A Deep Learning-Based LSTM for Stock Price Prediction Using Twitter Sentiment Analysis

Shimaa Ouf[1], Mona El Hawary[2]*, Amal Aboutabl[3], Sherif Adel[4]
Information Systems Department-Faculty of Commerce and Business Administration, Helwan University, Cairo, Egypt[1, 2]
Computer Science Department-Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt[3]
Administration Department-Faculty of Commerce and Business, Helwan University, Cairo, Egypt[4]

***Abstract*—Numerous economic, political, and social factors make stock price predictions challenging and unpredictable. This paper focuses on developing an artificial intelligence (AI) model for stock price prediction. The model utilizes LSTM and XGBoost techniques in three sectors: Apple, Google, and Tesla. It aims to detect the impact of combining sentiment analysis with historical data to see how much people's opinions can change the stock market. The proposed model computes sentiment scores using natural language processing (NLP) techniques and combines them with historical data based on Date. The RMSE, R^2, and MAE metrics are used to evaluate the performance of the proposed model. The integration of sentiment data has demonstrated a significant improvement and achieved a higher accuracy rate compared to historical data alone. This enhances the accuracy of the model and provides investors and the financial sector with valuable information and insights. XGBoost and LSTM demonstrated their effectiveness in stock price prediction; XGBoost outperformed the LSTM technique.**

***Keywords*—Sentiment analysis; stocks price prediction; correlation; natural language processing (NLP); machine learning model; LSTM; XGBoost**

I. Introduction

Stock market prediction has been a matter of interest to researchers, vendors, and investors for a long period. The main purpose of predicting the stock market is to achieve optimal results and decrease investment risk. It focuses on establishing an effective technique for predicting stock prices and providing well-informed, data-driven insights on market behavior [1]. Another purpose of stock market prediction is to predict the most accurate price in the future and determine the trend of the stock price, whether it is going up or down. This helps and guides investors to make better choices, avoiding potentially harmful investments that could increase their profits and reduce their losses [2]. Due to its volatility, unpredictability, and rapidness, stock market prediction is challenging. Various factors, such as public opinion, social media, feelings, and sentiments, influence the accuracy of stock market predictions [3].

Social media platforms (Facebook, Twitter, etc.) became a vital data source. Social media plays an important role for companies and people. People use social media every day to express their opinions and experiences and review products, services, or even companies [4]. On the other hand, companies utilize social media platforms to extract and analyze customers' opinions and feelings toward what they offer [5] [6]. Sentiment analysis (SA) is one of the disciplines that analyze social media data. Sentiment analysis can be defined as analyzing users' emotions using their opinions, sentiments, and subjective texts to decide if their interactions are positive, negative, or neutral. As stated, Sentiment Analysis is a type of subjectivity analysis that concentrates on identifying opinions, feelings, and respect conveyed through natural language [7]. It enhances the quality of goods and services by evaluating consumers' feedback on a certain product or service. Furthermore, natural language processing (NLP) is important for teaching machines to process human language and translate it into machine-readable format.

Sentiment analysis plays an important role in predicting stock prices by examining public opinion and social media to estimate the market mood and its effect on stock prices and aid the investors in overcoming the investment risk Merging sentiment analysis with traditional financial measures improves predictive accuracy. Companies can utilize risk management techniques to make strategic decisions, regardless of stock fluctuations [8].

Different domains have addressed sentiment analysis because it can help companies increase their capitalization by enhancing their products or services to meet customers' expectations. Stock markets leverage social media data, and SA can predict its stock prices depending on people's opinions. However, predicting stock prices is complicated because of their volatile and dynamic nature [9].

Historical data is another factor that plays an important role in stock price prediction. Historical data comprehends market behaviors and trends, leading to well-informed financial decisions based on data analysis. Examining historical data, which includes financial variables such as volume, price, high, low, and close, enables predictive algorithms to detect connections and recurrence patterns. Then train and test machine learning techniques using historical data to predict stock prices and improve their accuracy [10].

A. Background

1) The extreme gradient boosting (XGBoost) model: XGBoost in ML for regression and classification is an ensemble technique that builds a series of weak learners, usually decision trees (DTs). Every learner removes errors from the previous learners by minimizing errors between actual values and predicted values. This process is called the loss function [11]. XGBoost employs the L1 (Lasso) and L2 (Ridge) methods of regularization to reduce overfitting and improve the model's ability to generalize. Its parallel processing ability makes it computationally rapid and suitable for big data processing.

XGBoost works by figuring out an objective function in Eq. (1). Regularizes then use this function along with a loss function and a residual measurement to get rid of complicated models that cause overfitting [12]. In XGBoost, the objective function comprises a loss function that calculates the residuals and a second part that simplifies the model and reduces overfitting [13]. Model parameter optimization can reduce the objective function through gradient descent on the loss function. DTs persist in decreasing the residuals and repeatedly reduce the loss until it diminishes.

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \iota(\mathrm{y}_i, \hat{\mathrm{y}}_i) + \sum_{K=1}^{K} \Omega(f_{k)} \qquad (1)$$

Where $\mathcal{L}(\theta)$ is the objective function (overall), $\iota(\mathrm{y}_i, \hat{\mathrm{y}}_i)$– (evaluation metrics), or loss function was measured by (mean absolute error), $\Omega(f_{k)}$ Regularization term to minimize overfitting.

$$\hat{\mathrm{y}}_{\mathrm{i}}^{(\mathrm{t})} = \sum_{k=1}^{t} \mathrm{f}_k(\mathrm{x}_i) \qquad (2)$$

Prediction in XGBoost as shown in Eq. (2): At each stage, the forecast $\hat{\mathbf{y}}_{\mathbf{i}}$, is calculated by summing the predictions made by all (t) the trees and (f_k) the tree. The objective function decreases by using gradient descent to iteratively adjust model parameters and minimize the loss function [14]. During the optimization, the DTs aim to eliminate the residuals (the differences between actual and predicted values). This process continues until the algorithm reaches a minimum point where further tuning does not significantly reduce the loss.

2) XGBoost integration with sentiment analysis: XGBoost is a gradient-boosting technique that uses several decision trees to simulate the relationships between features, such as emotion scores and stock prices, and the target variable (the future stock price). Each decision tree works to make predictions better by using attributes and stock market dates repeatedly. Simultaneously, the loss function undergoes optimization to identify the least effective prediction trees [15]. XGBoost has integrated with sentiment analysis, specifically sentiment score or polarity, as one of its input features. XGBoost can manage nonlinear interactions and prioritize feature relevance, which is beneficial when combining textual sentiment analysis with historical data. XGBoost boasts several features that set it apart from the traditional model. XGBoost can define and determine how much the sentiment analysis score contributes to the prediction. Compared to stock features, regularization is another feature that enables XGBoost to overcome or prevent overfitting, particularly when the combined features have high dimensionality. Scalability is considered the most significant feature in XGBoost. It is used for large datasets. XGBoost can handle both historical and sentiment analysis [16].

3) Long short-term memory (LSTM): To overcome long-term dependency, a unique type of recurrent neural network was explicitly designed, namely LSTM. The LSTM incorporates a novel memory cell that replaces conventional artificial neurons in the hidden layers, enabling it to retain data over an extended duration. According to Hochreiter and Schmid Huber [17], the inability to handle long-term dependencies is an important factor for any dispute that contains a time series; this is the main struggle when applying ordinary neural network architecture to predict a stock price. Three gates organize the information, determining which data to store in the cell, which to discard, and what the cell's output will be. LSTM proves to be a confidential mechanism in many fields, not only computer science but also statistics, linguistics, and medicine. All these areas have tasks involving the analysis of sequential data, prediction, classification, and regression, for which LSTM proved to have a great ability to process them well [18].

4) Long short-term memory (LSTM) integrated with sentiment analysis: Long Short-Term Memory (LSTM), a type of deep learning, is considered the best technique for processing sentiment analysis and can manage and handle sequential data by capturing long-term dependencies within the text or sentence. This makes it suitable for sentiment analysis tasks, particularly when the word depends on the word in the surrounding context [19]. The LSTM architecture incorporates memory cells that store information related to stored sequences over extended periods. This enables the model to interpret the sentiment in the text more accurately and quickly. This is true even when there is a significant gap between the keywords. Sentiment analysis depends on data quality, such as the volume of text. LSTM can learn and forecast the next sentiment labeled more accurately based on the large volume of labeled text. This feature makes it an invaluable tool for other tasks, like monitoring social media and predicting stock prices [20].

This paper is organized as follows: Section II presents a literature review. Section III introduces the proposed methodology, while Section IV and Section V showcase experimental analysis and results respectively. Simultaneously, Section VI highlights the conclusions.

II. Literature Review

Many researchers have tackled improving prediction accuracy by using machine learning and deep learning algorithms. The goal of this paper is to review previous studies, find challenges, and overcome these gaps and challenges by providing a novel research method.

A study [21] employed LSTM regression models to estimate India's NIFTY 50 index. The deep learning-based LSTM model outperformed traditional machine learning methods. A study [22] compares ARIMA, LSTM, and BiLSTM models for forecasting financial time-series data and concludes that the BiLSTM model produced the best results. Another study in [23] offered an RNN-Boost model for forecasting Chinese stock market values, which outperformed the baseline RNN model.

The study in [24] employed deep learning, support vector regression, and linear regression for the stock market to forecast and evaluate the sentiment surrounding each event, focusing on four nations that represent established, emerging, and undeveloped economies: the United States, Hong Kong, Turkey, and Pakistan. This study assessed the system's performance using mean absolute error (MAE) and root mean square error

(RMSE). The results indicate that incorporating sentiment analysis for these events improves the system's performance.

This research paper aims to concentrate on the implementation of the stock-containing Long Short-Term Memory (LSTM) algorithms. The LSTM originates from the recurrent neural network in stock. It has a significant effect on time series data problems. This study establishes two models: the BP neural network model and the LSTM model. Next, integrate these models with the available stock data to generate a series of predictions. Undoubtedly, the prediction accuracy of LSTM models has improved. The accuracy rate can reach 60%–65%. During the modeling process, this study has refined traditional gradient descent algorithms and specifically designed the neural network's input data to mitigate the inevitable "sawtooth phenomenon" of the gradient descent algorithm. Additionally, they established a library of parameter combinations and utilized the dropout technique to achieve more accurate prediction results [25].

Machine learning models demonstrate that artificial neural networks (ANNs) can learn input-output correlations and assist in producing close estimates of daily closing prices when trained on the same data [26]. Deep learning techniques like convolutional neural network (CNN) was used in sentiment analysis research in the Indonesian language [27].

Many research studies use sentiment analysis to extract opinions from the text and categorize them as positive, negative, or neutral. Researchers can categorize the research into lexicon-based or machine-based learning-based methods. In recent years, many lexicons can be depended on and used to determine the text, including SentiWord Net [28], WordNet-Affect, and Sentic Net [29].

The author used the support vector machines for sentiment analysis. Experiments reveal an accuracy of 89.93% in predicting the direction of the SSE 50 index's movement, with an additional 18.6% increase in accuracy when adding sentiment-describing parameters. At the same time, this model supports investors in making better investment decisions [30].

In study [31], they create and evaluate forecasting models for stock prices and trends. They suggest a novel decision tree technique to predict stock performance by utilizing a large-scale sample of tweets relating to four companies: Apple, Google, Microsoft, and Netflix. They concluded that a decision tree model surpasses a multiple regression model.

In study [32] Deep learning models have been used to enhance the accuracy of stock price prediction. They depend on more than 265,000 news articles and S&P 500 companies' financial datasets to predict stock prices. They concluded that the RNN model performed well compared with other models (ARIMA and Facebook Prophet). Furthermore, RNN proved its efficiency with fixed/stable stocks rather than low prices.

In study [33] Applied machine learning historical stock prices and financial news from four distinct companies in different industries. They aim to ascertain the influence of financial news on the fluctuations in stock prices. They have also conducted experiments to evaluate the challenges associated with predicting certain stocks. This study discovered that of Tata Motors stock prediction, an automobile company, has the highest MAPE, resulting in a greater deviation from the actual prediction than other stocks.

The relationship between sentiment derived from financial news and tweets and the movements of the FTSE100 index is examined [34]. The investigation aimed to determine the strength of the correlation between sentiment indicators on a given day and market volatility and returns observed on a subsequent day. The experimental findings reveal evidence of a correlation between sentiment and stock market movements.

In study [35] Artificial Neural Networks (ANN) have been used to predict the prices in different periods one day, 7 days, 30 days, 60 days, and 90 days. They used different resources of social media datasets combined as input such as (Twitter, google Trends, web news, forum posts) and the Stock Exchange (GSE) from January 2010 to September 2019 of Ghana company dataset. They concluded that the information they obtained from social media can influence the effectiveness of stock price movement prediction.

The authors utilized deep learning models to track and evaluate the Chinese stock movements. They figured out that LSTM's forecast is closest to actual values, and increasing the number of hidden layers had no significant impact on accuracy [36].

The relationship between Twitter tweets and stock prices has been tested and proven using multiple models. While training the models, SVM reveals a superior performance. The neural network models proved their superiority while evaluating the difference in the closing stock prices between the two companies [37].

After reviewing the literature, we found that no study measures the correlation between SA and stock price prediction. Therefore, this correlation is represented by analyzing the performance of integrating the sentiment with historical data compared with using historical data only.

Therefore, this research shed light on this gap by developing an AI model to improve stock price prediction accuracy. It provides investors with more insight to make informed and valuable decisions regarding buying or selling. The proposed model depends on applying the LSTM and XGBoost techniques with historical data only and with sentiment analysis integrated with historical data. The integration of sentiment analysis (qualitative data) with historical data can improve prediction accuracy and yield valuable market insights.

III. Research Methodology

The research methodology involves utilizing natural language processing (NLP) techniques to extract sentimental information from social media as represented in Fig. 1. This sentiment analysis, categorized as positive, negative, or neutral, is then combined with the corresponding stock prices retrieved from Yahoo Finance. The existence of a correlation between sentiment analysis and stock prices is demonstrated using machine learning and deep learning models. Furthermore, two models, including LSTM (Long Short-Term Memory) and XGBoost, are utilized to enhance prediction accuracy based on sentiment analysis. These models are applied with historical data only and with sentiment analysis integrated with the historical

data. These models are measured through evaluation metrics such as Root Mean Squared Error (RMSE), R^2, and Mean Absolute Percentage Error (MAE). This experiment helps to measure the impact of integrating sentiment analysis and provides valuable insights into where to buy or sell in the financial sectors.

IV. Experimental Analysis

A. Dataset Description

This study utilizes three stocks captured from Kaggle: Dataset 1 collects people's tweets for Apple from June 2015 to December 2019. Dataset 2 collects tweets from June 2015 to September 2020 for Google. The final dataset gathers tweets from June 2020 to October 2020 for Tesla. Each dataset comprises two columns: the first column is the date of the post date, while the second column contains the tweet's text. The quantitative data type in this study necessitates a sentiment analysis. Then, other datasets were downloaded from Yahoo Finance, where the stock prices for Apple, Google, and Tesla span the same period. Each dataset consists of seven columns, 1) The date: marks the recording or reporting of the stock market data. 2) Open: The opening price of the company's stock on the given date. 3) High: The highest price reached during the trading day. The company's stock traded at its highest price. It indicates the highest price reached during the trading session. 4) Low: The trading day reached its lowest price. 5) Close: The price at which the stock closed on that trading day. 6) Adj Close refers to the adjusted closing price, which considers any corporate actions such as stock splits or dividends that may impact the stock's value. 7) Volume refers to the total number of shares traded during a trading day.

B. The Experimental Approach

This section discusses the experiments conducted to measure the effect of people's sentiments on total fluctuation. The two main subsections below provide detailed steps. The first subsection illustrates the data preprocessing. The second clarifies the experimental steps.

1) Data preprocessing: The first task in the preprocessing stage is to detect duplicated records and null values. Drop the duplicate records and use the mean imputation to address the missing value problem. Furthermore, the uppercase letters are converted to lowercase letters to streamline the training and testing of the model. The data cleaning task included the removal of all stop words, such as "that," "for," "the," "a," "he," and "has." Furthermore, the lemmatizing process is conducted on the previously mentioned datasets., an NLP tool allows end users to understand full sentence input from end users. Also, URLs, mentions, and # hashtags were removed in the preprocessing phase.

2) The experimental steps: This study applies the Long Short-Term Memory (LSTM) and XGBoost algorithms to detect the relationship between people's sentiment and the total fluctuation to predict the future stock price.

After cleaning the data, the polarity measure is conducted, utilizing two algorithms, such as VADER [38] From NLTK [39], which employs tools to identify positive, negative, or neutral comments and compound sentiment. VADER evaluates the polarity, or the intensity of the emotion, by determining whether the statement is positive, negative, or neutral as introduced in Fig. 2. Text Blob NLP evaluates the subjectivity and polarity of Twitter tweets.

Polarity is the positive, negative, or neutral feeling expressed in a text as represented in Fig. 3. It determines whether a text describes a good or negative attitude toward a particular entity or issue. NLP frequently uses polarity analysis for sentiment analysis, identifying and categorizing opinions expressed in text. on the other hand, Subjectivity hand describes how subjective or objective a piece of literature is. It is used to determine if a statement is a fact or opinion. Subjectivity analysis in NLP allows detective subjective language and its distinction from objective language. Subjectivity analysis is used to assess whether a text is objective or subjective [40].

The sentiment score of a piece of text (X tweet) can be calculated using various methods as presented in Eq. (3). A simple approach might be used to assign scores to positive and negative words and compute an overall score for the document [46].

$$S = \frac{(\Sigma p_l') - (\Sigma n_J')}{n_p + n_n} \quad (3)$$

Where S is the sentiment score, p_l' are the scores of positive words, n_j are the scores of negative words, n_p is the number of positive words, and n_n is the number of negative words in the text. Market volatility on a given day can be modeled as a function of sentiment scores from the previous day, incorporating both social media sentiment and traditional financial indicators [41].

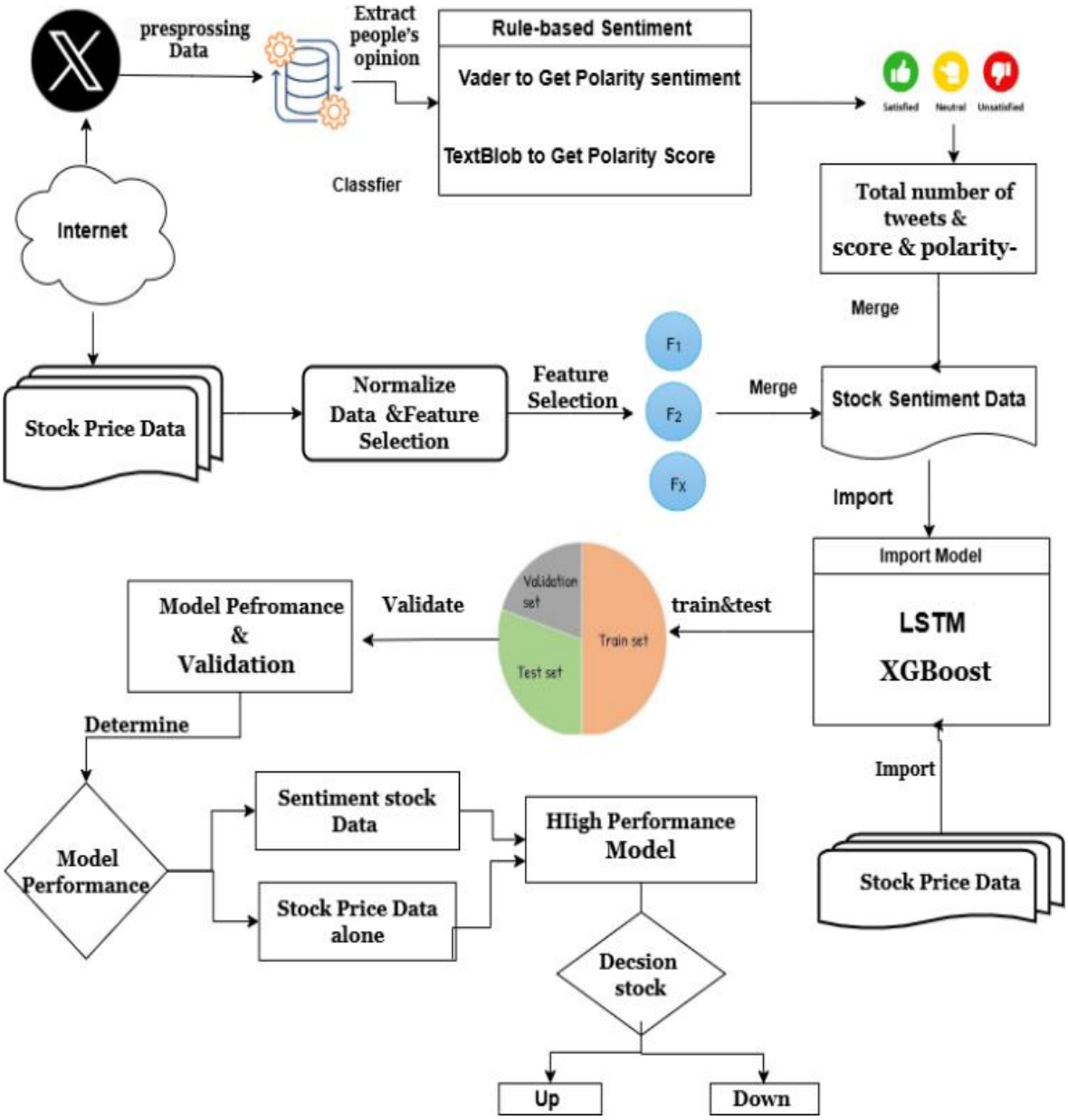

Fig. 1. The framework of stock prediction based on sentiment analysis.

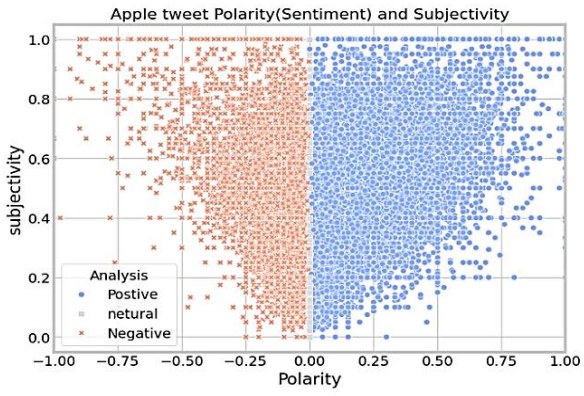

Fig. 2. Polarity vs. subjectivity.

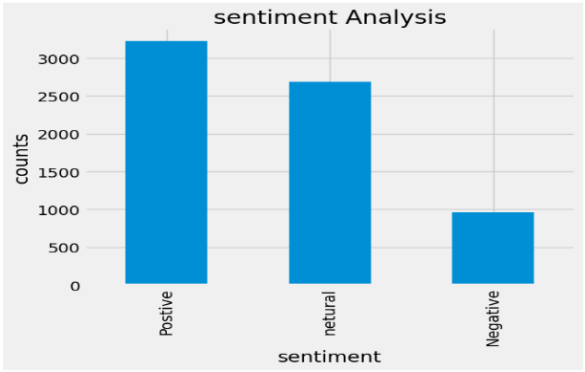

Fig. 3. Classifier of Apple tweet.

Fig. 4. The most positive words.

Fig. 4 illustrates the most common and frequent words in positive tweets about Apple, including “new”, “good”, and “read”, “Thanks”, which are strongly associated with positive sentiments.

Fig. 5. The most negative words.

Fig. 5 illustrates the most common and frequent words in negative tweets about Apple, such as “hate”, “worst”, “sorry”, “fail”, and “stupid”, and “stupid”, which are associated with negative sentiment.

The most common and frequent words in positive tweets about Google, including “top”, “share”, and “competition”, “penny”, are strongly associated with positive sentiments.

The most common and frequent words in negative tweets about Google, including “worst”, “pain”, and “less”, “spam”, are strongly associated with negative sentiments.

The most common and frequent words in positive tweets about Tesla, including “best”, “trade”, and “right”, “good”, are strongly associated with positive sentiments.

The most common and frequent words in negative tweets about Tesla, including “bad”, “wrong”, and “crazy”, “tesla”, are strongly associated with negative sentiments.

Three new features were added to the Apple, Google, and Tesla stock price datasets to evaluate the efficiency of future stock values. The first feature is total price fluctuation, which indicates the overall variation at the end of the day. This is measured by subtracting the volume value from the close value. The second feature, price difference is calculated by subtracting today's adjusted close value from the value from the previous day. Depending on the price difference feature, the direction of the stock price (Up or Down) is detected and stored in the trend score feature.

The price difference feature detects and stores the stock price's direction (up or down) in the trend score, which is the last feature. If the price difference is positive, the trend score is equal to 1 (up), and if the price difference is negative, the trend score is 0 (down), as illustrated in the snapshot of Table I.

TABLE I. SNAPCHAT ADDED THREE FEATURES TO THE STOCK PRICE

Date	Open	High	Low	Close	Adj Close	Volume	Total_price_fluctuation	Price_difference	Trend
2015-01-06	26.635000	26.857500	26.157499	26.56001	23.779427	263188400	6.991600e+09	0.002501	1
2015-01-07	26.799999	27.049999	26.67499	26.937500	24.112865	160423600	4.321411e+09	0.372499	1
2015-01-08	27.307501	28.037500	27.174999	27.972500	25.039347	237458000	6.642294e+09	1.035000	1
2015-01-09	28.167500	28.312500	27.552500	28.002501	25.066191	214798000	6.014881e+09	0.030001	1
2015-01-12	28.150000	28.157499	27.200001	27.312500	24.448545	198603200	5.424350e+09	-0.690001	0
2015-01-13	27.857500	28.200001	27.227501	27.555000	24.665621	268367600	7.394869e+09	0.242500	1
2015-01-14	27.260000	27.622499	27.125000	27.450001	24.571625	195826400	5.375435e+09	-0.104999	0
2015-01-15	27.500000	27.514999	26.665001	26.705000	23.904747	240056000	6.410695e+09	-0.745001	0
2015-01-16	26.757500	26.895000	26.299999	26.497499	23.719000	314053200	8.321624e+09	-0.207501	0
2015-01-20	26.959999	27.242500	26.625000	27.180000	24.329939	199599600	5.425117e+09	0.682501	1
2015-01-21	27.237499	27.764999	27.067499	27.387501	24.515686	194303600	5.321490e+09	0.207501	1
2015-01-22	27.565001	28.117500	27.430000	28.100000	25.153471	215185600	6.046715e+09	0.712499	1

The sentiment scores, polarity, and total number of **X** tweets are combined with the updated Apple, Google, and Tesla stock price datasets by matching dates to check for a correlation between people's opinions and the total fluctuation.

The first experiment tests the correlation to determine the extent to which people's opinions influence the overall fluctuation. The second experiment is to predict the expected stock price using the LSTM and XGBoost algorithms. The performance of the used algorithms is evaluated and ranked.

a) The first experiment: Fig. 6 and Fig. 7 depict the movements of both positive and negative tweets about the stock price. Depending on Fig. 6, it is obvious that the positive tweet movements pretty much match the stock price movements. However, the movement of negative tweets barely relates to the movements of the stock price. Thus, generally, we can deduce that there is a relationship between the tweets and stock market movements.

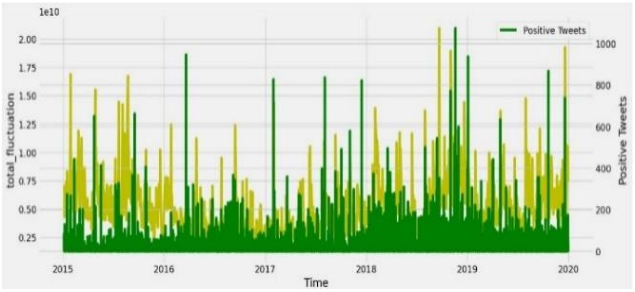

Fig. 6. Positive tweet movement.

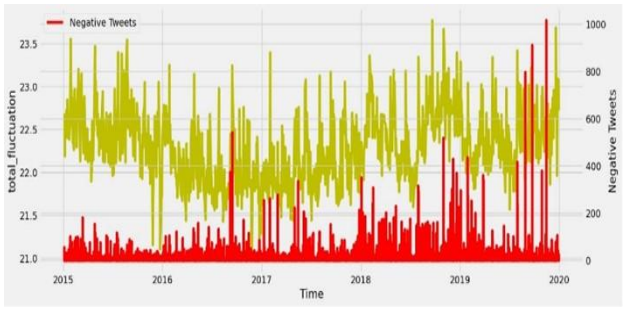

Fig. 7. Negative tweet movement.

To figure out to what extent the tweets can affect the stock price, the second experiment is conducted. The results show that positive people's opinions can affect the stock price by 47.56%. However, a negative correlation of 4.9% exists between people's negative tweets and the stock price. This means that when the tweets are negative, the stock price increases.

b) Predict stock prices utilizing LSTM and XGBoost models by integrating sentiment analysis: This approach utilizes the Long Short-Term Memory (LSTM) and XGBoost algorithms on a combined dataset, which includes the total number of tweets, polarity, and sentiment score, as input to predict the future stock prices of Apple, Google, and Tesla. Then take the classifier tweet, which can be positive, negative, or neutral, and calculate the average polarity and total number of tweets which sum up the number of tweets per day, then merge it with the closing price. as presented in Fig. 8.

Fig. 8. The number of tweets per day vs. polarity vs. close price.

To predict the stock price, the rolling window approach is applied to create numerous overlapping training samples through sliding fixed-size windows, which not only helps to improve the model's ability to observe the data but also aids in obtaining multiple patterns in the data. This approach is considered powerful and most suitable for LSTM and XGBoost prediction. The models analyze a tweet from **X** Twitter and predict future movements of the adjusted close price by referencing past days. The models normalized the input features (polarity score, total number of tweets, and close price) within the range of [0] to [1] as represented in Eq. (4).

$$CLSOE\ PRICE(x)normalize = \frac{x - min(X)}{max(X) - min(X)} \quad (4)$$

Then, the sentiment analysis features are integrated with historical data to figure a single dataset. And can represent this combined dataset as a shape (s), where s denotes the polarity score, (V) the total number of tweets, (X) the close price, (n) the end window size (y^t) the predicted value as introduced in Eq. (5) and Eq. (6). The system utilizes fixed-size windows with a length of n to generate a combined input for each time step (t). The last n-time step uses the close price and the polarity score (combined dataset) as input.

$$y^t = LSTM\ [x_t - n + 1, x_t - n + 2, \ldots., x_t,\ v_t - n + 1, v_t - n + 2, \ldots., v_t, s_t - n + 1, s_t - n + 2, , \ldots., s_t\ ] \quad (5)$$

Set the window size to three days of the combined dataset for feature engineering, which will serve as training to predict the next day. For instance, window 1 uses data from day 1 to day 3 (features) to forecast the next day. So, the predicted value.

$$y_{4=(}x_1, x_2, x_3\ )(\ v_1, v_2, v_3)\ (s_1, s_2, s_3) \quad (6)$$

The datasets are divided into a 90% training set and a 10% test set. Fig. 9 illustrates the division of the 90% training set into 80% training and 10% validation sets, utilizing the years 2015–2019 for training and 2019 for validation in the first dataset, utilizing the years 2015–2020 for training and 2020 for validation in the second dataset, and utilizing the June to October 2020 for training and 2020 for validation in the third dataset. Build the LSTM and XGBoost models by feeding it with training data. The LSTM models determine the input layer to obtain the sequence of previous time steps, which consists of three layers. The first layer, known as the input layer, receives window data in the form of a triangle (3, 3). The first (3) indicates the number of windows, while the second (3) indicates the number of three features (polarity scores, total number of tweets, and adjusted closing prices). The third layer is

responsible for processing time series data from previous observations of the rolling windows.

50 units make up the second layer, known as the LSTM layer, which uses gates to learn from time-dependent patterns in the data. Additionally, the design of memory cells, which also consist of 50 units, allows for the sequential data capture LSTM layer, comprising 50 units, to employ gates to learn from time-dependent patterns in the data. In addition to that, memory cells, which also consist of 50 units, process sequential data and capture temporal dependencies by adding more LSTM layers. Put Layer: one Dense Layer 1. The layer's design forecasts a single value (adjusted close price) for the upcoming time step. Then compile the model using the Adam optimizer. LSTM model is trained on data for a batch size of 16, which aids in obtaining single patterns by updating the model's weight after handling each sample. The number of epochs is assigned to 20 which indicates the number of loops in the training data.

The model is validated by monitoring while training the data, tracking the model's performance, adjusting the hyperparameters to enhance the performance, predicting the future stock price, and assessing the model using metrics like RMSE, R-squared, and MAE as illustrated in Eq. (7), Eq. (8) and Eq. (9).

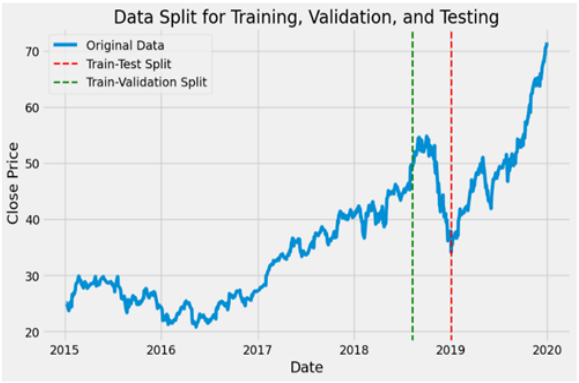

Fig. 9. The data split for sentiment analysis.

$$RMSE = \frac{1}{N}\sum_{i=1}^{n}(\text{yi} - \hat{\text{yi}})^2 \quad (7)$$

In machine learning, the R-squared value represents the coefficient of determination or the coefficient of multiple decisions in the context of multiple regression. Regression uses R squared as an evaluation metric to assess the scatter of data points around the fitted regression line. It indicates the percentage of variation in the dependent variable [42].

$$Squared\ Error = (\text{yi} - \hat{\text{yi}})2 \quad (8)$$

The proportion of the dependent variable's variance that the independent variable can explain is known as R-squared. R2 = the variance explained by the model. Total Variance The R-squared value stays between 0 and 100%. 0% corresponds to a model that fails to explain the variability of the response data around its meaning [43].

Mean Absolute Error (MAE) utilizing evaluation metrics, which indicates the difference between the actual and predicted value (value which is expected through a Model where actual values (target variable) [44] can be calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (9)$$

Predict Stock Price Using the LSTM and XGBoost Models Based on Historical Data

This study employs the LSTM and XGBoost models to predict future fluctuations in Apple, Google, and Tesla stock prices, relying solely on historical data. The data is gathered and preprocessed to evaluate the effectiveness of LSTM and XGBoost models, involves preparing the data for processing, which includes checking for any missing values by determining their meaning, removing any duplicated rows from the data, converting date columns to date-to-date time type, setting normalization values between [0:1], and dividing the datasets into three sets. The first set is for model training, the second for model validation, and the third for model testing. The training dataset (for The Model Training) makes up 70% of the total data; the validation dataset makes up 15%; and the testing dataset (for The Model Testing) makes up 15%. The model training uses the input sequences for LSTM. Each input sequence considers the output for the next step. Then set up the close price (target) as the preceding time step to predict the subsequent step to the close price. The regression approach is applied to the close price of Apple, Google, and Tesla stock. A single parameter, window size, determines the input format (close). The window size parameter establishes the days considered "dependent" for the stock price prediction. For example, with a window size of t = 60, the prediction will include the 60 days preceding day d. The longer the window, the better the model's view of previous data, but the higher the computing cost. Once we compute all sequence data, we organize each into a table of input features (X) and output targets (Y), which the model training process uses to adjust the LSTM model hyperparameters. The LSTM model consists of three layers: an input, an LSTM layer, and an output layer. Set the number of units to 50, the number of epochs to 100, and the batch size to 32. Then compile the model using the Adam optimizer and evaluate its performance using root means squared error (RMSE), R-squared, and MAE.

The same steps that were applied to evaluate the LSTM efficiency are used with XGBoost. Grid Search, Random Search, Max_Depth, Estimators, Sub-Sample, and L2 Regularization are used as hyperparameters.

V. Experimental Results

This section demonstrates the results obtained from two experiments. To detect the impact of combining sentiment analysis with historical data compared to historical data only. The previously mentioned algorithms are used to predict Apple, Google, and Tesla stock prices. As presented in Table II and figures from 10 -15, predicting stock prices based on integrating sentiment Analysis with historical data surpasses the performance of using historical data as presented in Table III and figures from 16 -21.

Fig. 10 displays the stock price prediction using XGBOOST by integrating Sentiment Analysis with historical data for Apple's stock, The model achieves a high accuracy rate of 99% from June 2017 to June 2020. It also achieves a low error rate between the actual and the predicted values for RMSE and MAE.

Fig. 11 predicted stock prices using XGBoost, which integrates sentiment analysis with historical data from 2018-7 to 2020-1 for Google. The model achieves a high accuracy of 95%, indicating a high correlation between actual and predicted values, and achieves a low prediction error of 0.0479 RMSE and 0.0358 with MAE, indicating a good fit.

TABLE II. MODELS PERFORMANCE-BASED INTEGRATION OF SENTIMENT ANALYSIS WITH HISTORICAL DATA

Model	APPLE	TESLA	GOOGLE
LSTM	R^2=91% RMSE=0.065 MAE=0.06	R^2=73% RMSE=0.1508 MAE=0.12	R^2=88% RMSE=0.0751 MAE=0.06
XGBoost	R^2=99% RMSE=0.042 MAE=0.0021	R^2=88% RMSE=0.1005 MAE=0.0816	R^2=95% RMSE=0.0479 MAE=0.0350

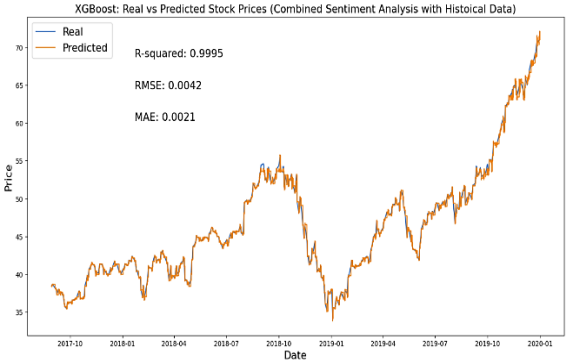

Fig. 10. XGBoost performance for Apple stock.

TABLE III. MODELS PERFORMANCE BY USING HISTORICAL DATA ALONE

Model name	APPLE	TESLA	GOOGLE
LSTM	R^2=98% RMSE=1.98 MAE=2.57	R^2=55% RMSE=0.45 MAE=6.94	R^2=95% RMSE=0.21 MAE=0.61
XGBoost	R^2=92% RMSE=1.85 MAE=1.29	R^2=99% RMSE=1.30 MAE=0.95	R^2=92% RMSE=1.85 MAE=1.29

Fig. 12 shows the stock price prediction using XGBOOST by integrating sentiment analysis with historical data for Tesla stock. The model's performance was 88.57%, which indicates a lower accuracy compared to another company's stock. It also achieved low error metrics, with RMSE (Root Mean Squared Error) at 0.1005, and MAE (Mean Absolute Error) at 0.0816 which indicates the difference between the predicted and actual values.

Fig. 13 Comparison of predicted stock prices using the LSTM Model, which integrates sentiment analysis with historical data for the Apple Company from 2017 to 2020. LSTM can capture the overall stock trend, but there are some gaps or errors in 2018 during high volatility.

Fig. 14 displays the integration of sentiment analysis with historical data for Google stock using the LSTM model. This model achieved good accuracy, at 88%. The model achieved a mean absolute error (MAE) of 0.06, indicating a low average error between the actual and the predicted values. And recorded in Root Mean Squared Error (RMSE): 0.0751.

Fig. 15 illustrates the comparison of predicted stock prices using the LSTM Model, which integrates sentiment analysis with historical data for the TSLA Company. The model achieves an accuracy of 75.48%, and achieves an RMSE of 0.1508, and an MAE of 0.12. This graph indicates the difference between the actual and predicted values. Often the predicted value fails to fit with the actual value.

Fig. 16 illustrates the performance of the XGBoost Model in predicting Apple's stock price. The model achieves an accurate rate of 92% with an average error of 1.29. These results declare a strong correlation between actual and predicted values.

Fig. 17 illustrates the performance of the LSTM Model for predicting Apple's stock price. The model achieves an accurate rate of 97% with an average error of 1.17 in RMSE and MAE = 1.27.

Fig. 18 compares the actual and predicted values of Google stock, using the XGBoost algorithm. The model achieved high accuracy of 92%, which indicates a strong relationship between actual and predictive values. XGBoost can closely track actual price movement while keeping low error metrics.

Fig. 19 displays Google's actual and predicted values, utilizing an LSTM (long short-term memory). LSTM has a high performance and accuracy of 95% and low error (RMSE = 0.21, MAE = 0.61). It appears that the predicted value is very close to the actual value, indicating that this model is a reliable tool for forecasting.

Fig. 20 compares Tesla's actual and predicted values, utilizing the XGBoost Model. The model achieves an accuracy of 99%, indicating a high performance and good fit between prediction and actual value. It achieves a low error metric of 1.30 of RMSE to 0.95 in MAE, the predicted value's deviation from the exact value.

Fig. 21 displays Tesla's actual and predicted values, utilizing an LSTM (long short-term memory). The model achieves an accuracy of 53%, indicating that its performance is not as high as it could be and requires additional data to improve. This is due to the insufficient training data set, which was from 01-08-2022 to 01-10-2022.

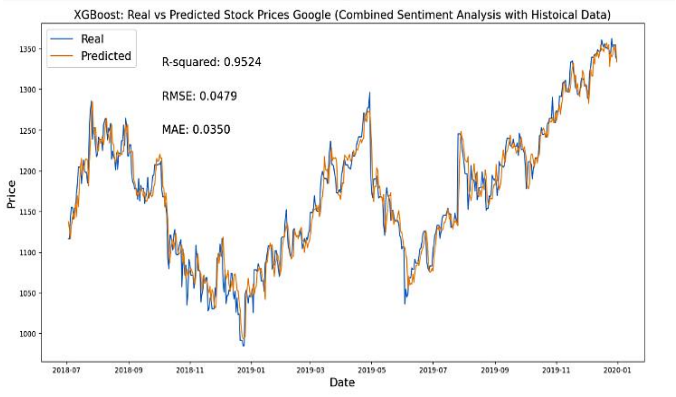

Fig. 11. XGBoost performance for Google stock.

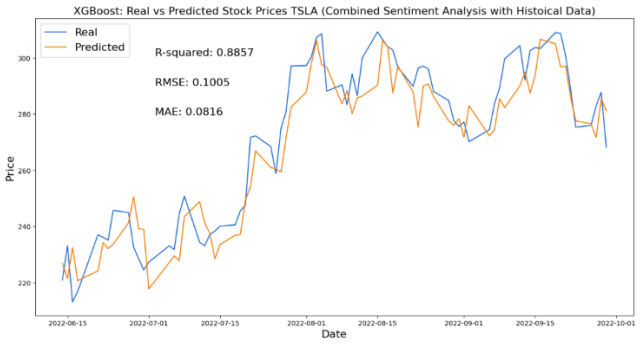

Fig. 12. XGBoost performance for Tesla stock.

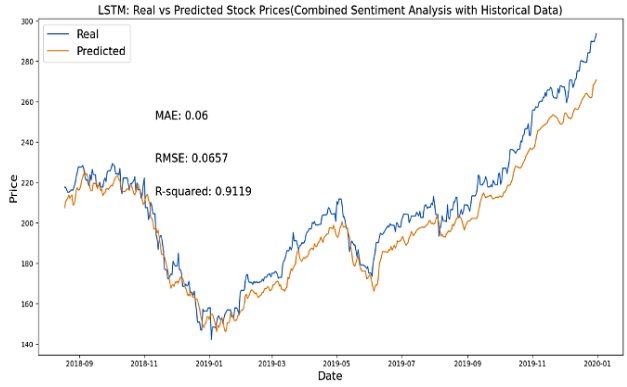

Fig. 13. LSTM performance for Apple stock.

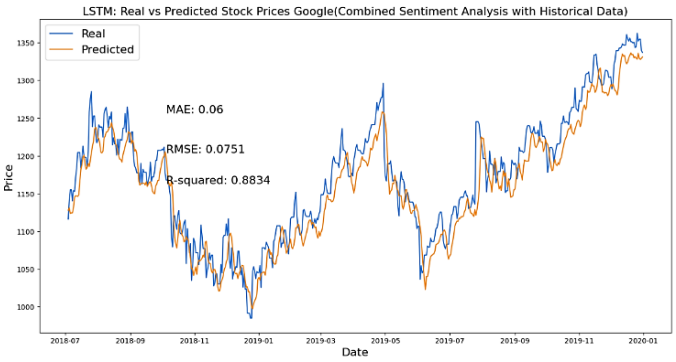

Fig. 14. LSTM performance for Google stock.

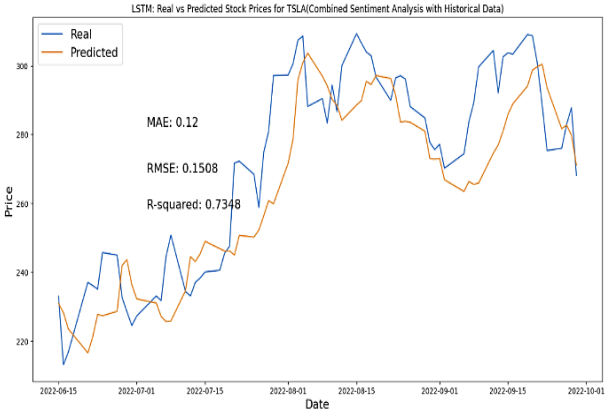

Fig. 15. LSTM performance for Tesla stock.

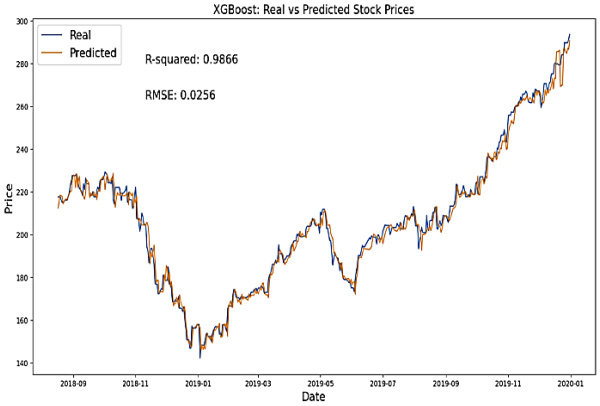

Fig. 16. Performance of XGBoost for Apple stock.

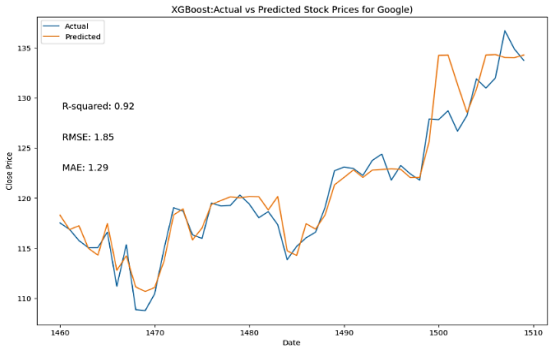

Fig. 17. Performance of XGBoost for Google stock.

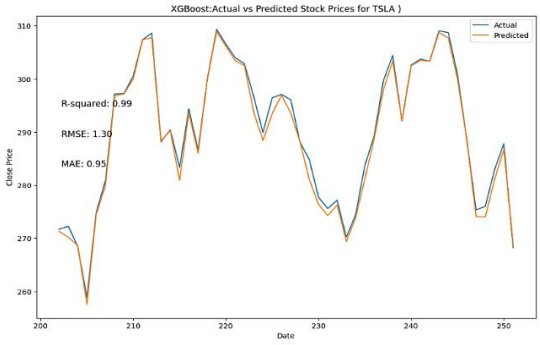

Fig. 18. Performance of XGBoost for Tesla stock.

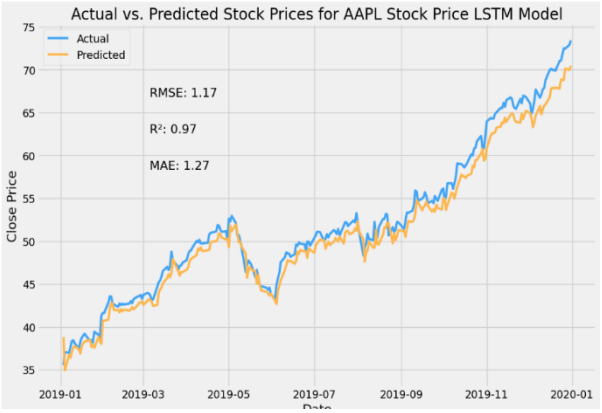

Fig. 19. Performance of LSTM for Apple stock.

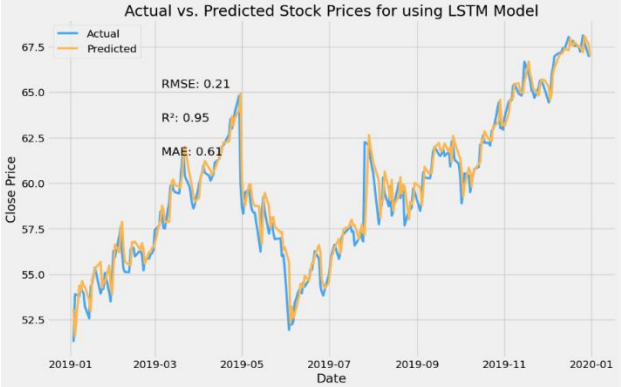

Fig. 20. Performance of LSTM for Google stock.

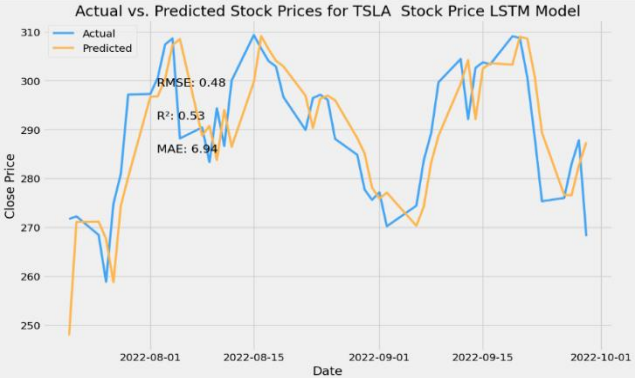

Fig. 21. Performance of LSTM for Tesla stock.

VI. Conclusion

This study proposed a framework to predict stock prices through integrated sentiment analysis with historical data, compared to using historical data only. Natural language processing (NLP) techniques extract sentimental information from X tweets. Then combine this sentiment analysis (positive, negative, or neutral) with the corresponding stock prices retrieved from Yahoo Finance. Machine learning and deep learning models test the combination of three stock datasets (Apple, Google, and Tesla) with and without sentiment analysis. This allows us to observe how integrating sentiment analysis with historical data correlates with each other and to what extent it improves performance accuracy. This study proved the correlation between stock price movement and people's opinions on social media. Two different classifiers are applied to predict the stock price, XGBoost and the LSTM models. The XGBoost outperformed the LSTM models in Apple, Google, and Tesla, achieving 99%, 95%, and 88%, respectively, by integrating sentiment analysis with historical data. On the other hand, LSTM outperformed XGBoost in Apple and Google, achieving 98% and 95%, except for Tesla, due to low data training when using historical data.

This study demonstrated the significance of combining and analyzing people's opinions, which is qualitative data that contributes to improving the understanding of integrating opinions with historical data, which is quantitative data. Both LSTM and XGBoost have demonstrated their efficacy.

References

[1] T. Julian, T. Devrison, V. Anora and K. M. Suryaningrum, "Stock Price Prediction Model Using Deep Learning Optimization Based on Technical Analysis Indicators," Elsevier Procedia Computer Science, vol. 227, pp. 939--947, 2023.

[2] K. Pawar, R. S. Jalem and V. Tiwari, "Stock price prediction based on deep neural networks," in Springer Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS, 2019.

[3] Z. Wang, S.-B. Ho and Z. Lin, "Stock market prediction analysis by incorporating social and news opinion and sentiment," in IEEE International Conference on Data Mining Workshops (ICDMW), 2018.

[4] H.-T. Duong and T.-A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," Springer Computational Social Networks, vol. 8, 2021.

[5] S. Hamed, M. Ezzat and H. Hefny, "A review of sentiment analysis techniques," International Journal of Computer Applications, vol. 176, pp. 20-24, 2020.

[6] Z. Drus and H. Khalid, "Sentiment analysis in social media and its application: Systematic literature review," Elsevier Procedia Computer Science, vol. 161, pp. 707-714, 2019.

[7] B. Ramalho, J. Jorge and S. Gama, "Representing uncertainty through sentiment and stance visualizations: A survey," Elsevier Graphical Models, vol. 129, p. 101191, 2023.

[8] Z. Wang, S.-B. Ho and Z. Lin, "Stock market prediction analysis by incorporating social and news opinion and sentiment," IEEE International Conference on Data Mining Workshops (ICDMW), pp. 1375-1380, 2018.

[9] A. Bhardwaj, Y. Narayan and M. Dutta, "Sentiment analysis for Indian stock market prediction using Sensex and nifty," Elsevier Procedia computer science, vol. 70, pp. 85-91, 2015.

[10] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," Neural Computing and Applications, vol. 32, pp. 1609--1628, 2020.

[11] N. Ghatasheh, I. Altaharwa and K. Aldebei, "Modified genetic algorithm for feature selection and hyper parameter optimization: case of XGBoost in spam prediction," IEEE Access, pp. 84365-84383.

[12] T. Liwei, F. Li, S. Yu and G. Yuankai, "Forecast of LSTM-XGBoost in Stock Price Based on Bayesian Optimization," Intelligent Automation & Soft Computing, vol. 29, pp. 855-868, 2021.

[13] H. Z. Wei Chen, Mukesh Kumar Mehlawat and Lifen Jia, "Mean–variance portfolio optimization using machine learning-based stock price prediction," Applied Soft ComputingApplied Soft Computing, vol. 100, p. 106943, 2021.

[14] Q. Wang, Y. Ma, K. Zhao and Y. Tian, "A comprehensive survey of loss functions in machine learning," Annals of Data Science, pp. 1-26, 2020.

[15] F. Balaneji and D. Maringer, "Applying Sentiment Analysis, Topic Modeling, and XGBoost to Classify Implied Volatility," in Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr), 2022.

[16] F. Balaneji and D. Maringer, "Applying Sentiment Analysis, Topic Modeling, and XGBoost to Classify Implied Volatility," in Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr), 2022.

[17] S. Selvin and R. Vinayakumar, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in international conference on advances in computing, communications and informatics (icacci), 2017.

[18] Q. M. Abdul, K. Sanjit, J. A. Chris, Arun Kumar Sivaraman, S. H. Kong Fah Tee and Janakiraman N, "Novel optimization approach for stock price forecasting using multi-layered sequential LSTM," Applied Soft Computing, vol. 134, p. 109830, 2023.

[19] U. D. Gandhi, P. Malarvizhi Kumar, G. Chandra Babu and G. Karthick, "Sentiment Analysis on Twitter Data by Using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM)," Wireless Personal Communications, 2021.

[20] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang and S. Qiao, "Attention-emotion-enhanced convolutional LSTM for sentiment analysis," transactions on neural networks and learning systems, vol. 33, pp. 4332-4345, 2021.

[21] S. Mehtab, J. Sen, A. Dutta, S. M. Thampi, S. Piramuthu, K.-C. Li, S. Berretti, M. Wozniak and D. Singh, "Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models," in Machine Learning and Metaheuristics Algorithms, and Applications, 2021.

[22] A. Chennupati, B. Prahas, B. A. Ghali, B. D. Jasvitha and K. Murali, "Comparative Analysis of Bitcoin Price Prediction Models: LSTM, BiLSTM, ARIMA and Transformer," pp. 1--7, 2024.

[23] W. Chen, C. K. Yeo, C. T. Lau and B. S. Lee, "Leveraging social media news to predict stock index movement using RNN-boost," Data & Knowledge Engineering, pp. 14-24, 2018.

[24] H. Maqsood, I. Mehmood, M. Maqsood, M. Yasir, S. Afzal, F. Aadil, M. M. Selim and K. Muhammad, "A local and global event sentiment based efficient stock exchange forecasting using deep learning," International Journal of Information Management, vol. 50, pp. 432-451, 2020.

[25] Y. Wang, Y. Liu, M. Wang and R. Liu, "LSTM model optimization on stock price forecasting," in 17th international symposium on distributed computing and applications for business engineering and science (dcabes), 2018.

[26] M. Qasem, R. Thulasiram and P. Thulasiram, "Twitter sentiment classification using machine learning techniques for stock markets," in International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015.

[27] M. F. Bashri and R. Kusumaningrum, "Sentiment analysis using Latent Dirichlet Allocation and topic polarity wordcloud visualization," in 5th International Conference on Information and Communication Technology (ICoIC7), 2017.

[28] M. Fikri and R. Sarno, "A comparative study of sentiment analysis using SVM and SentiWordNet," Indonesian Journal of Electrical Engineering and Computer Science, vol. 13, pp. 902-909, 2019.

[29] A. Segura Navarrete, C. Vidal-Castro, C. Rubio-Manzano and C. Martínez-Araneda, "The role of WordNet similarity in the affective analysis pipeline," Computación y Sistemas, vol. 23, pp. 1021-1031, 10 2019.

[30] R. Ren, D. D. Wu and T. Liu, "Forecasting stock market movement direction using sentiment analysis and support vector machine," IEEE Systems Journal, vol. 13, pp. 760-770, 2018.

[31] R. Chen and R. Dong, "The Relationship Between Twitter Sentiment and Stock Performance: A Decision Tree Approach," 2023.

[32] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia and D. C. Anastasiu, "Stock price prediction using news sentiment analysis," in fifth international conference on big data computing service and applications (BigDataService), 2019.

[33] J. Maqbool, P. Aggarwal, R. Kaur, A. Mittal and I. A. Ganaie, "Stock prediction by integrating sentiment scores of financial news and MLP-regressor: A machine learning approach," Procedia Computer Science, vol. 218, pp. 1067-1078, 2023.

[34] J. Deveikyte, H. Geman, C. Piccari and A. Provetti, "A sentiment analysis approach to the prediction of market volatility," Frontiers in Artificial Intelligence, vol. 5, p. 836809, 2022.

[35] I. K. Nti, A. F. Adekoya and B. A. Weyori, "Predicting stock market price movement using sentiment analysis: Evidence from Ghana," Applied Computer Systems, vol. 25, pp. 33-42, 2020.

[36] J. Long, Z. Chen, W. He, T. Wu and J. Ren, "An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market," Applied Soft Computing, vol. 91, p. 106205, 2020.

[37] S. V. Kolasani and R. Assaf, "Predicting stock movement using sentiment analysis of Twitter feed with neural networks," Journal of Data Analysis and Information Processing, vol. 8, pp. 309-319, 2020.

[38] A. Oad, I. Koondhar, P. Butt, M. Ahmed and S. Bhutto, "VADER sentiment analysis without and with English punctuation marks," Int. J. Adv. Trends Comput. Sci. Eng, vol. 10, pp. 1483--1488, 2021.

[39] M. Isnan, G. N. Elwirehardja and B. Pardamean, "Sentiment analysis for TikTok review using VADER sentiment and SVM model," Procedia Computer Science, vol. 227, pp. 168-175, 2023.

[40] E. Rosenberg, C. Tarazona, F. Mallor, H. Eivazi, D. Pastor-Escuredo, F. Fuso-Nerini and R. Vinuesa, "Sentiment analysis on Twitter data towards climate action," Results in Engineering, vol. 19, p. 101287, 2023.

[41] G. Murthy, S. R. Allu, B. Andhavarapu, M. Bagadi and M. Belusonti, "Text based sentiment analysis using LSTM," International Journal of Engineering Research & Technology, vol. 9 Issue 05, 2020.

[42] N. Makhoul, "Review of data quality indicators and metrics, and suggestions for indicators and metrics for structural health monitoring," Advances in Bridge Engineering, vol. 3, p. 17, 2022.

[43] M. Steurer, R. J. Hill and N. Pfeifer, "Metrics for evaluating the performance of machine learning based automated valuation models," Journal of Property Research, vol. 38, pp. 99-129, 2021.

[44] D. Chicco, M. Warrens and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," vol. 7, p. PeerJ Computer Science, 05 07 2021.

[45] P. Sandhya, R. Bandi and D. D. Himabindu, "Stock price prediction using recurrent neural network and lstm," 2022.

[46] P. Mukherjee, Y. Badr, S. Doppalapudi, S. M. Srinivasan, R. S. Sangwan and R. Sharma, "Effect of negation in sentences on sentiment analysis and polarity detection," Procedia Computer Science, vol. 185, pp. 370--379, 2021.