important to track the sentiment feature because it may have different effects. We utilized a sentiment score to measure our tweet's sentiment as the following equation:

$$Sentiment\_score = \begin{cases} 4 & if\ tweet\ is\ more\ positive \\ 3 & if\ tweet\ is\ positive \\ 2 & if\ tweet\ is\ neutral \\ 1 & if\ tweet\ is\ negative \\ 0 & if\ tweet\ is\ more\ negative \end{cases} \tag{2}$$

The overall sentiment score of a given day is the combined sentiment score of all the tweets for that particular day. As shown in Equation (2), if a particular day has a higher overall sentiment score, then it means that the sentiment positivity is higher on that day.

3.1.2. Sentiment Networks

To evaluate the stock prediction performance using sentiment analysis, we utilized five machine learning methods in our studies such as random forest (RF), linear support vector classification (LinearSVC), Gaussian naive Bayes (GaussianNB), long short-term memory (LSTM), and 1D-CNN.

Random Forest (RF)

RF is an ensemble machine learning algorithm that is used to solve classification and regression problems. It has shown compelling efficiency for stock market prediction using sentiment analysis on media and news data [46]. A random forest algorithm involves constructing a large number of decision trees from bootstrap samples in a training dataset, like bagging. In general, a decision tree is a decision support technique that forms a tree-like structure. The decision-trees method combines the "bagging" idea and the random selection of features from training sets. A decision-tree method divides the training dataset into branches, which further segregates it into other branches. This process continues until reaching a leaf node. Unlike bagging, the processes of finding the root node and splitting the feature nodes will run randomly in a random forest method. We employed our random forest algorithm using skicit-learn [50].

Similar to [46], we set the $min\_samples\_leaf$ equal to 1, and the criterion used Gini impurity. While the max_depth equaled 2 and the $random\_state$ parameter equaled 0, the other parameters received the default values.

Linear Support Vector Classification (LinearSVC)

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression, and outliers detection. The support vector machines method aims to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The objective of SVM is to find a plane that has the maximum margin because maximizing the margin distance means that the future data points can be classified with more confidence. SVM has proved successful in the sentiment classification [27,33,41,46,51]. LinearSVC is a support vector machine that generates a linear classifier. We explored the LinearSVC in our work since LinearSVC has more flexibility in choosing penalties and the loss function, and it scales better to the larger number of samples.

In our problem settings, we set $random\_state$ equal to 42 and the $class\_weight$ parameter equal to "balanced," while the other parameters received the default values. We also employed our LinearSVC algorithm using skicit-learn [50].

Gaussian Naive Bayes (GaussianNB)

A naive Bayes classifier is a supervised machine learning classification algorithm based on the Bayes theorem [52]. The algorithm is highly effective when the dimensionality of the inputs is high. Some researchers have applied naïve Bayes to classify text sentiment [27,51,53]. GaussianNB is a variant of naive Bayes that follows a Gaussian normal distribution and that supports continuous values. It is one of the most widely used machine learning algorithms. Similar to [46], we applied GaussianNB as one of the machine learning classifiers in our sentiment analysis problem. The Gaussian normal distribution is calculated using the following equation:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{3}$$

where $f(x)$, σ, and μ represent the probability density function, the standard deviation, and the mean, respectively. The Gaussian naive Bayes model calculates the means and standard deviation of input values for each class to summarize the distribution. At every data point, the z-score distance between that point and each class-mean is calculated and stored.

Long Short-Term Memory (LSTM)

A recurrent neural network (RNN) is a type of artificial neural network that maintains hidden internal states to model inputs with dependence through directed cyclic connections between its units. RNN is usually used to process sequential data or time-series data. Unlike traditional neural networks, the output of RNN depends on the primary elements within the sequence. For example, we wanted to predict the word following a given sentence. It is clear that the predicted word will have a semantic relationship with the other words in that sentence. RNN generates outputs based on current and previous inputs. The hidden states in RNN hold information on previous inputs. Figure 2 shows an RNN architecture unrolled into a full network, in which x_t, h_t, y_t represents the input, hidden, and output state at the time t, respectively.

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \tag{4}$$

$$y_t = W_{hy}h_t + b_Y, \tag{5}$$

where W_{hh}, W_{hy}, and W_{xh} represent the hidden-to-hidden layer, the output-to-hidden layer, and the input-to-hidden layer weight matrices, respectively. b_h and b_y are the biases of the hidden and the output layers, respectively.

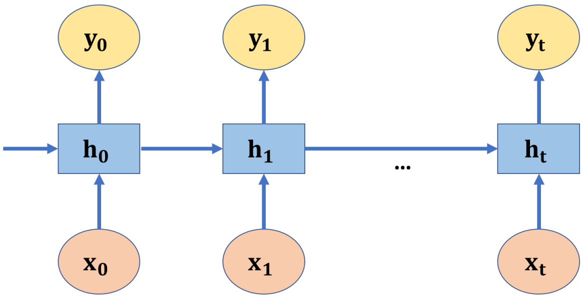


Figure 2. A recurrent neural network architecture.

The LSTM network is an extension of RNN by adding three gates: a forget gate to control whether to forget the current state, an input gate to control if the input should be stored, an output gate to control whether to output the state. The activations of LSTM are calculated as the following equations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \tag{6}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right), \tag{7}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \tag{8}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), \tag{9}$$

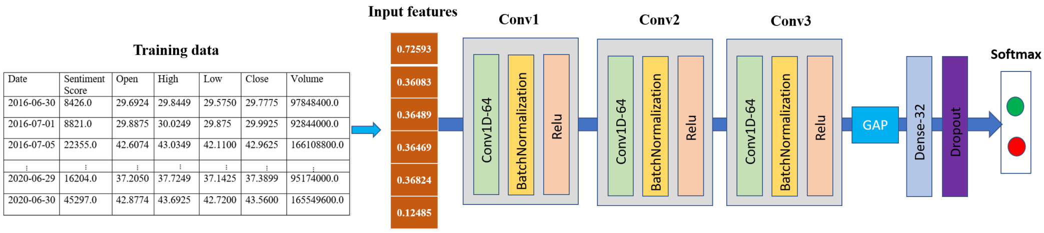
$$c_t h_t = o_t \tanh(c_t), \tag{10}$$

where i_t, f_t, c_t, and o_t denote the input gate, the forget gate, the cell activation vectors, and the output gate at time t. b_i, b_f, b_o, and b_c are the biases of the gates.

LSTM has shown great success in solving sequence modeling problems such as machine translation, time series, and sentiment analysis [54–56]. In this work, we used a simple LSTM with three layers. The first LSTM layer contained thirty internal units followed by a dropout layer with a probability of 0.5. The second LSTM layer contained 256 units. The final dense layer with two units corresponded to the number of classes of our stock trend prediction.

1D-CNN

In Wang's work [57], it was demonstrated that a fully convolutional network (FCN) with global average pooling (GAP) achieves premium performance to other state-of-the-art approaches in time series data. Inspired by this work, we designed a 1D-CNN network for performing sentiment analysis for our stock price prediction. The selected parameters (the kernel size and the number of filters) were found via random search using Keras Tuner [58]. Our 1D-CNN sentiment network consisted of three 3 × 3 1D convolutional layers. Each layer comprised 64 channels, followed by the BatchNormalization layer and ReLu activation. There was a GAP layer with 32 units, ReLu activation, and one dropout layer with a dropout probability of 0.5. Finally, the softmax output layer with two units corresponded to the number of classes of our stock trend prediction. Figure 3 exhibits our 1D-CNN architecture.


Training data

Date	Sentiment Score	Open	High	Low	Close	Volume
2016-06-30	8426.0	29.6924	29.8449	29.5750	29.7775	97848400.0
2016-07-01	8821.0	29.8875	30.0249	29.875	29.9925	92844000.0
2016-07-05	22355.0	42.6074	43.0349	42.1100	42.9625	166108800.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2020-06-29	16204.0	37.2050	37.7249	37.1425	37.3899	95174000.0
2020-06-30	45297.0	42.8774	43.6925	42.7200	43.5600	165549600.0

Figure 3. 1D Convolutional neural network architecture for sentiment data.

3.2. *CandleStickChart for Stock Trend Prediction*

3.2.1. Candlestick Chart

The candlestick chart was developed by Homma [59], a famous rice trader known in Sakata city, Japan. After that, many Japanese traders used it to predict the future prices in rice trading contracts [60]. A candlestick chart is a combination between a line-chart and a bar used to describe the price movements for a given period of time. Figure 4 shows an example of a candlestick chart. Candlestick charts consist of three components, such as an upper shadow, a lower shadow, and a real body. Each bar of the candlestick chart represents the information of a trading day, such as the open, the close, the low, and the high price. The thick part of the candlestick chart represents the different distances between the opening and closing prices. The different colors of the candlestick also indicate a different meaning. The real body will be filled in a red color if the open price is higher than the closing price. Otherwise, the real body will be filled with a green color. The upper and lower lines at the end of the real body represent the upper and lower shadows, respectively. The upper and lower shadows represents the high and low price ranges of candlestick charts in a given time period, respectively.

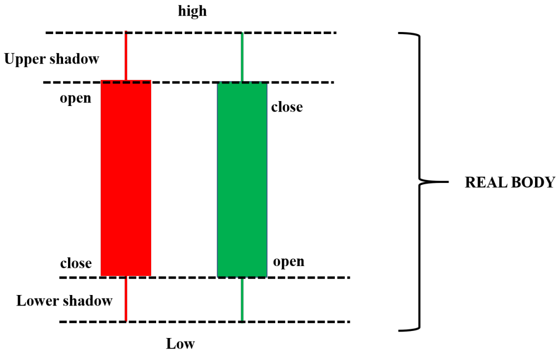


Figure 4. An example of a candlestick chart.

3.2.2. Candlestick Chart Network

Our previous work [21] demonstrated that a 2D-convolutional neural network (2D-CNN) outperformed the other popular networks (VGG16, RestNet50, and random forest) for stock price movement prediction using candlestick chart data. Similar to our previous work [21], we employed this 2D-CNN for our candlestick chart network. The 2D-CNN architecture contained four 3 × 3 2D convolutional layers. Layer 1, layer 2, layer 3, and layer 4 had 32, 48, 64, and 96 channels, respectively. Each layer contained a 2 × 2 max pooling layer, with ReLu activation. There were three dropouts and a fully connected layer with 32 units, with a Relu activation. Finally, a soft-max output layer with two units represents stock trend prediction. Figure 5 exhibits our 2D-CNN architecture.

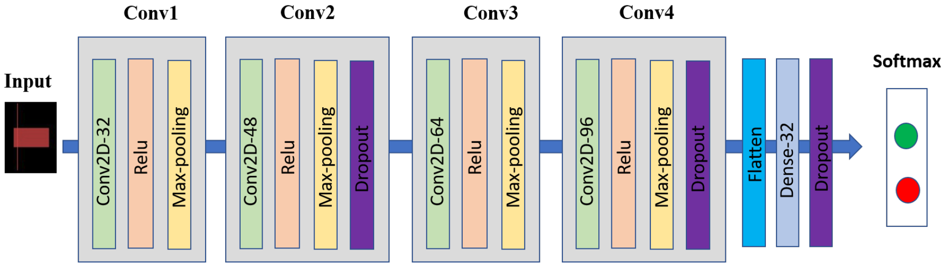


Figure 5. 2D convolutional neural network architecture for candlestick chart data.

3.3. *Joint Sentiment and Candlestick Chart Model*

To improve our prediction result, we proposed a joint prediction model based on sentiment analysis and candlestick chart patterns that could incorporate all of a stock's features. Our collaborative network included two branches: (1) the first branch was a 1D-CNN to perform sentiment analysis as described in Section 3.1.2. (2) The second branch performed image classification based on the candlestick charts through the 2D-CNN described in Section 3.2.2. Finally, the output of these branches was then concatenated, and they were fed into a standard set of dense layers where the last layer had two neurons to classify the respective stock price as increasing or decreasing in the near future. Figure 1 exhibits our joint prediction model.

3.4. Performance Measures

There were various statistics measures utilized to evaluate the performance of the classification model. In this study, we used the accuracy primary classification metric and three within-class classification metrics such as Precision, Recall, and the F-score.

The accuracy metric determines how good our model is, and it is a ratio of the number of correct predictions to the total observations. The accuracy metric is formulated as follows:

$$Accuracy = \frac{number\,of\,correct\,predictions}{total\,number\,of\,predictions} = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

where TP is True Positive, representing that the increased future trend of the stock has been identified as an increasing trend; FP is False Positive, representing that the decreased future trend of the stock has been identified as an increasing trend; TN is True Negative, representing that the decreased future trend of stock has been identified as a decreasing trend; and FN is False Negative, representing that the increased future trend of stock has been identified as a decreasing trend.

The precision metric represents how precise our model is out of those predicted increased trends and how many of them are actual increasing trends. This metric is a good measure to determine when the cost of False Positive is high. The precision metric can be calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

The recall metric calculates how many of the actual increasing trends our model captures by labeling it as an increasing trend (True Positive). This metric is a good measure to determine when the cost associated with False Negative is high. Recall that the metric can be calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

The F-score was used to seek a balance between Precision and Recall. The value of the F-score was 1.0, indicating the perfect Precision and Recall. In contrast, the lowest value of the F-score was 0, indicating that either the Precision or the Recall was zero. The F-score is formulated as follows:

$$F\text{-}score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{14}$$

Additionally, sometimes the accuracy metric values did not differ substantially between the models. Because stock traders are highly aware of the range of profits they can earn in stock trading, we proposed an accuracy_level metric to show the accuracy range between classification models. We defined the accuracy_level as the following equation:

$$Accuracy\_level = \begin{cases} 0, & accuracy \in [0,10)\% \\ 1, & accuracy \in [10,20)\% \\ 2, & accuracy \in [20,30)\% \\ 3, & accuracy \in [30,40)\% \\ 4, & accuracy \in [40,50)\% \\ 5, & accuracy \in [50,60)\% \\ 6, & accuracy \in [60,70)\% \\ 7, & accuracy \in [70,80)\% \\ 8, & accuracy \in [80,90)\% \\ 9, & accuracy \in [90,100)\% \\ 10, & accuracy = 100\% \end{cases} \quad (15)$$

4. Experiment

In this section, we first describe the details of our data-collection and processing procedures. We then describe the improvement in sentiment analysis achieved by using a CNN. Finally, we demonstrate the improvement in our joint network by evaluating different aspects of the prediction of movement in stock prices and discuss the prediction of stock trends in different periods of time.

4.1. Data Collection and Processing

4.1.1. Historical Data from Yahoo Finance

We selected five high-demand stocks in the United States: Apple, Tesla, IBM, Amazon, and Google (abbreviated as AAPL, TSLA, IBM, AMZN, and GOOG, respectively). Stocks were chosen based on a variety of factors, including their market capitalization in the top stock markets in the United States and their status as well-known stocks popular with global institutional investors. From Yahoo! Finance's API, we collected historical time series data for each stock. The time period of stock data was four years of data from 30 June 2016 to 30 June 2020. We then extracted six features for each stock, including the date, the open, the high, the low, the close, and the volume corresponding to the stock's traded date, the stock's open price, the stock's highest trading price, the stock's lowest trading price, the stock's close price, and the number of shares traded, respectively. The information of selected stocks is shown in Table 1.

Table 1. Selected stocks' symbols, stock name, and stock exchange.

No	Stock Name	Ticker Symbol	Country/Stock Exchange
1	Apple	AAPL	USA
2	Tesla	TSLA	USA
3	IBM	IBM	USA
4	Amazon	AMZN	USA
5	Alphabet (Google)	GOOG	USA

In this work, we wanted to use features information in the current day as input X_i to predict the future trend of stock in the next n days. It means that we wanted to find the difference between the closing price of a stock's current day and the closing price after n days. The future trend's label = 1 represents that the stock's closing price will increase after n days. If the future trend's label = 0, the stock's closing price will decrease or not change after n days. The future trend's label is determined with the following equation:

$$Label = \begin{cases} 1, & P_t > P_{t+n} \\ 0, & P_t \leq P_{t+n} \end{cases} \quad (16)$$

We would find the future trend of the stock price in the next 4 days, 6 days, 8 days, and 10 days. We randomly kept 20% of the data for testing and the rest of the data for training from our stock data.

4.1.2. Sentiment Data from Social Media (Twitter)

To collect the tweets from Twitter for each particular stock, we utilized the snscrape [61] library, which allowed us to scrape tweets without the restrictions of Tweepy [62]. Snscrape is a scraper for social networking services (SNS). It will scrape things such as profiles, hashtags, or searches and returns the discovered items, e.g., the relevant posts. In our work, the Python application would obtain input parameters such as a search query, symbols of the selected stock, the start date, and the end date. All tweets of the selected stocks were downloaded in raw format between a start date and an end date. The tweet data contained four features: Date, TweetText, ReTweetCount, and LikeCount—which indicate the date on which the tweet was posted, the tweet's content, the number of re-tweets, and the number of likes on that tweet, respectively. Table 2 shows an example of tweet data for AAPL stock.

Table 2. A view of tweet data for stock price movement prediction.

Date	TweetText	RetweetCount	LikeCount
30 June 2020	Jim Cramer break favorite stock…	7	72
30 June 2020	Today Recap look ahead spx aapl…	3	37
30 June 2020	top 10 holding stock portfolio…	0	10
30 June 2020	Setting team success Look name…	3	11
30 June 2020	MNST COO AAPL…	0	56
30 June 2020	top 10 holding make 62 portfol…	5	74
30 June 2020	SPOT AAPL ht @a16z…	3	16

We then processed the downloaded tweets by the following steps:

- Removing the unimportant/spam tweets that contained ReTweetCount and LikeCount values < 5 and <10, respectively.
- Tweets were split and converted into word tokens.
- Removing the non-Ascii char, hyperlinks, punctuations, tokenize, stopwords(for example, the, are, is, an, etc.) inside the tweets because these components carry no useful information for sentiment analysis.

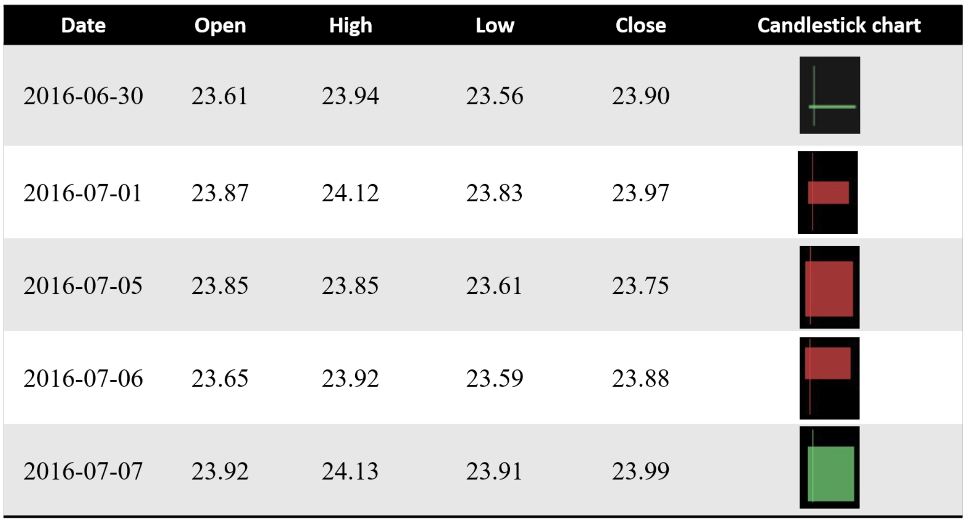
Finally, we applied the sentiment analysis for processed tweets on each day to obtain the overall sentiment score as described in Section 3.1 for that particular day. Table 3 shows an example of the final sentiment data format.

Table 3. A view of sentiment data for stock price movement prediction.

Date	SentimentScore	Open	High	Low	Close	Volume
30 June 2016	36	23.61	23.94	23.56	23.90	143,345,600
1 July 2016	86	23.87	24.12	23.83	23.97	104,106,000
5 July 2016	182	23.85	23.85	23.61	23.75	110,820,800
6 July 2016	218	23.65	23.92	23.59	23.88	123,796,400
7 July 2016	240	23.92	24.13	23.91	23.99	100,558,400
8 July 2016	262	24.12	24.22	24.01	24.17	115,648,400
11 July 2016	322	24.19	24.41	24.18	24.25	95,179,600
12 July 2016	346	24.29	24.42	24.28	24.35	96,670,000

4.1.3. From Historical Data to Candlestick Chart

A candlestick chart essentially has four features—the open, the high, the low, and the close—called OHLC. To create a candlestick chart, we utilized the *mpl_finance* module [37] containing a new *matplotlib.finance* API that makes it easier to generate financial plots. Figure 6 illustrates an example of candlestick data.



Date	Open	High	Low	Close	Candlestick chart
2016-06-30	23.61	23.94	23.56	23.90	🟩
2016-07-01	23.87	24.12	23.83	23.97	🟥
2016-07-05	23.85	23.85	23.61	23.75	🟥
2016-07-06	23.65	23.92	23.59	23.88	🟥
2016-07-07	23.92	24.13	23.91	23.99	🟩

Figure 6. A view of candlestick chart data for stock price movement prediction.

4.2. Experimental Setting

Deep learning models are full of hyperparameters; therefore, finding the best configuration in the CNN model is a time-consuming and resource-consuming process. In this work, we performed some procedures to determine the optimal values of hyperparameters to achieve the model with the highest performance. Those hyperparameters were the batch size, the optimizer function, and the learning rate. We applied a GridSearchCV approach from the scikit-learn wrapper class in Keras API to find the optimal batch size and optimizer function. We defined the grid search of batch size in the set of (8, 16, and 24) and the optimizer function in (Adam, SGD, and RMSprop). The GridSearchCV constructed and evaluated the model for each combination of these two set parameters using three-fold cross-validation.

To control the learning rate and save the best model, we applied ReduceLROnPlateau and Modelcheckpoint callbacks API in Keras. The learning rate would be reduced by a factor of 0.5 if there was no improvement after 20 continuous epochs. This process continued until reaching the lower bound. We set the coarse learning rate = 0.001 and the lower bound on the learning rate = 0.0001. Modelcheckpoint stored the best model according to the validation loss value at every epoch.

4.3. Improvement in Stock Trend Prediction Based on Sentiment Analysis and 1D-CNN

To choose an effective stock trend prediction model based on sentiment analysis, we conducted experiments using the five methods discussed in Section 3. We implemented each method to predict the stock trend for five high-demand stocks (AAPL, TSLA, IBM, AMZN, and GOOG) over various time periods (4 days, 6 days, 8 days, and 10 days). The highest accuracy is highlighted in bold. As shown in Table 4, The 1D-CNN model achieved more favorable performance than the random forest (RF), LinearSVC, GaussianNB, and LSTM models in most cases. We achieved the highest accuracy of 71.36% on AAPL stock for the prediction of the subsequent 8 days using the 1D-CNN model. By comparison, the model achieved the lowest accuracy of 46.73% for IBM stock in the prediction of the subsequent 4 days using RF classifiers. Therefore, we used this 1D-CNN for our sentiment analysis of stock trend prediction.

Table 4. Accuracy improvement of stock trend prediction based on sentiment analysis and 1D-CNN.

Stock	Time Period	RF	LinearSVC	GaussianNB	LSTM	1D-CNN
AAPL	10 days	62.81%	51.76%	58.79%	62.81%	**71.36%**
	8 days	61.31%	56.28%	60.30%	60.30%	**71.86%**
	6 days	56.78%	55.28%	59.30%	57.29%	**65.33%**
	4 days	61.31%	51.76%	55.28%	60.80%	**67.84%**
TSLA	10 days	60.80%	55.78%	58.29%	**64.82%**	63.32%
	8 days	57.29%	55.78%	56.28%	60.30%	**63.32%**
	6 days	58.79%	54.77%	52.76%	61.81%	**68.34%**
	4 days	57.29%	55.78%	57.29%	54.27%	**59.80%**
IBM	10 days	60.30%	62.81%	61.31%	60.80%	**64.82%**
	8 days	59.80%	60.30%	59.80%	62.31%	**64.82%**
	6 days	55.28%	56.28%	59.30%	56.78%	**59.80%**
	4 days	46.73%	56.78%	54.27%	**57.79%**	52.76%
AMZN	10 days	64.80%	52.80%	61.20%	62.80%	**69.60%**
	8 days	61.60%	53.60%	60.00%	58.00%	**64.80%**
	6 days	61.20%	58.40%	59.60%	56.40%	**61.60%**
	4 days	**64.40%**	60.80%	64.00%	60.40%	61.60%
GOOG	10 days	55.28%	59.30%	53.77%	55.78%	**67.34%**
	8 days	57.29%	59.30%	56.28%	56.28%	**67.34%**
	6 days	56.78%	57.29%	57.29%	**58.79%**	**58.79%**
	4 days	54.27%	53.77%	56.78%	59.30%	**62.81%**

4.4. Improvement inStock Trend Prediction Based on the Joint Network

To further improve our stock prediction result, we proposed a collaborative network with multiple channels that can incorporate all the features from sentiment analysis and candlestick charts. In this section, we first evaluate different aspects of our collaborative network using accuracy evaluation and accuracy0level metrics on five high-demand stocks (AAPL, TSLA, IBM, AMZN, and GOOG). We then evaluate the stock trend prediction of our proposed model on different time periods.

4.4.1. Ablation Study

In this section, we examine how each component in the proposed method affected our prediction performance. We evaluate three models: (1) sentiment analysis model, (2) candlestick chart model, and (3) our full model (joint network based on sentiment analysis and candlestick charts).

- Sentiment: the sentiment analysis network based on social-media data and the 1D-convolutional neural network described in Section 3.1.2.
- Candlestick chart: 2D-convolutional neural network based on candlestick-chart data described in Section 3.2.2.

Table 5 summarizes our ablation study results on different time periods (4 days, 6 days, 8 days, and 10 days). The highest accuracy and accuracy_level are highlighted in bold. We observed a significant performance improvement when the proposed method was used, indicating the benefits of stock prediction with a multichannel network. As shown in Table 5, rhe sentiment network seemed to work well with a 4-day time period, while the proposed method worked well for the other time periods (6 days, 8 days, and 10 days). The single-channel network using candlestick-chart data achieved the lowest results; however, it helped to improve the accuracy performance up by 8% when combing with the single-channel sentiment network in the multichannel proposed network.

Table 5. Accuracy improvement in stock trend prediction based on the joint network.

Stock	Period Time	Sentiment	Candlestick Chart	Joint Network
		Accuracy (Accuracy_Level)	Accuracy (Accuracy_Level)	Accuracy (Accuracy_Level)
AAPL	10 days	71.36% **(7)**	62.81% (6)	**75.38% (7)**
	8 days	71.86% **(7)**	61.31% (6)	**74.37% (7)**
	6 days	65.33% **(6)**	57.29% (5)	**66.83% (6)**
	4 days	**67.84% (6)**	61.31% **(6)**	64.82% **(6)**
TSLA	10 days	63.32% (6)	50.75% (5)	**71.86% (7)**
	8 days	63.32% **(6)**	52.76% (5)	**69.35% (6)**
	6 days	**68.34% (6)**	56.28% (5)	64.32% **(6)**
	4 days	59.80% (5)	52.76% (5)	**64.32% (6)**
IBM	10 days	64.82% **(6)**	47.74% (4)	**67.84% (6)**
	8 days	64.82% **(6)**	49.25% (4)	**69.35% (6)**
	6 days	59.80% (5)	49.25% (4)	**60.08% (6)**
	4 days	**52.76% (5)**	44.72% (4)	51.26% **(5)**
AMZN	10 days	69.60% (6)	61.20% (6)	**74.80% (7)**
	8 days	**64.80% (6)**	57.60% (5)	64.00% **(6)**
	6 days	61.60% **(6)**	56.40% (5)	**67.60% (6)**
	4 days	61.60% **(6)**	60.40% **(6)**	**66.00% (6)**
GOOG	10 days	**67.34% (6)**	54.77% (5)	**67.34% (6)**
	8 days	67.34% **(6)**	56.28% (5)	**67.84% (6)**
	6 days	58.79% (5)	56.78% (5)	**60.30% (6)**
	4 days	**62.81% (6)**	57.29% (5)	62.31% **(6)**

The same conclusion with the evaluation using the accuracy_level metric; Table 5 shows that utilizing the multichannel proposed network achieved a more favorable outcome than a single sentiment network and single candlestick-chart network, indicating that stock traders have a higher chance of earning profit when using our proposed method.

4.4.2. Stock Trend Prediction on Different Time Periods

Table 6 and Figure 7 show the performance of our proposed method for five high-demand stocks (AAPL, TSLA, IBM, AMZN, and GOOG) for different periods of time. The highest accuracy, precision, recall, and F-score of each stock are highlighted in bold. Figure 7 shows that the prediction accuracy increased gradually from day 4 to day 10. This indicates that stock performance over longer periods of time achieved a more favorable result than that over shorter periods of time. This may be because the stock's movement trended over a longer period. Therefore, this contributes to the accuracy of predictions over longer periods of time. As listed in Table 6, the proposed method obtained the most favorable performance with 75.38% accuracy on AAPL stock over a 10-day time period. By contrast, IBM stock, over a time period of 4 days, achieved the least-favorable performance with 51.26% accuracy. Additionally, our model achieved the most balanced results for TSLA stock over a 10-day time period with 71.86% accuracy, 68.42% precision, 79.59% recall, and a 73.58% F-score.

Table 6. Performance of proposed method for five hot stocks (AAPL, TSLA, IBM, AMZN, and GOOG) on different time periods (4, 6, 8, and 10 days).

Stock	Time Period	Accuracy	Precision	Recall	F-Score
AAPL	10 days	**75.38%**	**68.66%**	**62.16%**	**65.25%**
	8 days	74.37%	76.00%	49.35%	59.84%
	6 days	66.83%	64.18%	50.59%	56.58%
	4 days	64.48%	65.22%	19.48%	30.00%
TSLA	10 days	**71.86%**	**68.42%**	**79.59%**	**73.58%**
	8 days	69.35%	70.37%	60.64%	65.14%
	6 days	64.32%	60.53%	52.87%	56.44%
	4 days	64.32%	60.75%	69.15%	64.68%

Table 6. *Cont.*

Stock	Time Period	Accuracy	Precision	Recall	F-Score
IBM	10 days	67.84%	**72.73%**	61.53%	66.67%
	8 days	**69.35%**	70.83%	**67.33%**	**69.04%**
	6 days	60.80%	65.33%	48.51%	55.68%
	4 days	51.26%	57.83%	43.64%	49.74%
AMZN	10 days	**74.80%**	68.09%	**65.98%**	**67.02%**
	8 days	64.00%	61.11%	41.51%	49.44%
	6 days	67.60%	**68.42%**	47.71%	56.22%
	4 days	66.00%	60.94%	39.39%	47.85%
GOOG	10 days	67.34%	**82.05%**	35.56%	49.61%
	8 days	**67.84%**	78.05%	**36.78%**	**50.00%**
	6 days	60.30%	57.78%	30.23%	39.69%
	4 days	62.31%	66.67%	23.53%	34.78%

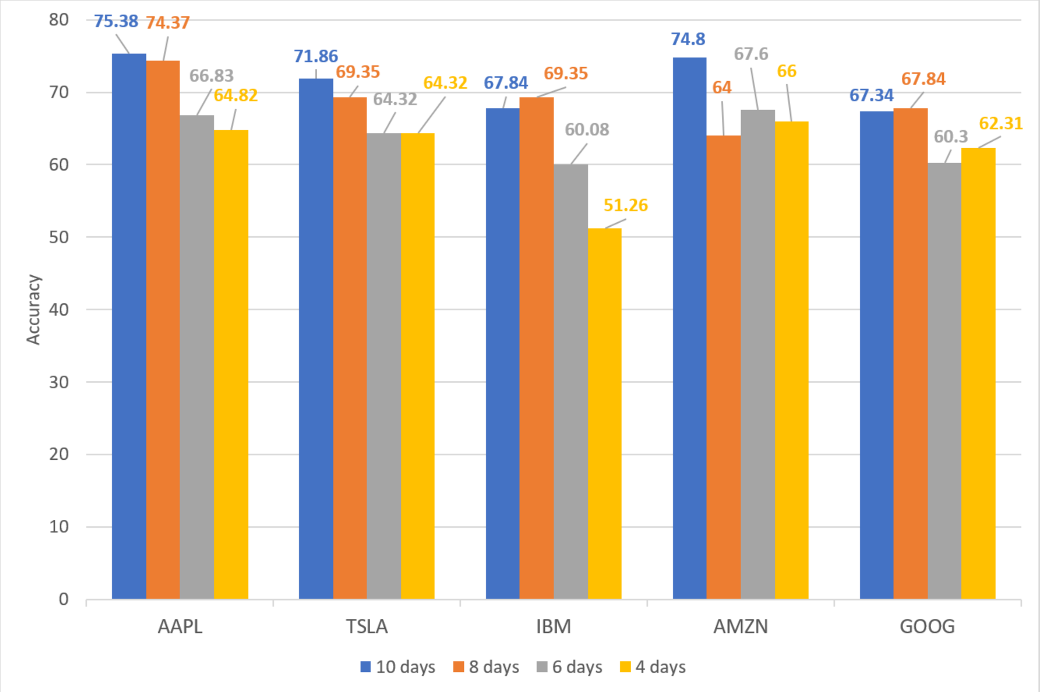


Figure 7. Performance of proposed method for five hot stocks (AAPL, TSLA, IBM, AMZN, and GOOG) on different period time (4, 6, 8, and 10 days).

5. Conclusions

In this study, we presented a new multichannel network for predicting stock prices based on five high-demand stocks (AAPL, TSLA, IBM, AMZN, and GOOG) selected by the size of their market capitalization. We first employed the Natural Language Toolkit to perform sentiment analysis for text content in Twitter tweets to obtain insights into users' sentiments about a particular stock. From classifying sentiment analysis results, we obtained an overall sentiment score for a given day for the stock. We then generated candlestick chart images based on historical time series data by using computer graphic techniques to visualize the daily price and stock movement for a given stock. These two types of data were input into our joint multichannel network. Our joint network consisted of two branches: (1) the first branch included a 1D-CNN performing classifications based on sentiment analysis. (2) The second branch contained a 2D-CNN performing image classifications based on 2D candlestick chart images. The outputs of the two branches were concatenated and fed into a standard set of dense layers where the last layer computed the data and predicted the stock movement for the near future. The experiment results indicated that our proposed joint network achieved promising results for stock prediction and outperformed networks using either single sentiment data or candlestick charts alone. This indicates that using both types of data is more effective in the prediction of stock trends because both types of data can change and affect a stock's price movement and traders' decisions. The experiment results also indicated that the performance of stock prediction over longer periods of time achieved a more favorable result than did predictions made over shorter periods of time. Our proposed method achieved the most favorable performance with 75.38% accuracy for AAPL stock over a 10-day time period.

Author Contributions: Conceptualization, T.-T.H.; data curation, T.-T.H.; formal analysis, T.-T.H.; methodology, T.-T.H.; project administration, T.-T.H. and Y.H.; supervision, Y.H.; writing—review and editing, T.-T.H. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Ministry of Science and Technology of the Republic of China under grant MOST109-2221-E-001-019-MY3. The work was also supported by Academia Sinica under grant AS-KPQ-109-DSTCP.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Malkiel, B.G.; Fama, E.F. Efficient capital markets a review of theory and empirical work. *J. Financ.* **1970**, *25*, 383–417. [CrossRef]
2. Hua, K.-L.; Ho, T.-T.; Jangtjik, K.-A.; Chen, Y.-J.; Yeh, M.-C. Artist-based painting classification using Markov random fields with convolution neural network. *Multimed. Tools Appl.* **2020**, *79*, 12635–12658. [CrossRef]
3. Jangtjik, K.A.; Ho, T.-T.; Yeh, M.-C.; Hua, K.-L. A CNN-LSTM framework for authorship classification of paintings. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 2866–2870.
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
5. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
6. Ho, T.-T.; Virtusio, J.J.; Chen, Y.-Y.; Hsu, C.-M.; Hua, K.-L. Sketch-guided deep portrait generation. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2020**, *16*, 1–18. [CrossRef]
7. Zhang, Z.; Xu, W.; Qin, Q.; Long, Z. Downscaling solar-induced chlorophyll fluorescence based on convolutional neural network method to monitor agricultural drought. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 1012–1028. [CrossRef]
8. Jiao, L.; Dong, S.; Zhang, S.; Xie, C.; Wang, H. AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection. *Comput. Electron. Agric.* **2020**, *174*, 105522. [CrossRef]
9. Liu, L.; Wu, F.-X.; Wang, Y.-P.; Wang, J. Multi-receptive-field CNN for semantic segmentation of medical images. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 3215–3225. [CrossRef] [PubMed]
10. Tseng, K.-K.; Zhang, R.; Chen, C.-M.; Hassan, M.M. DNetUnet: A semi-supervised CNN of medical image segmentation for super-computing AI service. *J. Supercomput.* **2021**, *77*, 3594–3615. [CrossRef]
11. Lu, W.; Li, J.; Wang, J.; Qin, L. A CNN-BiLSTM-AM method for stock price prediction. *Neural Comput. Appl.* **2021**, *33*, 4741–4753.
12. Mehtab, S.; Sen, J.; Dasgupta, S. Analysis and forecasting of financial time series using CNN and LSTM-based deep learning models. In Proceedings of the 2nd International Conference on Advances in Distributed Computing and Machine Learning (ICADCML'21), Bhubaneswar, India, 15–16 January 2021; Volume 33, pp. 15–16.
13. Perdiz, J.; Garrote, L.; Pires, G.; Nunes, U.J. A Reinforcement Learning Assisted Eye-Driven Computer Game Employing a Decision Tree-Based Approach and CNN Classification. *IEEE Access* **2021**, *9*, 46011–46021. [CrossRef]
14. Lee, J.; Lee, S. Robust CNN Compression Framework for Security-Sensitive Embedded Systems. *Appl. Sci.* **2021**, *11*, 1093. [CrossRef]
15. Ji, Y.; Zhang, H.; Zhang, Z.; Liu, M. CNN-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances. *Inf. Sci.* **2021**, *546*, 835–857. [CrossRef]
16. Hegazy, O.; Soliman, O.S.; Salam, M.A. A machine learning model for stock market prediction. *arXiv* **2014**, arXiv:1402.7351.
17. Shen, S.; Jiang, H.; Zhang, T. *Stock Market Forecasting Using Machine Learning Algorithms*; Department of Electrical Engineering, Stanford University: Stanford, CA, USA, 2012; pp. 1–5.
18. Chen, L.; Qiao, Z.; Wang, M.; Wang, C.; Du, R.; Stanley, H.E. Which artificial intelligence algorithm better predicts the Chinese stock market? *IEEE Access* **2018**, *6*, 48625–48633. [CrossRef]
19. Yetis, Y.; Kaplan, H.; Jamshidi, M. Stock market prediction by using artificial neural network. In Proceedings of the 2014 World Automation Congress (WAC), Waikoloa, HI, USA, 3–7 August 2014; pp. 718–722.
20. Chen, L.; Qiao, Z.; Wang, M.; Wang, C.; Du, R.; Stanley, H.E. Prediction of stock market index movement by ten data mining techniques? *Mod. Appl. Sci.* **2009**, *3*, 28–42.

21. Kusuma, R.M.I.; Ho, T.-T.; Kao, W.-C.; Ou, Y.-Y.; Hua, K.-L. Using deep learning neural networks and candlestick chart representation to predict stock market. *arXiv* **2019**, arXiv:1903.12258.
22. Tsai, C.-F.; Quan, Z.-Y. Stock prediction by searching for similarities in candlestick charts. *ACM Trans. Manag. Inf. Syst. (TMIS)* **2014**, *5*, 1–21. [CrossRef]
23. Hu, G.; Hu, Y.; Yang, K.; Yu, Z.; Sung, F.; Zhang, Z.; Xie, F.; Liu, J.; Robertson, N.; Hospedales, T.; et al. Deep stock representation learning: From candlestick charts to investment decisions. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2706–2710.
24. Hung, C.-C.; Chen, Y.-J. DPP: Deep predictor for price movement from candlestick charts. *PLoS ONE* **2021**, *16*, e0252404. [CrossRef] [PubMed]
25. Birogul, S.; Temür, G.; Kose, U. YOLO object recognition algorithm and "buy-sell decision" model over 2D candlestick chart. *IEEE Access* **2020**, *8*, 91894–91915. [CrossRef]
26. do Prado, H.A.; Ferneda, E.; Morais, L.C.R.; Luiz, A.J.B.; Matsura, E. On the effectiveness of candlestick chart analysis for the Brazilian stock market. *Neurocomputing* **2013**, *22*, 1136–1145. [CrossRef]
27. Urolagin, S. Text mining of tweet for sentiment classification and association with stock prices. In Proceedings of the 2017 International Conference on Computer and Applications (ICCA), Doha, United Arab Emirates, 6–7 September 2017; pp. 384–388.
28. Chakraborty, P.; Pria, U.S.; Rony, M.R.A.H.; Majumdar, M.A. Predicting stock movement using sentiment analysis of Twitter feed. In Proceedings of the 2017 6th International Conference on Informatics, Electronics and Vision & 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMHT), Himeji, Japan, 1–3 September 2017; pp. 1–6.
29. Mehta, P.; Pandya, S.; Kotecha, K. Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. *PeerJ Comput. Sci.* **2021**, *7*, e476. [CrossRef] [PubMed]
30. Khatri, S.K.; Srivastava, A. Using sentimental analysis in prediction of stock market investment. In Proceedings of the 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 7–9 September 2016; pp. 566–569.
31. Yan, D.; Zhou, G.; Zhao, X.; Tian, Y.; Yang, F. Predicting stock using microblog moods. *China Commun.* **2016**, *13*, 244–257. [CrossRef]
32. Zhou, Z.; Zhao, J.; Xu, K. Can online emotions predict the stock market in China? In Proceedings of the International Conference on Web Information Systems Engineering, Shanghai, China, 8–10 November 2016; pp. 328–342.
33. Dang, M.; Duong, D. Improvement methods for stock market prediction using financial news articles. In Proceedings of the 2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS), Danang, Vietnam, 14–16 September 2016; pp. 125–129.
34. Vargas, M.R.; dos Anjos, C.E.M.; Bichara, G.L.G.; Evsukoff, A.G. Deep leaming for stock market prediction using technical indicators and financial news articles. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de janeiro, Brazil, 8–13 July 2018; pp. 1–8.
35. Chen, W.; Yeo, C.K.; Lau, C.T.; Lee, B.S. A study on real-time low-quality content detection on Twitter from the users' perspective. *PLoS ONE* **2017**, *12*, e0182487. [CrossRef] [PubMed]
36. Li, Q.; Wang, T.; Li, P.; Liu, L.; Gong, Q.; Chen, Y. The effect of news and public mood on stock movements. *Inf. Sci.* **2014**, *278*, 826–840. [CrossRef]
37. dgoldfarb. Mplfinance a Matplotlib Utilities for the Visualization, and Visual Analysis, of Financial Data. 2020. Available online: https://github.com/matplotlib/mplfinance (accessed on 24 June 2021).
38. Schöneburg, E. Stock price prediction using neural networks: A project report. *Neurocomputing* **1990**, *2*, 17–27. [CrossRef]
39. Khoa, N.L.D.; Sakakibara, K.; Nishikawa, I. Stock price forecasting using back propagation neural networks with time and profit based adjusted weight factors. In Proceedings of the 2006 SICE-ICASE International Joint Conference, Busan, Korea, 18–21 October 2006; pp. 5484–5488.
40. Chen, J.-F.; Chen, W.-L.; Huang, C.-P.; Huang, S.-H.; Chen, A.-P. Financial time-series data analysis using deep convolutional neural networks. In Proceedings of the 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China, 16–18 November 2016; pp. 87–92.
41. Al Amrani, Y.; Lazaar, M.; El Kadiri, K.E. Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Comput. Sci.* **2018**, *127*, 511–520. [CrossRef]
42. Ali, F.; El-Sappagh, S.; Kwak, D. Fuzzy ontology and LSTM-based text mining: A transportation network monitoring system for assisting travel. *Sensors* **2019**, *19*, 234. [CrossRef]
43. Ali, F.; Ali, A.; Imran, M.; Naqvi, R.A.; Siddiqi, M.H.; Kwak, K.-S. Traffic accident detection and condition analysis based on social networking data. *Accid. Anal. Prev.* **2021**, *151*, 105973. [CrossRef] [PubMed]
44. Basiri, M.E.; Nemati, S.; Abdar, M.; Cambria, E.; Acharya, U.R. ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *115*, 279–294. [CrossRef]
45. Li, W.; Shao, W.; Ji, S.; Cambria, E. BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis. *Neurocomputing* **2022**, *467*, 73–82. [CrossRef]
46. Khan, W.; Ghazanfar, M.A.; Azam, M.A.; Karami, A.; Alyoubi, K.H.; Alfakeeh, A.S. Stock market prediction using machine learning classifiers and social media, news. *J. Ambient. Intell. Humaniz. Comput.* **2020**, 1–24. [CrossRef]

47. Team NLTK. NLTK Is a Suite of Libraries and Programs for Symbolic and Statistical Natural Language Processing for English Written in the Python Programming Language. 2001. Available online: https://www.nltk.org (accessed on 24 June 2021).
48. Smailović, J.; Grčar, M.; Lavrač, N.; Žnidaršič, M. Stream-based active learning for sentiment analysis in the financial domain. *Inf. Sci.* **2014**, *285*, 181–203. [CrossRef]
49. Risius, M.; Akolk, F.; Beck, R.L. Differential emotions and the stock market-the case of company-specific trading. In Proceedings of the Twenty-Third European Conference on Information Systems (ECIS), Munster, Germany, 26–29 May 2015.
50. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
51. Joshi, R.; Tekchandani, R. Comparative analysis of Twitter data using supervised classifiers. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; Volume 3, pp. 1–6.
52. Swinburne, R. Bayes' Theorem. *Rev. Philos. Fr.* **2004**, *194*, 2825–2830.
53. Lakshmi, V.; Harika, K.; Bavishya, H.; Harsha, C.S. Sentiment analysis of twitter data. *Int. Res. J. Eng. Technol.* **2017**, *4*, 2224–2227.
54. Xiao, Q.; Chang, X.; Zhang, X.; Liu, X. Multi-information spatial–temporal LSTM fusion continuous sign language neural machine translation. *IEEE Access* **2020**, *8*, 216718–216728. [CrossRef]
55. Karevan, Z.; Suykens, J.A.K. Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Netw.* **2020**, *125*, 1–9. [CrossRef] [PubMed]
56. Behera, R.K.; Jena, M.; Rath, S.K.; Misra, S. Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. *Inf. Process. Manag.* **2021**, *58*, 102435. [CrossRef]
57. Wang, Z.; Yan, W.; Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1578–1585.
58. Malley, T.; Bursztein, E.; Long, J.; Chollet, F.; Jin, H.; Invernizzi, L.KerasTuner Is an Easy-to-Use, Scalable Hyperparameter Optimization Framework That Solves the Pain Points of Hyperparameter Search. 2019. Available online: https://github.com/keras-team/keras-tuner (accessed on 24 July 2021).
59. Morris, G.L. *Candlestick Charting Explained: Timeless Techniques for Trading Stocks and Futures: Timeless Techniques for Trading Stocks and Sutures*; McGraw Hill Professional: New York City, NY, USA, 2006.
60. Nison, S. *Japanese Candlestick Charting Techniques: A Contemporary Guide to the Ancient Investment Techniques of the Far East*; Penguin: London, UK, 2001.
61. GNU General Public License. Snscrape a Scraper for Social Networking Services (SNS). 2020. Available online: https://github.com/JustAnotherArchivist/snscrape (accessed on 24 May 2021).
62. Joshua, R. Tweepy An Easy-to-Use Python Library for Accessing the Twitter API. 2009. Available online: https://www.tweepy.org/ (accessed on 24 May 2021).