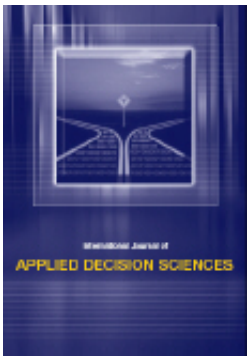


International Journal of Applied Decision Sciences

ISSN online: 1755-8085 - ISSN print: 1755-8077
https://www.inderscience.com/ijads

Sentiment analysis on stocks: a hybrid feature extraction technique on 14 classifiers

Meera George, R. Murugesan

DOI: 10.1504/IJADS.2025.10059408

Article History:

Received:	08 June 2023
Last revised:	06 August 2023
Accepted:	08 August 2023
Published online:	03 December 2024

Sentiment analysis on stocks: a hybrid feature extraction technique on 14 classifiers

Meera George* and R. Murugesan

Department of Humanities and Social Sciences,
National Institute of Technology,
Tiruchirappalli – 620015, Tamil Nadu, India
Email: 409121002@nitt.edu
Email: pa.ap3years@gmail.com
*Corresponding author

Abstract: Accurately predicting stock prices is challenging and has garnered massive attention from researchers and investors alike. Though the literature has shown sentiment analysis as a promising approach for efficient stock price prediction, it has found a considerable gap in studies using multiple feature extraction techniques with hybrid models for the efficient sentiment classification. Under these circumstances, this study aims to perform sentiment analysis using five feature extraction techniques including a hybrid and 14 classifiers for the accurate classification of stock tweets. The study extracted 21,121 tweets spanning March 2022 to December 2022 using Twitter application programming interface. The empirical result shows the superiority of the hybrid feature extraction technique over the other methods. The support vector machine classifier with a hybrid feature extraction technique is found to be the best-performing sentiment analysis model for Twitter stock data. The study has potential applications in building optimal investment strategies and decision-making.

Keywords: stock price; sentiment analysis; classifiers; feature extraction; hybrid.

Reference to this paper should be made as follows: George, M. and Murugesan, R. (2025) 'Sentiment analysis on stocks: a hybrid feature extraction technique on 14 classifiers', *Int. J. Applied Decision Sciences*, Vol. 18, No. 1, pp.84–112.

Biographical notes: Meera George is a PhD candidate in Economics in the Department of Humanities and Social Sciences, National Institute of Technology Tiruchirappalli. Her research focuses on sentiment analysis and stock price prediction using machine learning and deep learning techniques.

R. Murugesan is a Professor in the Department of Humanities and Social Sciences, National Institute of Technology Tiruchirappalli. With 25 years of teaching and research experience, he has several publications in journals of international repute and many internationally sponsored research projects to his credit. His areas of interest include econometrics, ML techniques, financial economics, microeconomics, and macroeconomics.

1 Introduction

Stock price, being dynamic, nonlinear, noisy, and highly chaotic, though theoretically determined by the law of supply and demand, is affected by many interrelated factors like economics, industry, psychology, and politics (Gao et al., 2022). Predicting stock prices is essential in the financial area as it provides investors with helpful information for profit generation (Li et al., 2020). Stock price prediction has become a centre of debate in academia and research communities (Jing et al., 2021). The complex nature of the stock market has made stock price prediction a highly challenging task. Researchers are constantly striving to improve the accuracy of stock price prediction by integrating multiple resources.

Over the years, researchers have used two critical approaches for stock prediction: technical analysis (SMA, EMA, RSI, MOM, WMA, MACD, MFI, stochastic indicator, TR, CCI, PPO, TRIX, ULTOSC) and fundamental analysis using macroeconomic variables such as exchange rate, inflation, CPI, LTIR, IIP (Al Silni et al., 2021; Karasu et al., 2020; Gruszka and Szwabiński, 2021; Ma et al., 2023; Hoseinzade and Haratizadeh, 2019; Zhang and Tang, 2023; Tzeng, 2022; Goel and Singh, 2021). In recent years, machine learning models [logistic regression (LR), support vector machine (SVM), decision tree (DT), random forest (RF), quadratic discriminant analysis (QDA)] and deep learning models [convolution neural networks (CNN), recurrent neural network (RNN), multilayer perceptron (MLP), gated recurrent units (GRU), long short-term memory (LSTM), Bi-LSTM] have gained massive attention from academia, and their growing prominence led to the extensive application of sentiment analysis (SA) for stock price prediction.

The SA, called opinion mining, is a natural language processing task used to extract sentiments and opinions from textual and visual data. It finds the polarity of the data and classifies them into negative, positive, and neutral sentiments. The investor sentiments extracted using SA significantly contribute to predicting stock prices, influencing the investor's investment decision (Qiu et al., 2022). Recent studies have shown that an increasing number of researchers use SA to extract investors' sentiments from texts, images, and videos and apply machine learning algorithms to predict prices (Picasso et al., 2019).

Studies have used different feature extraction techniques for the SA of the stock data. Wu et al. (2021) classified the stock sentiments using CNN with word vectors extracted from Word2Vec as the input layer. Almalis et al. (2022) performed the SA on the financial news using TF-IDF and different classifiers, including DistilBert, LSTM, GRU, SVM, FinBERT, ERT, RF, and NB. Yang et al. (2020) employed four different vectorisation techniques-Word2Vec, Doc2Vec, TF-IDF, and CountVectorizer, to convert stock news into vectors. Mishev et al. (2019) combined Word2Vec, Glove, and FastText word embedding with Bi-GRU and CNN for the sentiment extraction of financial news related to stocks. Based on the available literature, it can be inferred that the majority of the studies have used single feature extraction technique (Carosia et al., 2019; Souma et al., 2019) or are limited to a maximum of three feature extraction techniques (Yang et al., 2020; Ganesh et al., 2021) in the SA of the stock data. The literature also reveals that no studies have used a hybrid feature extraction technique using Doc2Vec and TF-IDF for the SA of tweets related to stocks. This study attempts to fill these gaps using five feature extraction techniques – Word2Vec, Doc2Vec, TF-IDF, FastText, and hybrid (Doc2Vec+TF-IDF). The literature also shows the application of different classifiers like

NB, MLP, SVM, MaxEnt, RF, SVM, NB, and LR (Renault, 2019; Hu and Tripathi, 2017) for classifying investor sentiments. However, no studies have used a combination of 14 machine-learning classifiers for the SA of stock tweets. Reacting to this, the study uses 14 classifiers – RF, K-nearest neighbours (KNN), LR, GB, DT, SVM, ET, AB, dummy classifier (DC), ridge classifier (RC), MNB, binomial naive Bayes (BNB), linear discriminant (LD), and QD for the classification of the Twitter data related to the stock market which is first of its kind. Thus, the primary objective of the study is to perform a SA on Twitter data related to stock using five feature extraction techniques and 14 machine learning classifiers.

The study contributes to the literature as follows:

1 To the best of the authors' knowledge, the study is the first to employ a hybrid feature extraction technique combining Doc2Vec and TF-IDF on stock tweets.

2 The study uses 14 machine learning classifiers – RF, KNN, LR, GB, DT, SVM, ET, AB, DC, RC, MNB, BNB, LD, and QD for the classification of the Twitter stock data, which is the first of its kind.

3 Limited studies have used multiple feature extraction techniques for the SA of stock data. No study has done a comparative analysis of the following feature extraction techniques: Word2Vec, Doc2Vec, TF-IDF, and FastText. This study uses five feature extraction techniques, including a hybrid.

The results of the study show that the SVM with a hybrid feature extraction technique is the best-performing model for Twitter stock data SA. Moreover, the hybrid technique using TF-IDF and Doc2Vec is superior to the other four feature extraction techniques for 12 classifiers. The study is capable of understanding the general sentiments of the investors, thereby aiding the decision-making of investors and other stakeholders.

This study is organised as follows: Section 1 deals with the introduction. Section 2 presents the literature review of the study. The methodology is presented in section 3. Section 4 gives the results and discussion, and Section 5 presents the study's main conclusion.

2 Review of literature

The literature presents extensive studies on SA of stock data using different ML and DL techniques. Carosia et al. (2019) analysed the investor sentiments from news and tweets related to the Brazilian stock market using FastText and different classifiers like NB, LR, SVM, MLP, LSTM, and CNN, where CNN was found to be better performing with 86.5% accuracy. Jing et al. (2021) proposed a SA model for the Chinese stock forum using CNN classifier and Word2Vec. The model's performance was evaluated and compared with other baseline classifiers, LR, SVM, RNN, and LSTM, where the proposed model was found to perform better with a precision of 87.5%. Liu et al. (2023) analysed the synergy between stock prices and investor sentiments. The authors used SA to extract the investor sentiments of 30,200 investor messages in the Dongfang Fortune stock bar. The study employed Word2vec embedding technique and multiple CNNs for the text classification and received an accuracy of 89.14%.

Li et al. (2021) proposed a novel SA model using BERT to classify the Chinese stock market reviews. The study obtained the feature vectors from the pre-trained language

model, BERT, and was used as input for the three classification layers: BERT+FC (fully connected layer), BERT+LSTM, and BERT+CNN. The BERT+FC model showed higher performance with an F1 score of 92.50%, and the effectiveness of this model was further tested using four other methods: Word2Vec+ TextCNN, Word2Vec+ TextRNN, Word2Vec+ Att-BLSTM., and Word2Vec+ TextCRNN. The proposed model was shown a higher performance with 92.65% accuracy. Souma et al. (2019) used the pre-trained word vector of Glove and RNN with LSTM to define the polarity of the news sentiments related to stock. The model obtained an accuracy of 76% for predicting the positive news as positive and 75% for predicting the negative news as negative. Yang et al. (2020) applied different NLP tools: TF-IDF, Word2Vec, CountVectorizer, Doc2Vec, and machine learning classifiers: Adaboost (AB), XGboost, LR, DT, and gradient-boosted decision trees (GBDT) to predict the future rise and fall of stocks. The proposed model with Doc2Vec and GBDT outperformed other models and obtained an accuracy of 64.56%.

Ganesh et al. (2021) predicted the sentiment of the news snippets extracted from the LexisNexis database using four different vectorisation methods: Word2Vec, TF-IDF, Doc2Vec, and TF-IDF+ Doc2Vec and five different classifiers: NB, LR, SGD, SVM, and RF. The SVM with TF-IDF+ Doc2Vec model outperformed the other models with an accuracy of 80.8%. Shi et al. (2020) proposed a new SA system with Word2Vec as a text embedding technique and CNN, GRU and LR for the sentiment classification. The study found the deep learning models, CNN and GRU to have a 9% improvement over the machine learning model LR. Dey and Das (2023) proposed a hybrid SA model with TF-IDF for vectorisation and CNN and LSTM for the sentiment classification. The proposed model outperformed other baseline models with an accuracy of 80.3% for financial news dataset. Adhikari et al. (2023) performed SA on financial news dataset using Word2Vec, BERT and POS vectors for word representation and CNN for the sentiment classification. The proposed model was found to have an accuracy of 86% compared to other baseline classification models. Zong et al. (2022) analysed the sentiments of the financial news related to stock market using Word2Vec and SVM classifier. The accuracy of the news sentiment classifier was found to be 86%.

To establish a correlation between the investor sentiment index and stock price volatility, Xue et al. (2021) performed a SA using the posts from the stock forum Guba. The study employed the TF-IDF vectorisation method and maximum entropy classifier to extract the investor sentiments and obtained an accuracy of 72.3%. Mishev et al. (2019) executed the SA on financial news headlines related to the stocks using different word embeddings, sentence embeddings, machine learning, and deep learning classifiers. The study used Word2Vec, Glove, and FastText embeddings with two deep learning models, BiGRU-Attention and CNN, wherein the model with Glove and BiGRU-Attention exhibited a higher performance with an F1 score of 89.3. The summary of the literature is presented in Table 1.

Apart from the text embedding techniques, the studies have also used several lexicon based approaches like sentiment dictionaries and online twitter sentiments for the SA. Gu and Kurov (2018) used Bloomsberg's firm specific Twitter sentiments along with RNN and LSTM to analyse stock market returns. Thormann et al. (2021) analysed the sentiments of the Twitter data using TextBlob and predicted the stock prices of Apple 30 m and 60 m ahead using LSTM. Kolasani and Assaf (2020) used Sentiment40 Twitter data and trained the machine learning models to predict the closing prices of Apple Inc. AAPL and DJIA stocks. Groß-Klußmann et al. (2019) examined the relationship between

the large social media data and stock indices using sentiments extracted from 102 million tweets using a dictionary approach.

Table 1 Summary of the studies discussed in the review of the literature

Study	*Feature extraction techniques*	*Methods*
Carosia et al. (2019)	FastText	NB, LR, SVM, MLP, LSTM, CNN
Jing et al. (2021)	Word2Vec	CNN, LR, SVM, RNN, LSTM
Li et al. (2021)	BERT	BERT+FC, BERT+LSTM, BERT+CNN
Liu et al. (2023)	Word2Vec	CNN
Yang et al. (2020)	TF-IDF, Word2Vec, CountVectorizer, Doc2Vec	Adaboost, XGboost, LR, DT, GBDT
Ganesh et al. (2021)	TF-IDF, Word2Vec, Doc2Vec, hybrid	LR, SGD, SVM, NB, RF
Xue et al. (2021)	TF-IDF	ME
Mishev et al. (2019)	Word2Vec, Glove, FastText	Bi-GRU, CNN
Souma et al. (2019)	Glove	RNN with LSTM
Shi et al. (2020)	Word2Vec	CNN, GRU, LR
Dey and Das (2023)	TF-IDF	CNN+LSTM
Adhikari et al. (2023)	Word2Vec, BERT, POS vectors	CNN
Zong et al. (2)022	Word2Vec	SVM
Proposed model	*TF-IDF, Word2Vec, Doc2Vec, FastText, Doc2Vec+TF-IDF*	*RF, KNN, LR, GB, DT, SVC, ET, AB, DC, RC, GNB, BNB, LD and QD*

The extensive literature survey helps to draw the following inferences. SA in the stock market is largely carried out using machine learning and deep learning approaches. The literature sights the application of different feature extraction techniques like TF-IDF, Word2Ve, Doc2Vec, FastText, Glove, and BERT for extracting features from text data. Numerous studies have employed multiple machine learning and deep learning classifiers like NB, LR, CNN, ME, SVM, and RF to classify sentiments. However, the inferences lead to a major research gap, where limited studies have used multiple feature extraction techniques, and no studies have used 14 classifiers for the SA of stock tweets.

Hence, this study performs SA on 21,000 tweets related to the stock market. The study uses five feature extraction techniques: Word2Vec, Doc2Vec, TF-IDF, FastText, and hybrid (Doc2Vec+TF-IDF) and 14 machine learning classifiers: RF, KNN, LR, GB, DT, SVM, ET, AB, DC, RC, MNB, BNB, LD, and QD. The tweets are labelled using the lexicon sentiment dictionary, text blob, and are further used in training the classifiers. The study found that the hybrid feature extraction technique, along with the SVM, outperforms all other classifiers in the SA of stock tweets.

3 Methodology

3.1 Data

To perform the SA on the stock market, tweets related to the stock market were collected using the keywords ‘stock market’ and ‘stocks’. 21,121 tweets for 10 months spanning March 2022 to December 2022, were collected using Twitter API. The re-tweets were filtered out and were then pre-processed for analysis.

3.2 Pre-processing

This study employs nine pre-processing techniques: lowercase conversion, white space removal, tokenisation, stop words removal, punctuation removal, frequent words removal, stemming, lemmatisation, URL removal, and HTML removal. Table 2 gives a picture of the nine pre-processing techniques used by the study using an example.

- URL removal: It is an essential step in data pre-processing where the URL from the texts is removed.
- Lowercase conversion: The text data is wholly converted into lowercase such that words with the same meaning but different cases (upper case, lower case, a mixture of upper and lower case) are not treated differently.
- White space removal: Extra spaces that do not provide any values to the data but consume the text size are removed in this step.
- Tokenisation: The text is split into tokens of sentences, words, or characters in this step.
- Stop words removal: In this step, certain trivial words are removed to reduce the noise in the clean data.
- Punctuation removal: The data set is standardised by removing the punctuations or characters that provide no values to the text using the re function.
- Stemming: The affixes from the words are removed to extract the base form of the words in the text.
- Lemmatisation: Similar to stemming, the affixes of the words are removed to extract the root word from the text.
- HTML removal: Twitter data consisting of multiple tags which act as noise during the classification tasks are removed from the text.

3.3 Feature extraction

The study employs different feature extraction techniques: TF-IDF, Word2Vec, Doc2Vec, FaxtText, and a hybrid technique combining TF-IDF and Doc2Vec for SA.

Table 2 Example for pre-processing techniques done on the study

#Nomura upgrades #Concor stock to BUY from NEUTRAL, <increases the target price to ₹918 from ₹775> https://example.com	Original text
#Nomura upgrades #Concor stock to BUY from NEUTRAL, <increases the target price to ₹918 from ₹775>	URL removal
#nomura upgrades #concor stock to buy from neutral, <increases the target price to ₹918 from ₹775>	Lower case
#nomura upgrades #concor stock to buy from neutral, <increases the target price to ₹918 from ₹775>	White space removal
[‘#’, ‘nomura’, ‘upgrades’, ‘#’, ‘concor’, ‘stock’, ‘to’, ‘buy’, ‘from’, ‘neutral’, ‘,’, ‘<‘, ‘increases’, ‘the’, ‘target’, ‘price’, ‘to’, ‘₹’, ‘918’, ‘from’, ‘₹’, ‘775’, ‘>‘]	Tokenisation
[‘#’, ‘nomura’, ‘upgrades’, ‘#’, ‘concor’, ‘stock’, ‘buy’, ‘neutral’, ‘,’, ‘<‘, ‘increases’, ‘target’, ‘price’, ‘₹’, ‘918’, ‘₹’, ‘775’, ‘>‘]	Stop words removal
[‘nomura’, ‘upgrades’, ‘concor’, ‘stock’, ‘buy’, ‘neutral’, ‘increases’, ‘target’, ‘price’, ‘918’, ‘775’]	Punctuation removal
[‘nomura’, ‘upgrade’, ‘concor’, ‘stock’, ‘buy’, ‘neutral’, ‘increase’, ‘target’, ‘price’, ‘918’, ‘775’]	Lemmatisation
[‘nomura’, ‘upgrad’, ‘concor’, ‘stock’, ‘buy’, ‘neutral’, ‘increas’, ‘target’, ‘price’, ‘918’, ‘775’]	Stemming
nomura upgrad concor stock buy neutral increase target price 918,775	HTML removal

3.3.1 Word2Vec

Word2Vec is a two-layer neural network trained to refine the linguistic contexts of words (Yilmaz and Toklu, 2020). This tool identifies the semantic relationships between the words in a sentence and transforms these words into vectors (Ballı and Karasoy, 2019). The model has two learning algorithms primarily; CBOW and continuous skip-gram. CBOW predicts the target word from its context words, while continuous skip-gram predicts the context words from an input word (Khatua et al., 2019). This study uses the continuous skip-gram algorithm.

3.3.2 Doc2Vec

Doc2vec is a deep learning algorithm for text vectorisation (ZhengWei et al., 2022), encompassing word and document vectors. It is an extension of word2vec applied on a whole document instead of individual words, thereby vectorising a document rather than a word (Chen and Sokolova, 2021).

3.3.3 TF-IDF

TF-IDF is a statistical technique indicating the significance of words in a collection of documents (Agarwal et al., 2020). The computation uses normalised term frequency (TF) and document frequency (IDF). Here, TF calculates the frequency of the word/term in a given document, and DF calculates the frequency of a word/term in a collection of documents.

3.3.4 FastText

FastText, based on neural networks, is a popular word embedding technique for learning high-quality word representations. It is an extension of Word2Vec and uses a skip-gram model for obtaining the embeddings (Ghosal and Jain, 2023).

3.3.5 Hybrid technique

The study proposes a hybrid feature extraction technique for Twitter stock data SA. This method involves the concatenation of TF-IDF vectors and Word2Vec vectors. The following equations (1)–(4) show the significant steps involved in this technique.

Step 1 Calculating TF-IDF scores

$$TF_{t,d} = \frac{F_{td}}{\sum_n F_{nd}} \tag{1}$$

$$DF_t = \frac{d \in D : m \in d}{|D|} \tag{2}$$

$$IDF_t = \log \frac{|D|}{d \in D : m \in d} \tag{3}$$

$$TFIDF_{t,d} = TF_{t,d} * \log \frac{|D|}{d \in D : m \in d} \tag{4}$$

where F_{td} is the frequency of t^{th} term in the d^{th} document, and $\sum_n F_{nd}$ is the mean frequency of all the n words in the document, d or can also be referred to as the length of the document. $|D|$ is the collection of documents, and $d \in D$: $m \in d$ is the occurrence of the term m in the collection of documents $|D|$.

Step 2 Calculate the Doc2Vec vectors

Step 3 Concatenate the TF-IDF vectors and Doc2Vec vectors

$$T(d_i) = \varnothing\left(TFIDF(d_i), d2v(d_i)\right) \tag{5}$$

where $TFIDF(d_i)$ represents the vectors of the i^{th} document extracted using the TF-IDF approach, and $d2v(d_i)$ shows the vectors of the i^{th} document extracted using the Doc2Vec approach. $\varnothing$ represents the concatenation process. $T(d_i)$ is the total vectors of the i^{th} document after the concatenation.

3.4 Sentiment classifiers

The SA is carried out using 14 classifiers that include the extra trees (ET) classifier, random forest classifier (RFC), RC, linear discriminant analysis (LDA), QDA, K-neighbours classifier, gradient boosting (GB) classifier, DT classifier, Ada boost classifier, multinomial naïve Bayes (MNB) classifier, Bernoulli naïve Bayes (BNB) classifier, LR, DC, and SVM-linear kernel.

3.4.1 Random forest classifier

RFC is an ensemble model classifier built using several DTs. Given s = 1, 2, 3, ……, S bootstrap samples $\{X_s, y_s\}$, the final class from the random forest, $\widehat{C_{RF}}(x)$, can be deduced with the help of equation (6).

$$\hat{C}_{RF}(x) = majorityvote \tag{6}$$

3.4.2 KNN

KNN classifier performs the classification by computing the similarity between the new dataset and the trained dataset to detect the k-nearest neighbours and allocating the new dataset to the class with the most k-neighbours with it (Kumbure et al., 2020). Euclidean distance is the most common distance metric used and is represented in equation (7).

$$d_i = \sum_{i=1}^{S} (x_i - x_t)^2 \tag{7}$$

where x_i represent training samples, and x_t represent the testing sample. S is the total number of samples.

3.4.3 Ridge classifier

RC is a modified ridge regression model employed to perform classification tasks and can be calculated using equation (8).

$$\beta_r = (A'A + \alpha U)^{-1} A'y \tag{8}$$

where β_r is the parameter coefficient vector, A represents the feature matrix, U is the unit matrix, and y is the predicted vector. α is the tuning parameter, also called the shrinking parameter, whose value lies between $0 < \alpha < 1$.

3.4.4 Linear discriminant analysis

LDA classifier is an extension of the Fishers discriminant function (Dodia et al., 2019) capable of data classification and dimensionality reduction (Al-Dulaimi et al., 2019). The LDA score, D_L, can be computed using equation (9).

$$D_L = \left(A_i - \overline{A_c}\right)' \mathbf{\Sigma_p}^{-1} \left(A_i - \overline{A_c}\right) - 2\log_e \rho_c \tag{9}$$

where A_i is the unknown measurement vector for the i^{th} sample, $\overline{A_c}$ is the mean measurement vector for class c, $\mathbf{\Sigma p}$ is the pooled covariance matrix, and ρ_c is the prior probability of class c. The ρ_c, Σ_p, Σ_c (covariance matrix of class c) can be calculated using equations (10)–(12).

$$\rho_c = \frac{T_c}{T} \tag{10}$$

$$\Sigma_p = \frac{1}{T} \sum_{c=1}^{C} T_c \Sigma_c \tag{11}$$

$$\Sigma_p = \frac{1}{T_c}\sum_{i=1}^{T_c}\left(A_i - \overline{A_c}\right)\left(A_i - \overline{A_c}\right)' \tag{12}$$

where T_c is the number of objects of class c, T is the total number of objects in the training dataset, and C is the total number of classes.

3.4.5 Quadratic discriminant analysis

Unlike LDA, QDA provides nonlinear data analysis (Toğaçar et al., 2020). The QDA score, D_Q, can be calculated using equation (13).

$$D_Q = \left(A_i - \overline{A_c}\right)' \boldsymbol{\Sigma}_{\mathbf{c}}^{-1}\left(A_i - \overline{A_c}\right) + \log_e |\boldsymbol{\Sigma}_{\mathbf{c}}| - 2\log_e \rho_c \tag{13}$$

where $\boldsymbol{\Sigma}_c$ is the covariance matrix of class c.

3.4.6 Gradient boosting

GB classifier is a machine learning model used for regression and classification tasks that combines a group of weak classifiers to create a robust predictive model. Given the N number of classifiers, the n^{th} weak classifier is calculated using equation (14).

$$f_N(x) = \sum_{n=1}^{M} r_m t_m(x) \tag{14}$$

where r_m is the residual and $t_m(x)$ is the DT obtained during the residual.

3.4.7 Decision tree

The DT classifier is a non-parametric supervised machine learning classifier. It uses tree-like structures wherein the conditions and class labels are represented in the internal and external nodes, respectively (Zulfiker et al., 2020). The attributes are selected using information gain as represented in equations (15) and (16).

$$Info\ Gain(D, F) = E(D) - \sum_{i \in V} \frac{D_i}{D} E\left(D_i\right) \tag{15}$$

$$E(D) = -\sum_{i=1}^{S} p_i \left(\log_2 p_i\right) \tag{16}$$

where D is the training dataset, and $E(D)$ is its entropy. $V(F)$ is the set of all possible values for the feature, F. D_i is the training subset of D where F has value i. S is the sum of all classes, and p_i is the proportion of class i in S.

3.4.8 Extra trees

ET classifier, an extremely randomised tree, forms several independent DTs to perform classification and regression tasks (Kiala et al., 2021). This classifier utilises the whole training data set for the DT construction and employs different randomisation techniques for selecting the node splits. The information gain selects the attributes as represented in equation (17).

$$G_i = 1 - \sum_{l=1}^{S} (p_l)^2 \tag{17}$$

where S is the number of unique class labels, and p_l is the probability of an element being classified as label, l.

3.4.9 AdaBoost

AB is an iterative training algorithm where multiple weak classifiers are trained and assembled to create a strong classifier (Hu et al., 2020). The linear combination of a series of weak classifiers creates a strong classifier, as represented in equation (18).

$$C(x) = \sum_{s=1}^{S} \sigma_s c_s(x) \tag{18}$$

where $c_s(x)$ is a weak classifier, σ_s is the weight of $c_s(x)$ in the strong classifier, and $C(x)$ is the linear combination of weak classifiers.

3.4.10 Multinomial naïve Bayes

MNB is a probabilistic classifier based on multinomial distribution. It effectively calculates the frequency of an item. MNB algorithm can be represented using equation (19).

$$P(s/t) \propto p(s)\Pi_{1 \le i \le n_t} p(w_i / s) \tag{19}$$

$P(s / t)$ is the posterior probability of getting class s from the given document t. $p(s)$ is the prior probability of class s. $p(w_i / s)$ is the probability of getting i^{th} word w_i in the class s. n_t represents the total number of words in the document t.

3.4.11 Bernoulli naïve Bayes

BNB classifier is another probabilistic classifier that accounts for the presence or absence of a word in the document without considering its frequency. The equation for BNB can be represented as in equation (20).

$$P(x_i / s) = a_i P(x_i / s) + 1 - a_i \left(1 - p(x_i / s)\right) \tag{20}$$

where $P(x_i / s)$ is the conditional probability of feature x_i in class s. a_i is a binary indicator that represents the presence or absence of the feature (if $a_i = 1$, the feature is present, and $a_i = 0$, the feature is absent).

3.4.12 Logistic regression

LR is a machine-learning model for solving supervised classification problems (Beitia-Antero et al., 2018). It models the conditional probability as given in equation (21).

$$p(y_i \mid x_i) = \frac{1}{1 + e^{-y_i (w^T x_i + c)}} \tag{21}$$

where x_i is a feature vector in the training dataset with $y_i \in (+1, -1)$ labels, w is the weight vector, and c is the global bias.

3.4.13 Dummy classifier

DC makes predictions based on simple rules and acts as a baseline classifier to compare the accuracy of the other classifiers. If n number of classes are present and are distributed equally within the dataset, the probability of assigning label b_s to an input is given in equation (22).

$$P(b_s) = \frac{1}{n} \text{ where } s = 1, 2, ..., n \tag{22}$$

3.4.14 SVM-linear kernel (SVM)

SVM is a generalised linear classifier used for binary data classification in supervised machine learning (Liu et al., 2020). The SVM, which uses a linear kernel, is called a linear SVM and is mathematically given in equation (23).

$$k(x_i, x) = x_i x \tag{23}$$

where x_i and x are two data points, and k is the kernel.

3.5 Performance evaluation

The performance of the classifiers is evaluated using the following evaluation metrics.

- Accuracy: It is the sum of correctly classified samples to the total and can be represented using equation (24).

$$Accuracy = \frac{T_p + T_n}{T_p + F_p + T_n + F_n} \tag{24}$$

  where T_p, T_n, F_p, F_n are true positives, true negatives, and false positives and false negatives, respectively.

- Recall: This is the total of accurately classified positive samples to the total positive samples. It is calculated using equation (25).

$$Recall = \frac{T_p}{T_p + F_n} \tag{25}$$

- Precision: This calculates the sum of accurately classified positive samples to the total classified positive samples and can be expressed as in equation (26).

$$Precision = \frac{T_p}{T_p + F_p} \tag{26}$$

- F1 score: It is the harmonic mean of recall and precision, which is given by equation (27).

$$F_1\ score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{27}$$

- AUC: It measures the ability of the classifier to differentiate between positive and negative classes, and their value lies between 0 and 1.

4 Results and discussion

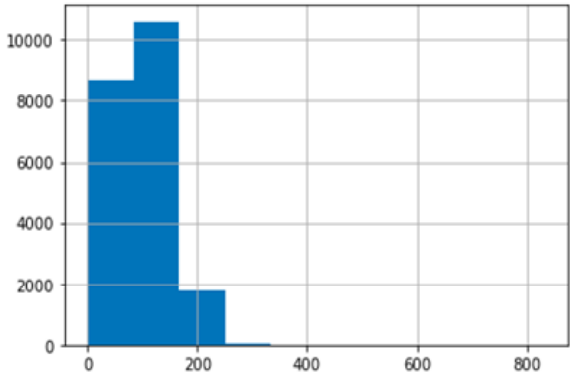
The study used popular Python libraries – Panda, Numpy, Scikit-learn, NLTK, TextBlob, and Genism libraries to carry out the analysis. The methodology of this study follows four major steps: data extraction, pre-processing, feature extraction, and sentiment classification. 21,121 tweets were extracted for the study using Twitter API. Figure 1 shows the actual text data with the first five and last five rows. The number of characters in the tweets ranges from 50–300, and the number of words ranges from 5–50, as shown in Figures 2 and 3, respectively. This indicates the different lengths of the extracted tweets, with some longer than others.

Figure 1 Actual text data

0	#Wipro $WIPRO Last 36 months Daily #StockMovem...
1	RT @smartmoneyEmpi1: still fresh. join us an...
2	$SUB in -0.23% Downtrend, declining for three ...
3	TATA COMMUNICATION\n\nEXCELLENT CHANNEL PATTER...
4	RT @ProdigalTrader: Ã¢Â Â Accumulation is fini...

21117	Yes, beating the bear market is possible. Thes...
21118	#StockMarket outlook: Rally could spur more ha...
21119	EMERGING MARKETS-Brazil's #stocks, real lead L...
21120	US #stocks Higher; Dow Rises Over 100 Points -...
21121	Carol, a former stock trader working in global...

Figure 2 Number of characters (see online version for colours)



To capture the contextual information, the text is further categorised into bigrams and trigrams. Bigrams and trigrams are different n-grams with continuous sequences of n items in a document. Figures 4 and 5 represent the specific bigrams and trigrams in the study, along with their frequencies in the text data. In this study, bigrams such as 'stock market,' 'chines stock,' 'trade stock,' and trigrams: 'view odd indic,' '2022 view odd', 'trade stock market,' 'day mov average' dominates the tweets. The word cloud, as shown

in Figure 6, visually represents the most frequent words in the corpus of tweets. It provides a quick overview of the major topics in the dataset. Here, stock, market, trade, stockmarket are the words with the highest frequency in the extracted data.

Figure 3 Number of words (see online version for colours)

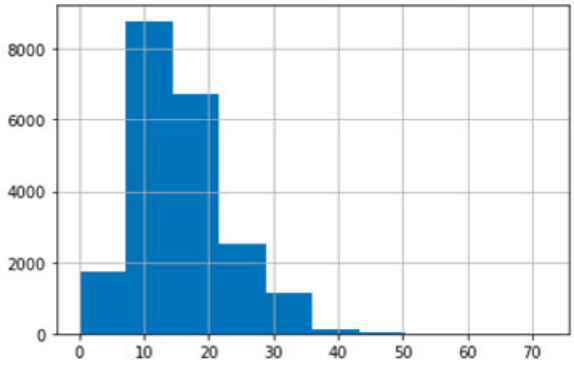
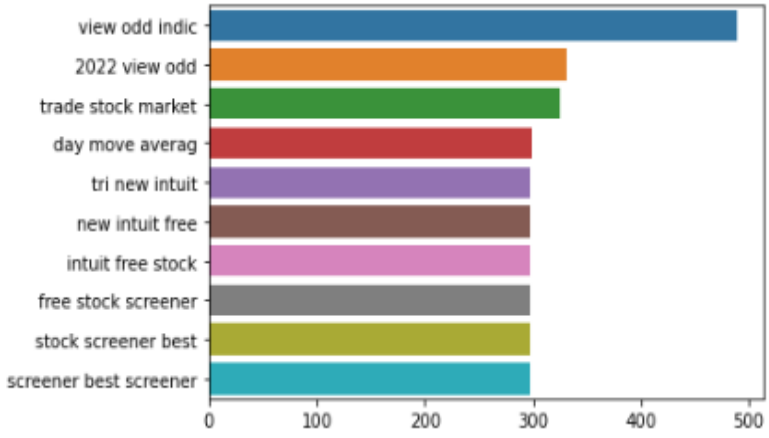


Figure 4 Bi-grams (see online version for colours)



The extracted data contains many inconsistencies and ambiguities, which are removed in the process of pre-processing using the NLTK library in Python. Pre-processing is done using nine techniques: lowercase conversion, white space removal, tokenisation, stop words removal, punctuation removal, frequent words removal, stemming, lemmatisation, URL removal, and HTML removal. These steps help in cleaning and standardising the data for further analysis. The results of the pre-processing steps executed on the extracted data are shown in Figures 7(a)–7(h).

The study employs 14 classifiers with five featurisation techniques: TF-IDF, Word2Vec, Doc2Vec, FastText, and a hybrid technique combining TF-IDF and Doc2vec for the SA of the stock tweets. The performance of the classifiers is evaluated using accuracy, precision, recall, F1 score, AUC-receiver operating characteristic (ROC) curve, and AUC-precision-recall (PR) curve. The performance results based on accuracy, precision, recall, and F1 score are presented in Table 3. Given five feature extraction techniques, the best-performing technique for each classifier is highlighted in the Table 3.

The study finds that among five featurisation techniques, the hybrid technique has higher accuracy for 12 classifiers: SVM-linear kernel (95%), DT (95%), random forest (94%), LR (93%), RC (93%), LDA (92%), QDA (87%), GB (87%), BNB (86%), DTs (84%), AB (84%) and ET (80%). For the classifiers KNN and Gaussian Naïve Bayes, Doc2Vec performs better with an accuracy of 83% and 66%, respectively. The top five best-performing classifiers are found to be SVM-linear kernel (95%), DT (95%), random forest (94%), LR (93%), and RC (93%), and the average of the classification accuracies of these classifiers can be summarised as SVM > DT > RFC > LR > RC. This finding aligns with the studies (Luo, 2021; Alsmadi and Hoon, 2018; Elnagar et al., 2019) where SVM outperformed the other machine learning classifiers in text classification.

Figure 5 Tri-grams (see online version for colours)

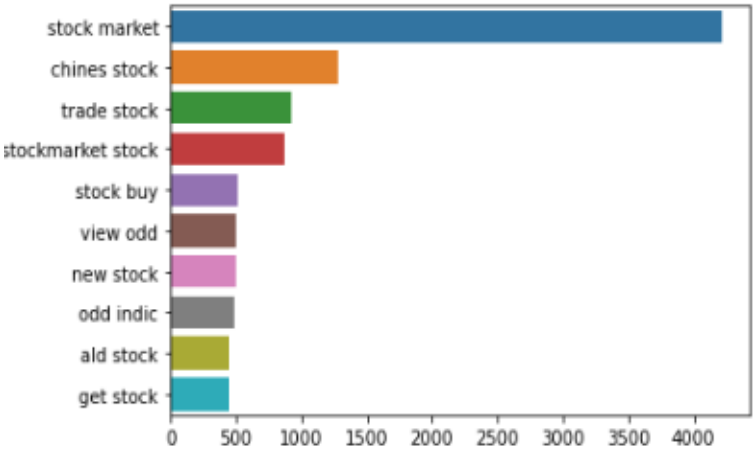
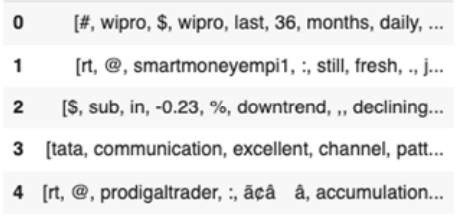


Figure 6 Word cloud (see online version for colours)

Figure 7 (a) Results after URL removal (b) Results after lowercase conversion (c) Results after white space removal and tokenisation (d) Results after punctuation removal (e) Results after stop words removal (f) Results after lemmatisation (g) Results after stemming (h) Results after tag removal

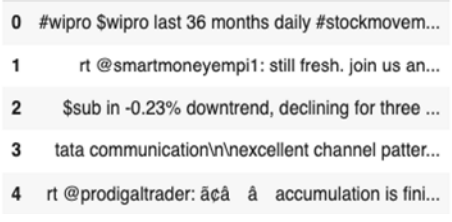


0	[#, wipro, $, wipro, last, 36, months, daily, ...
1	[rt, @, smartmoneyempi1, :, still, fresh, ., j...
2	[$, sub, in, -0.23, %, downtrend, ,, declining...
3	[tata, communication, excellent, channel, patt...
4	[rt, @, prodigaltrader, :, ã¢â â, accumulation...

(a)

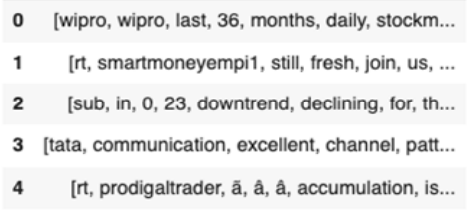
0	#Wipro $WIPRO Last 36 months Daily #StockMovem...
1	RT @smartmoneyEmpi1: still fresh. join us an...
2	$SUB in -0.23% Downtrend, declining for three ...
3	TATA COMMUNICATION\n\nEXCELLENT CHANNEL PATTER...
4	RT @ProdigalTrader: Ã¢Â Â Accumulation is fini...

(b)



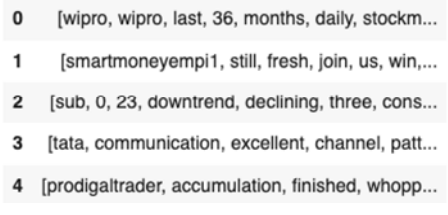
0	#wipro $wipro last 36 months daily #stockmovem...
1	rt @smartmoneyempi1: still fresh. join us an...
2	$sub in -0.23% downtrend, declining for three ...
3	tata communication\n\nexcellent channel patter...
4	rt @prodigaltrader: ã¢â â accumulation is fini...

(c)



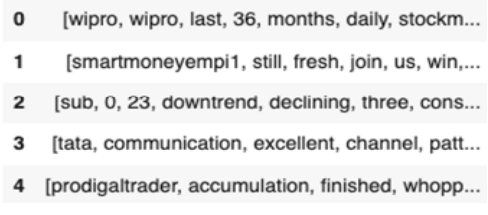
0	[wipro, wipro, last, 36, months, daily, stockm...
1	[rt, smartmoneyempi1, still, fresh, join, us, ...
2	[sub, in, 0, 23, downtrend, declining, for, th...
3	[tata, communication, excellent, channel, patt...
4	[rt, prodigaltrader, ã, â, â, accumulation, is...

(d)



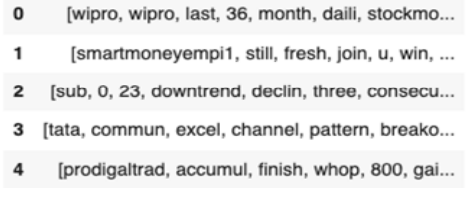
0	[wipro, wipro, last, 36, months, daily, stockm...
1	[smartmoneyempi1, still, fresh, join, us, win,...
2	[sub, 0, 23, downtrend, declining, three, cons...
3	[tata, communication, excellent, channel, patt...
4	[prodigaltrader, accumulation, finished, whopp...

(e)



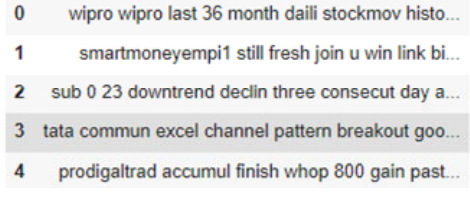
0	[wipro, wipro, last, 36, months, daily, stockm...
1	[smartmoneyempi1, still, fresh, join, us, win,...
2	[sub, 0, 23, downtrend, declining, three, cons...
3	[tata, communication, excellent, channel, patt...
4	[prodigaltrader, accumulation, finished, whopp...

(f)



0	[wipro, wipro, last, 36, month, daili, stockmo...
1	[smartmoneyempi1, still, fresh, join, u, win, ...
2	[sub, 0, 23, downtrend, declin, three, consecu...
3	[tata, commun, excel, channel, pattern, breako...
4	[prodigaltrad, accumul, finish, whop, 800, gai...

(g)



0	wipro wipro last 36 month daili stockmov histo...
1	smartmoneyempi1 still fresh join u win link bi...
2	sub 0 23 downtrend declin three consecut day a...
3	tata commun excel channel pattern breakout goo...
4	prodigaltrad accumul finish whop 800 gain past...

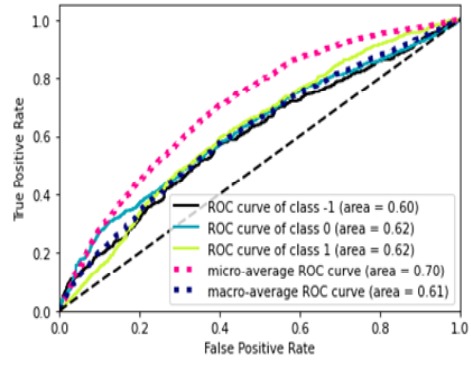
(h)

The performance of each feature extraction technique on 14 different classifiers can also be interpreted from Table 3. The Word2Vec feature extraction technique exhibits a higher performance on the RF classifier with an accuracy of 71%. The Doc2Vec obtains a higher accuracy of 83% for KNN classifier among all the 14 classifiers. FastText performs better with RC with 75% accuracy, and TF-IDF feature extraction performs better with SVM-linear kernel with an accuracy of 93%.

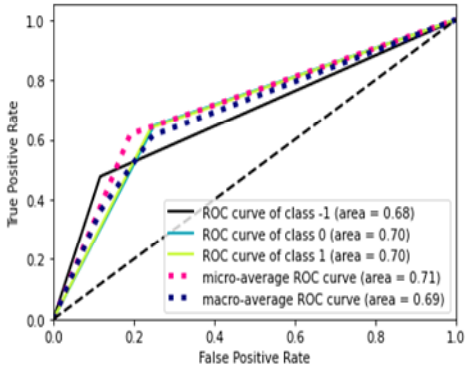
Table 3 Performance results of 14 classifiers

Classifiers	*WORD2VEC*				*DOC2VEC*				*FASTTEXT*				*TFIDF*				*HYBRID*			
	ACC	*PREC*	*REC*	*F1*	*ACC*	*PREC*	*REC*	*F1*	*ACC*	*PREC*	*REC*	*F1*	*ACC*	*PREC*	*REC*	*F1*	*ACC*	*PREC*	*REC*	*F1*
RFC	0.71	0.72	0.70	0.70	0.82	0.82	0.82	0.82	0.70	0.70	0.70	0.70	0.92	0.92	0.92	0.92	*0.94*	*0.94*	*0.94*	*0.94*
KNN	0.67	0.67	0.67	0.67	*0.83*	*0.82*	*0.83*	*0.82*	0.65	0.65	0.65	0.65	0.74	0.78	0.74	0.74	0.72	0.81	0.72	0.72
LR	0.50	0.48	0.50	0.44	0.78	0.76	0.78	0.77	0.49	0.42	0.49	0.45	0.90	0.9	0.90	0.90	*0.93*	*0.93*	*0.93*	*0.92*
GB	0.64	0.65	0.64	0.62	0.81	0.80	0.81	0.80	0.64	0.66	0.64	0.63	0.85	0.87	0.85	0.85	*0.87*	*0.89*	*0.87*	*0.87*
DT	0.62	0.62	0.62	0.62	0.76	0.77	0.76	0.77	0.61	0.61	0.61	0.61	0.92	0.92	0.92	0.92	*0.95*	*0.94*	*0.95*	*0.95*
SVC	0.50	0.42	0.50	0.45	0.78	0.76	0.78	0.77	0.48	0.41	0.48	0.43	0.93	0.93	0.93	0.93	*0.95*	*0.95*	*0.95*	*0.95*
ET	0.66	0.66	0.66	0.66	0.79	0.80	0.79	0.79	0.62	0.62	0.62	0.62	0.78	0.79	0.78	0.78	*0.80*	*0.81*	*0.80*	*0.80*
AB	0.58	0.58	0.58	0.57	0.78	0.76	0.78	0.77	0.54	0.54	0.54	0.52	0.82	0.84	0.82	0.82	*0.84*	*0.87*	*0.84*	*0.84*
DC	0.58	0.58	0.58	0.57	0.78	0.76	0.78	0.77	0.54	0.54	0.54	0.52	0.82	0.84	0.82	0.82	*0.84*	*0.87*	*0.84*	*0.84*
RC	0.51	0.52	0.51	0.47	0.80	0.77	0.75	0.77	0.75	0.42	0.50	0.46	0.91	0.91	0.91	0.91	*0.93*	*0.93*	*0.93*	*0.93*
GNB	0.47	0.52	0.47	0.48	*0.66*	*0.79*	*0.66*	*0.69*	0.45	0.48	0.45	0.44	0.59	0.66	0.59	0.60	0.62	0.78	0.62	0.65
BNB	0.51	0.54	0.51	0.51	0.69	0.76	0.69	0.72	0.49	0.49	0.49	0.47	0.85	0.84	0.85	0.84	*0.86*	*0.86*	*0.86*	*0.86*
LD	0.51	0.50	0.51	0.48	0.76	0.75	0.76	0.75	0.51	0.49	0.51	0.47	0.66	0.90	0.91	0.90	*0.92*	*0.92*	*0.92*	*0.92*
QD	0.67	0.70	0.67	0.66	0.70	0.73	0.70	0.71	0.49	0.64	0.49	0.45	0.42	0.85	0.86	0.85	*0.87*	*0.88*	*0.87*	*0.86*

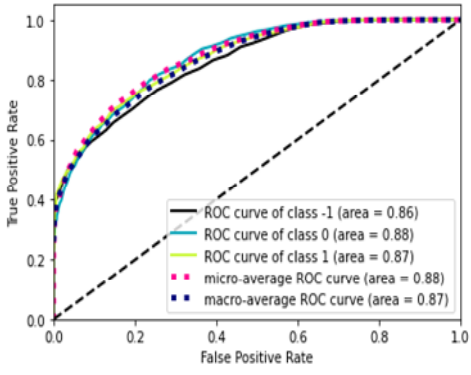
Figure 8 (a) ROC plot for SVM using Word2Vec (b) ROC plot for DT using Word2Vec (c) ROC plot for RFC using Word2Vec (d) ROC plot for SVM using Doc2Vec (e) ROC plot for DT using Doc2Vec (f) ROC plot for RFC using Doc2Vec (g) ROC plot for SVM using FastText (h) ROC plot for DT using FastText (i) ROC plot for RFC using FastText (j) ROC plot for SVM using TFIDF (k) ROC plot for DT using TFIDF (l) ROC plot for RFC using TFIDF (m) ROC plot for SVM using hybrid (n) ROC plot for DT using hybrid (o) ROC plot for RFC using hybrid (see online version for colours)



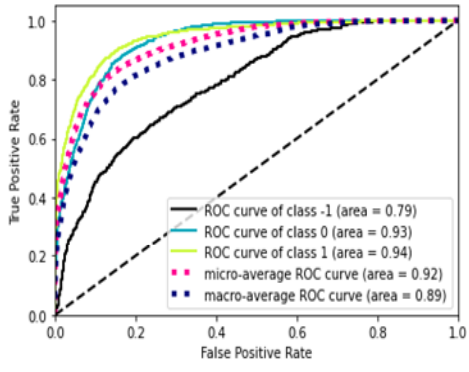
(a)



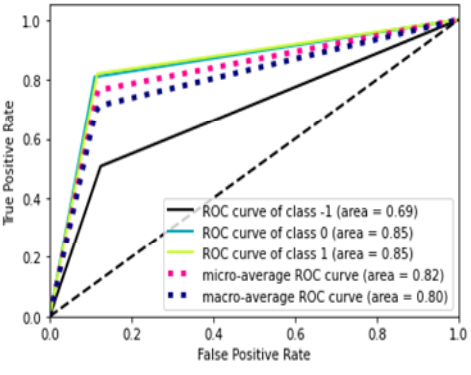
(b)



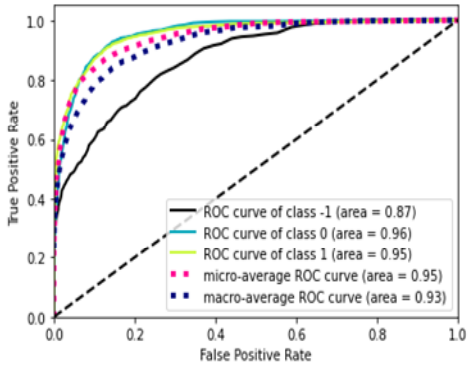
(c)



(d)

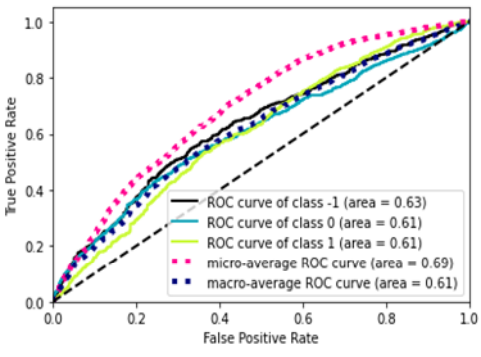


(e)

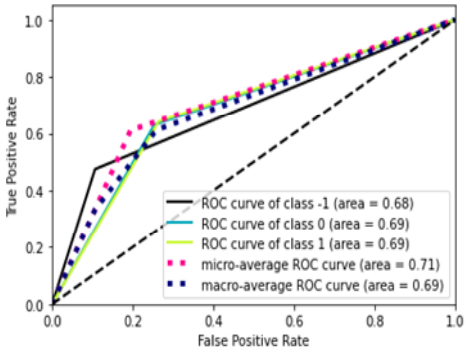


(f)

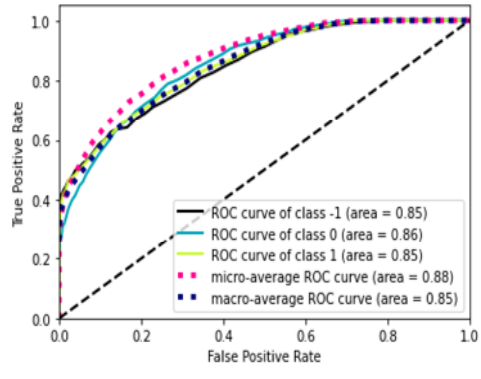
Figure 8 (a) ROC plot for SVM using Word2Vec (b) ROC plot for DT using Word2Vec (c) ROC plot for RFC using Word2Vec (d) ROC plot for SVM using Doc2Vec (e) ROC plot for DT using Doc2Vec (f) ROC plot for RFC using Doc2Vec (g) ROC plot for SVM using FastText (h) ROC plot for DT using FastText (i) ROC plot for RFC using FastText (j) ROC plot for SVM using TFIDF (k) ROC plot for DT using TFIDF (l) ROC plot for RFC using TFIDF (m) ROC plot for SVM using hybrid (n) ROC plot for DT using hybrid (o) ROC plot for RFC using hybrid (continued) (see online version for colours)



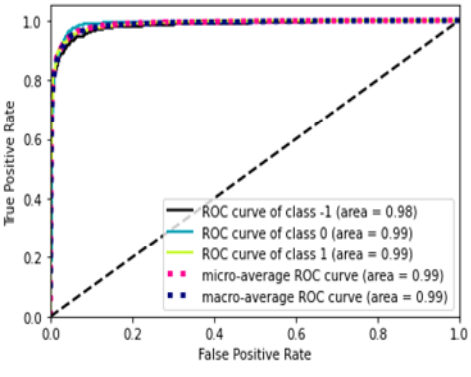
(g)



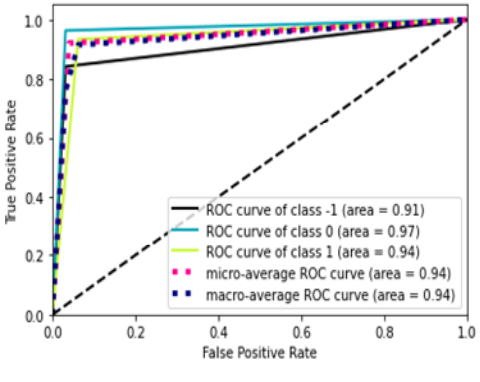
(h)



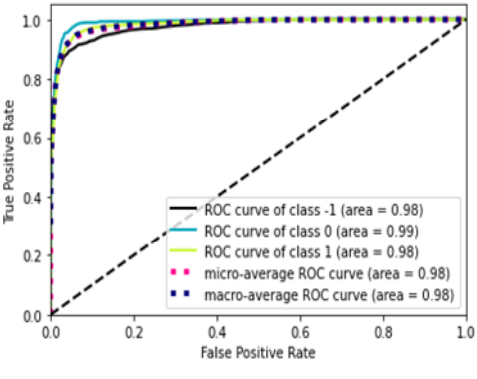
(i)



(j)

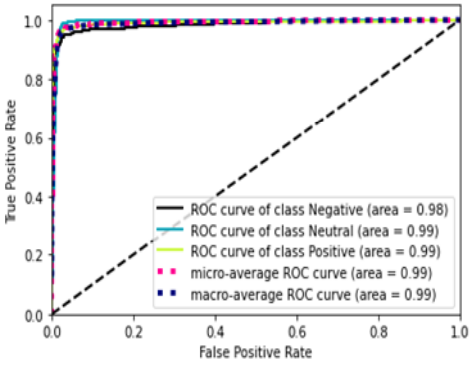


(k)

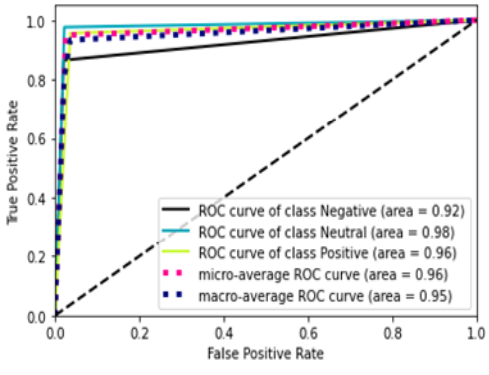


(l)

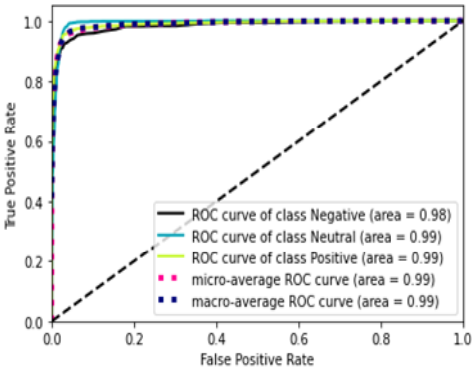
Figure 8 (a) ROC plot for SVM using Word2Vec (b) ROC plot for DT using Word2Vec (c) ROC plot for RFC using Word2Vec (d) ROC plot for SVM using Doc2Vec (e) ROC plot for DT using Doc2Vec (f) ROC plot for RFC using Doc2Vec (g) ROC plot for SVM using FastText (h) ROC plot for DT using FastText (i) ROC plot for RFC using FastText (j) ROC plot for SVM using TFIDF (k) ROC plot for DT using TFIDF (l) ROC plot for RFC using TFIDF (m) ROC plot for SVM using hybrid (n) ROC plot for DT using hybrid (o) ROC plot for RFC using hybrid (continued) (see online version for colours)



(m)



(n)



(o)

The PR and ROC curves for the top 3 classifiers (SVM, DT and RFC) are presented in Figures 8(a)–8(o) and 9(a)-9(o). The area under the micro-averaged ROC curve and PR curve helps in computing the overall model performance. A higher AUC-ROC and AUC-PR (close to 0) indicates a better performing classifier. In this study, RFC is the best-performing algorithm with AUC equal to 0.88, 0.95, and 0.88 for Word2Vec, Doc2Vec, and FastText, respectively. SVM performs best with TF-IDF (AUC = 0.99), and both SVM and RFC perform well in the case of the hybrid technique (AUC = 0.99). The precision-recall curve shows RFC as a better classifier for Word2Vec, Doc2Vec, and FastText feature extraction techniques with AUC equal to 0.84, 0.95 and 0.83, respectively. It also shows SVM as the best classifier for the hybrid technique (AUC = 0.99). Given the results, it can be inferred that among the feature extraction

techniques, the classifiers using the hybrid technique yields higher performance. Specifically, SVM demonstrates a robust performance with higher AUC-ROC and AUC-PR for the hybrid technique. Thus, it can be concluded that SVM with a hybrid feature extraction technique (TF-IDF+Doc2Vec) exhibits higher performance in the SA of Twitter data related to the stock market.

Figure 9 (a) PR plot for SVM using Word2Vec (b) PR plot for DT using Word2Vec (c) PR plot for RFC using Word2Vec (d) PR plot for SVM using Doc2Vec (e) PR plot for DT using Doc2Vec (f) PR plot for RFC using Doc2Vec (g) PR plot for SVM using FastText (h) PR plot for DT using FastText (i) PR plot for RFC using FastText (j) PR plot for SVM using TFIDF (k) PR plot for DT using TFIDF (l) PR plot for RFC using TFIDF (m) PR plot for SVM using hybrid (n) PR plot for DT using hybrid (o) PR plot for RFC using hybrid (see online version for colours)

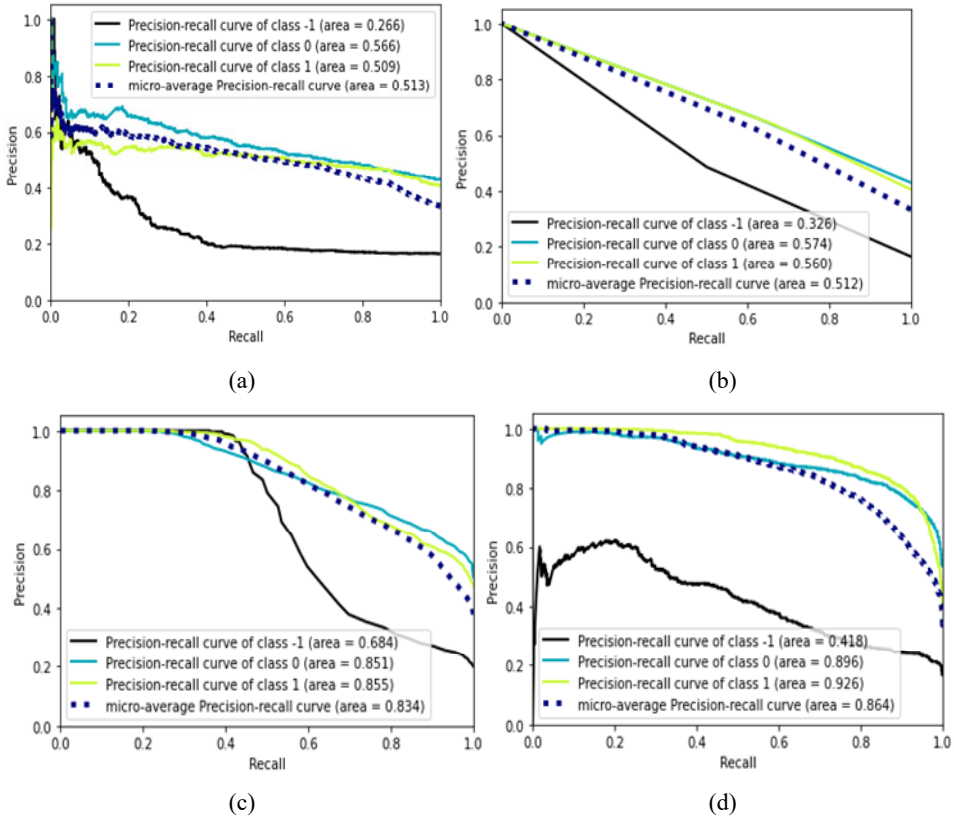
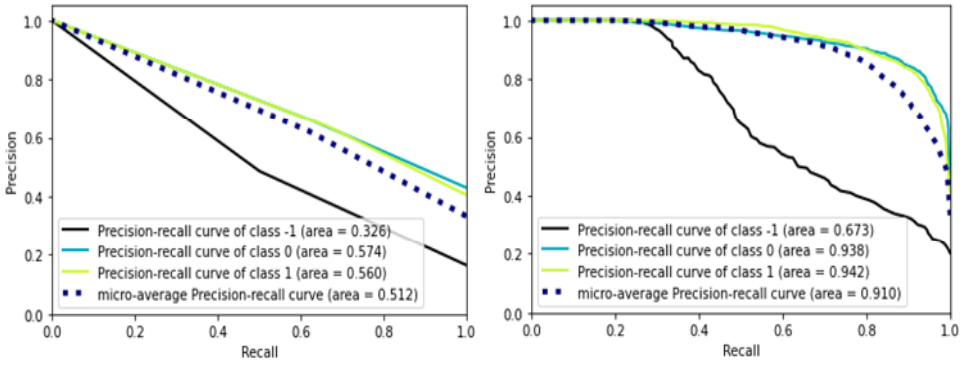
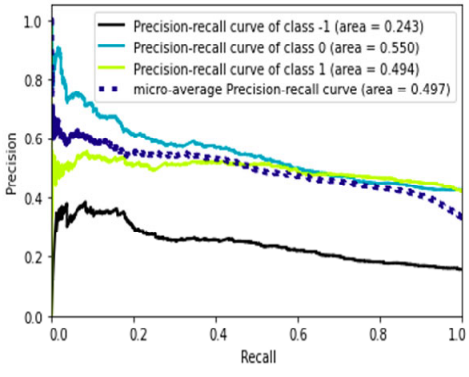


Figure 9 (a) PR plot for SVM using Word2Vec (b) PR plot for DT using Word2Vec (c) PR plot for RFC using Word2Vec (d) PR plot for SVM using Doc2Vec (e) PR plot for DT using Doc2Vec (f) PR plot for RFC using Doc2Vec (g) PR plot for SVM using FastText (h) PR plot for DT using FastText (i) PR plot for RFC using FastText (j) PR plot for SVM using TFIDF (k) PR plot for DT using TFIDF (l) PR plot for RFC using TFIDF (m) PR plot for SVM using hybrid (n) PR plot for DT using hybrid (o) PR plot for RFC using hybrid (continued) (see online version for colours)

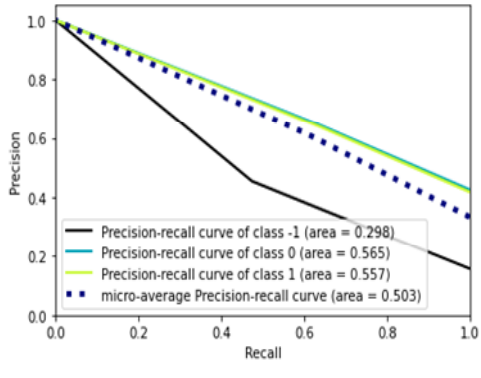


(e)

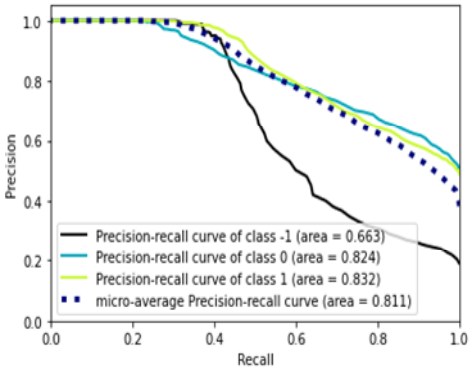
(f)



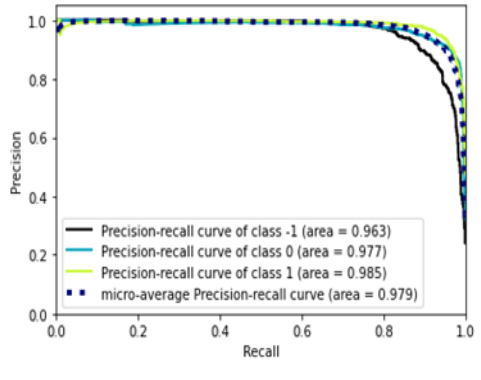
(g)



(h)

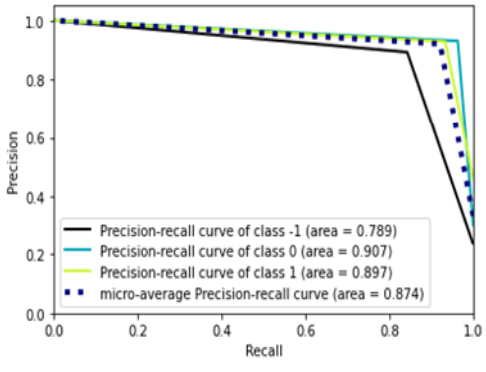


(i)

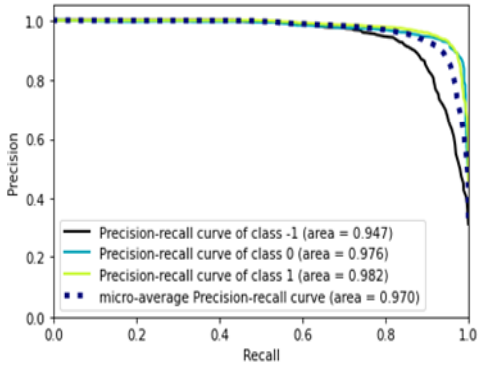


(j)

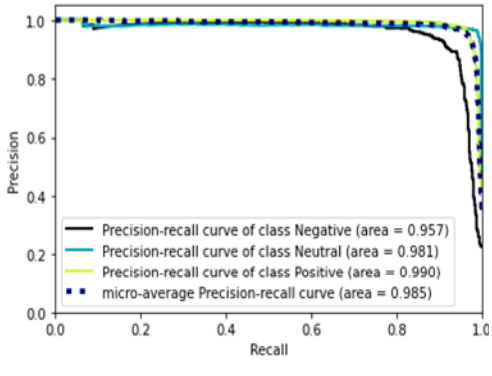
Figure 9 (a) PR plot for SVM using Word2Vec (b) PR plot for DT using Word2Vec (c) PR plot for RFC using Word2Vec (d) PR plot for SVM using Doc2Vec (e) PR plot for DT using Doc2Vec (f) PR plot for RFC using Doc2Vec (g) PR plot for SVM using FastText (h) PR plot for DT using FastText (i) PR plot for RFC using FastText (j) PR plot for SVM using TFIDF (k) PR plot for DT using TFIDF (l) PR plot for RFC using TFIDF (m) PR plot for SVM using hybrid (n) PR plot for DT using hybrid (o) PR plot for RFC using hybrid (continued) (see online version for colours)



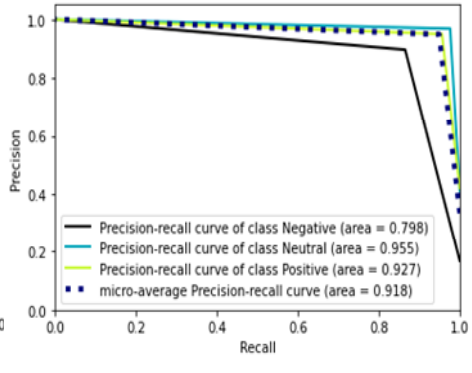
(k)



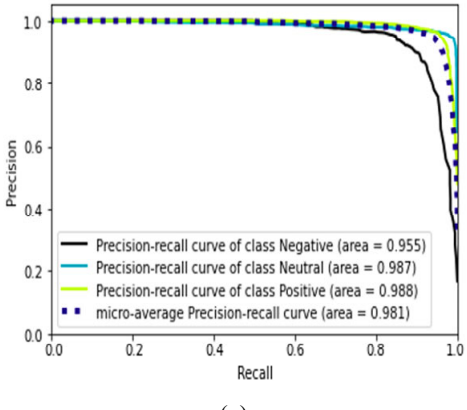
(l)



(m)



(n)



(o)

5 Conclusions

Stock market forecasting plays a substantial role in determining the optimal investment strategies and efficient financing plans. SA facilitates the possibility of predicting the price and movement of the stock market precisely. This study analyses the sentiments of the Twitter data related to the global stock market using 14 classifiers: RFC, KNN, LR, GB, DT, SVM, ET, AB, DC, RC, MNB, BNB, LD, and QD and five feature extraction techniques: TF-IDF, Word2Vec, Doc2Vec, FastText, and a hybrid technique using TF-IDF and Doc2vec.The results indicates that the SVM with hybrid feature extraction technique exhibits the highest performance. In addition, the hybrid technique using TF-IDF and Doc2Vec exhibits higher performance compared to the other four featurisation techniques for 12 classifiers. Thus, this novelty in the feature extraction technique can outperform the others in the field of SA of stock tweets for a small dataset.

The proposed model helps in calculating investor sentiments with higher accuracy which in turn can provide an early insight to the investors regarding the general market sentiment. It can further influence the investment decisions and risk management approaches of an investor. The financial analysts can incorporate the sentiments generated from the proposed model to incorporate with technical and fundamental analysis to improve the stock price prediction. The highly accurate investor sentiments helps the stakeholders like traders, credit rating agencies and investment banks in capitalising on the short-term sentiment driven market. Future work can include the application of deep learning classifiers and hybrid feature extraction techniques on a large Twitter dataset related to stocks. Further, an in-depth study can be conducted on the SA on the stock market using the investor's sentiment derived from sentiment indexes and other social media platforms.

References

Adhikari, S., Thapa, S., Naseem, U., Lu, H.Y., Bharathy, G. and Prasad, M. (2023) 'Explainable hybrid word representations for sentiment analysis of financial news', *Neural Networks*, Vol. 164, pp.115–123, https://doi.org/10.1016/j.neunet.2023.04.011.

Agarwal, N., Sikka, G. and Awasthi, L. K. (2020) 'Enhancing web service clustering using length feature weight method for service description document vector space representation', *Expert Systems with Applications*, Vol. 161, p.113682, https://doi.org/10.1016/j.eswa.2020.113682.

Al Silni Ahmed, E.R. and Goyal, S.B. (2021) 'Impact of technical parameters for short- and long-term analysis of stock behavior', *Materials Today: Proceedings*, https://doi.org/10.1016/j.matpr.2021.05.474.

Al-Dulaimi, K., Chandran, V., Nguyen, K., Banks, J. and Tomeo-Reyes, I. (2019) 'Benchmarking HEP-2 specimen cells classification using linear discriminant analysis on higher order spectra features of cell shape', *Pattern Recognition Letters*, Vol. 125, pp.534–541, https://doi.org/10.1016/j.patrec.2019.06.020

Almalis, I., Kouloumpris, E. and Vlahavas, I. (2022) 'Sector-level sentiment analysis with deep learning', *Knowledge-Based Systems*, Vol. 258, p.109954, https://doi.org/10.1016/j.knosys.2022.109954.

Alsmadi, I. and Hoon, G.K. (2018) 'Term weighting scheme for short-text classification: twitter corpuses', *Neural Computing and Applications*, Vol. 31, No. 8, pp.3819–3831, https://doi.org/10.1007/s00521-017-3298-8.

Ballı, S. and Karasoy, O. (2019) 'Development of content-based SMS classification application by using word2vec-based feature extraction', *IET Software*, Vol. 13, No. 4, pp.295–304, https://doi.org/10.1049/iet-sen.2018.5046.

Beitia-Antero, L., Yáñez, J. and de Castro, A.I. (2018) 'On the use of logistic regression for stellar classification', *Experimental Astronomy*, Vol. 45, No. 3, pp.379–395, https://doi.org/10.1007/s10686-018-9591-4.

Carosia, A.E., Coelho, G.P. and Silva, A.E. (2019) 'Analyzing the Brazilian financial market through Portuguese sentiment analysis in social media', *Applied Artificial Intelligence*, Vol. 34, No. 1, pp.1–19, https://doi.org/10.1080/08839514.2019.1673037.

Chen, Q. and Sokolova, M. (2021) 'Specialists, scientists, and sentiments: Word2vec and doc2vec in analysis of scientific and medical texts', *SN Computer Science*, Vol. 2, No. 5, https://doi.org/10.1007/s42979-021-00807-1.

Dey, R.K. and Das, A.K. (2023) 'Modified term frequency-inverse document frequency based deep hybrid framework for sentiment analysis', *Multimedia Tools and Applications*, https://doi.org/10.1007/s11042-023-14653-1.

Dodia, S., Edla, D.R., Bablani, A., Ramesh, D. and Kuppili, V. (2019) 'An efficient EEG based deceit identification test using wavelet packet transform and linear discriminant analysis', *Journal of Neuroscience Methods*, Vol. 314, pp.31–40, https://doi.org/10.1016/j.jneumeth.2019.01.007.

Elnagar, A., Khalifa, Y.S. and Einea, A. (2017) 'Hotel Arabic-reviews dataset construction for sentiment analysis applications', *Intelligent Natural Language Processing: Trends and Applications*, pp.35–52, https://doi.org/10.1007/978-3-319-67056-0_3.

Ganesh, V., Kumar, H.S. and Sivasankar, E. (2021) 'Financial sentiment analysis: a study of feature engineering methodologies', in Viswanathan, V. (Ed.): *Soft Computing and Signal Processing*, Vol. 1325, pp.225–240, Essay, Springer.

Gao, R., Cui, S., Xiao, H., Fan, W., Zhang, H. and Wang, Y. (2022) 'Integrating the sentiments of multiple news providers for stock market index movement prediction: a deep learning approach based on evidential reasoning rule', *Information Sciences*, Vol. 615, pp.529–556, https://doi.org/10.1016/j.ins.2022.10.029.

Ghosal, S. and Jain, A. (2023) 'Depression and suicide risk detection on social media using FastText embedding and XGBoost classifier', *Procedia Computer Science*, Vol. 218, pp.1631–1639, https://doi.org/10.1016/j.procs.2023.01.141.

Goel, H. and Singh, N.P. (2021) 'Dynamic prediction of Indian stock market: an artificial neural network approach', *International Journal of Ethics and Systems*, Vol. 38, No. 1, pp.35–46, https://doi.org/10.1108/ijoes-11-2020-0184.

Groß-Klußmann, A., König, S. and Ebner, M. (2019) 'Buzzwords build momentum: global financial twitter sentiment and the aggregate stock market', *SSRN Electronic Journal*, https://doi.org/10.2139/ssrn.3412908.

Gruszka, J. and Szwabiński, J. (2021) 'Advanced strategies of portfolio management in the Heston market model', *Physica A: Statistical Mechanics and its Applications*, Vol. 574, p.125978, https://doi.org/10.1016/j.physa.2021.125978.

Gu, C. and Kurov, A. (2018) 'Informational role of social media: evidence from twitter sentiment', *SSRN Electronic Journal*, https://doi.org/10.2139/ssrn.3206093.

Hoseinzade, E. and Haratizadeh, S. (2019) 'CNNpred: CNN-based stock market prediction using a diverse set of variables', *Expert Systems with Applications*, Vol. 129, pp.273–285, https://doi.org/10.1016/j.eswa.2019.03.029.

Hu, G., Yin, C., Wan, M., Zhang, Y. and Fang, Y. (2020) 'Recognition of diseased pinus trees in UAV images using deep learning and AdaBoost classifier', *Biosystems Engineering*, Vol. 194, pp.138–151, https://doi.org/10.1016/j.biosystemseng.2020.03.021.

Hu, T. and Tripathi, A. (2017) 'The performance evaluation of machine learning classifiers on financial microblogging platforms', *Lecture Notes in Business Information Processing*, pp.74–83, https://doi.org/10.1007/978-3-319-69644-7_7.

Jing, N., Wu, Z. and Wang, H. (2021) 'A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction', *Expert Systems with Applications*, Vol. 178, p.115019, https://doi.org/10.1016/j.eswa.2021.115019.

Karasu, S., Altan, A., Bekiros, S. and Ahmad, W. (2020) 'A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series', *Energy*, Vol. 212, p.118750, https://doi.org/10.1016/j.energy.2020.118750.

Khatua, A., Khatua, A. and Cambria, E. (2019) 'A tale of two epidemics: contextual word2vec for classifying twitter streams during outbreaks', *Information Processing & Management*, Vol. 56, No. 1, pp.247–257, https://doi.org/10.1016/j.ipm.2018.10.010.

Kiala, Z., Mutanga, O., Odindi, J. and Masemola, C. (2021) 'Optimal window period for mapping parthenium weed in South Africa, using high temporal resolution imagery and the extratrees classifier', *Biological Invasions*, Vol. 23, No. 9, pp.2881–2892, https://doi.org/10.1007/s10530-021-02544-1.

Kolasani, S.V. and Assaf, R. (2020) 'Predicting stock movement using sentiment analysis of twitter feed with neural networks', *Journal of Data Analysis and Information Processing*, Vol. 8, No. 4, pp.309–319, https://doi.org/10.4236/jdaip.2020.84018.

Kumbure, M.M., Luukka, P. and Collan, M. (2020) 'A new fuzzy k-nearest neighbor classifier based on the Bonferroni mean', *Pattern Recognition Letters*, Vol. 140, pp.172–178, https://doi.org/10.1016/j.patrec.2020.10.005.

Li, M., Chen, L., Zhao, J. and Li, Q. (2021) 'Sentiment analysis of Chinese stock reviews based on Bert model', *Applied Intelligence*, Vol. 51, No. 7, pp.5016–5024, https://doi.org/10.1007/s10489-020-02101-8.

Li, X., Wu, P. and Wang, W. (2020) 'Incorporating stock prices and news sentiments for stock market prediction: a case of Hong Kong', *Information Processing & Management*, Vol. 57, No. 5, p.102212, https://doi.org/10.1016/j.ipm.2020.102212.

Liu, M., Cao, Z., Zhang, J., Wang, L., Huang, C. and Luo, X. (2020) 'Short-term wind speed forecasting based on the Jaya-SVM model', *International Journal of Electrical Power & Energy Systems*, Vol. 121, p.106056, https://doi.org/10.1016/j.ijepes.2020.106056.

Liu, Q., Lee, W-S., Huang, M. and Wu, Q. (2023) 'Synergy between stock prices and investor sentiment in social media', *Borsa Istanbul Review*, Vol. 23, No. 1, pp.76–92, https://doi.org/10.1016/j.bir.2022.09.006.

Luo, X. (2021) 'Efficient English text classification using selected machine learning techniques', *Alexandria Engineering Journal*, Vol. 60, No. 3, pp.3401–3409, https://doi.org/10.1016/j.aej.2021.02.009.

Ma, Y., Mao, R., Lin, Q., Wu, P. and Cambria, E. (2023) 'Multi-source aggregated classification for stock price movement prediction', *Information Fusion*, Vol. 91, pp.515–528, https://doi.org/10.1016/j.inffus.2022.10.025.

Mishev, K., Gjorgjevikj, A., Stojanov, R., Mishkovski, I., Vodenska, I., Chitkushev, L. and Trajanov, D. (2019) 'Performance evaluation of word and sentence embeddings for finance headlines sentiment analysis', *Communications in Computer and Information Science*, pp.161–172, https://doi.org/10.1007/978-3-030-33110-8_14.

Picasso, A., Merello, S., Ma, Y., Oneto, L. and Cambria, E. (2019) 'Technical analysis and sentiment embeddings for market trend prediction', *Expert Systems with Applications*, Vol. 135, pp.60–70, https://doi.org/10.1016/j.eswa.2019.06.014.

Qiu, Y., Song, Z. and Chen, Z. (2022) 'Short-term stock trends prediction based on sentiment analysis and machine learning', *Soft Computing*, Vol. 26, No. 5, pp.2209–2224, https://doi.org/10.1007/s00500-021-06602-7.

Renault, T. (2019) 'Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages', *Digital Finance*, Vol. 2, Nos. 1–2, pp.1–13, https://doi.org/10.1007/s42521-019-00014-x.

Shi, Y., Zheng, Y., Guo, K. and Ren, X. (2020) 'Stock movement prediction with sentiment analysis based on deep learning networks', *Concurrency and Computation: Practice and Experience*, Vol. 33, No. 6, https://doi.org/10.1002/cpe.6076.

Souma, W., Vodenska, I. and Aoyama, H. (2019) 'Enhanced news sentiment analysis using deep learning methods', *Journal of Computational Social Science*, Vol. 2, No. 1, pp.33–46, https://doi.org/10.1007/s42001-019-00035-x.

Thormann, M-L., Farchmin, J., Weisser, C., Kruse, R-M., Säfken, B. and Silbersdorff, A. (2021) 'Stock price predictions with LSTM neural networks and twitter sentiment', *Statistics, Optimization & Information Computing*, Vol. 9, No. 2, pp.268–287, https://doi.org/10.19139/soic-2310-5070-1202.

Toğaçar, M., Ergen, B. and Cömert, Z. (2020) 'Classification of white blood cells using deep features obtained from convolutional neural network models based on the combination of feature selection methods', *Applied Soft Computing*, Vol. 97, p.106810, https://doi.org/10.1016/j.asoc.2020.106810.

Tzeng, K.Y. (2022) 'The ability of US macroeconomic variables to predict Asian financial market returns', *International Journal of Finance & Economics*, https://doi.org/10.1002/ijfe.2606.

Wu, S., Liu, Y., Zou, Z. and Weng, T-H. (2021) 'S_I_LSTM: dtock price prediction based on multiple data sources and sentiment analysis', *Connection Science*, Vol. 34, No. 1, pp.44–62, https://doi.org/10.1080/09540091.2021.1940101.

Xue, L., Wang, H., Wang, F. and Ma, H. (2021) 'Sentiment analysis of stock market investors and its correlation with stock price using maximum entropy', *Computer and Information Science*, Summer, pp.29–44, https://doi.org/10.1007/978-3-030-79474-3_3.

Yang, J.S., Zhao, C.Y., Yu, H.T. and Chen, H.Y. (2020) 'Use GBDT to predict the stock market', *Procedia Computer Science*, Vol. 174, pp.161–171, https://doi.org/10.1016/j.procs.2020.06.071.

Yilmaz, S. and Toklu, S. (2020) 'A deep learning analysis on question classification task using word2vec representations', *Neural Computing and Applications*, Vol. 32, No. 7, pp.2909–2928, https://doi.org/10.1007/s00521-020-04725-w.

Zhang, D. and Tang, P. (2023) 'Forecasting European union allowances futures: the role of technical indicators', *Energy*, Vol. 270, p.126916, https://doi.org/10.1016/j.energy.2023.126916.

ZhengWei, H., JinTao, M., YanNi, Y., Jin, H. and Ye, T. (2022) 'Recommendation method for academic journal submission based on doc2vec and xgboost', *Scientometrics*, Vol. 127, No. 5, pp.2381–2394, https://doi.org/10.1007/s11192-022-04354-1.

Zong, H., Wu, S. and Wei, G. (2022) 'Sentiment analysis of news on the stock market', *Lecture Notes in Operations Research*, pp.284–296, https://doi.org/10.1007/978-981-16-8656-6_27.

Zulfiker, M.S., Kabir, N., Amin, A., Chakraborty, P. and Mahfujur, M. (2020) 'Predicting students' performance of the private universities of Bangladesh using machine learning approaches', *International Journal of Advanced Computer Science and Applications*, Vol. 11, No. 3, https://doi.org/10.14569/ijacsa.2020.0110383.

Nomenclature

AB	AdaBoost
ACC	Accuracy
API	Application programming interface
ARIMA	Autoregressive integrated moving average
ARMA	Auto regressive moving average
Att-BiLSTM	Attention-based bidirectional long short-term memory
AUC	Area under curve
Bi-GRU	Bidirectional GRU
BNB	Binomial naive Bayes
CBOW	Continuous bag of words
CCI	Commodity channel index
CNN	Convolution neural networks
CPI	Consumer price index
DC	Dummy classifier
DL	Deep learning
DT	Decision tree
EMA	Exponential moving average
ERT	Extremely randomised trees
ET	Extra trees
GARCH	Generalised autoregressive conditional heteroskedasticity
GB	Gradient boosting
GBDT	Gradient-boosted decision trees
GNB	Gaussian naïve Bayes
GRU	Gated recurrent units
HTML	Hypertext markup language
IIP	Index of industrial production
KNN	K-nearest neighbours
LD	Linear discriminant
LR	Logistic regression
LSTM	Long short-term memory
LTIR	Long-term interest rate
MACD	Moving average convergence divergence
MaxEnt	Maximum entropy
MFI	Money flow index
ML	Machine learning
MLP	Multilayer perceptron
MOM	Momentum indicator
NB	Naïve Bayes
NLTK	Natural language toolkit

Nomenclature (continued)

PPO	Percentage price oscillator
PREC	Precision
QDA	Quadratic discriminant analysis
RC	Ridge classifier
RF	Random forest
RNN	Recurrent neural network
RSI	Relative strength index
SA	Sentiment analysis
SGD	Stochastic gradient descent
SMA	Simple moving average
SVM	Support vector classifier
SVM	Support vector machine
TF-IDF	Term frequency inverse document frequency
TR	True range
TRIX	Triple exponential average
ULTOSC	Ultimate oscillator
URL	Uniform resource locators
WMA	Weighted moving average
XG-Boost	Extreme gradient boosting