Stock Market Price Prediction through News Sentiment Analysis & Ensemble Learning

Santosh Kumar Bharti
Dept. of CSE
School of Technology
PDEU, Gandhinagar

Priyansh Tratiya
Dept. of CSE
School of Technology
PDEU, Gandhinagar

Rajeev Kumar Gupta
Dept. of CSE
School of Technology
PDEU, Gandhinagar

2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC) | 978-1-6654-9056-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/iSSSC56467.2022.10051623

Abstract— **Stock market trends tend to vary according to real-life situations and market sentiments. The growing interest in the Financial-Technology sector has culminated in more research being carried out in this area. For sentiments, news headlines have proven to be a major source and can be used without the further context of the news article. Most of the research for stock market price prediction has been carried out based only on daily stock data or using standard machine learning algorithms. Hence, in this research project, we find the importance of using the sentiment data extracted from news headlines and daily stock data combined for the prediction of trends and stock prices. To carry out market trends prediction we propose the use of the ensemble technique XGBoost, whereas, for price prediction time series, LSTM (Long Short-Term Memory) cells are used. Our results indicate that combining both data as input using ensemble techniques and time series models gave much better and more accurate output than the standard methods that have been employed before.**

Keywords— Sentiment Analysis, Stock Market, LSTM, XGBoost, Price Prediction, Trend Analysis

I. INTRODUCTION

The Stock Market is the confluence of buyers and sellers of stocks (also referred to as shares) which represents the ownership of a particular company. These also include securities that are recorded on the public stock exchanges, however, securities are sold and traded privately. One example of it can be the stocks of any private company which are sold to financial backers via equity crowdfunding platforms. Stock market investment is generally done through some stock brokers or online electronic trading marketplaces. Investment in the stock market is generally made with an economical investment plan and strategy in mind considering various trends and economic outcomes. There seems to be a new wave of algorithmic trading forming up. According to Credit Suisse Group AG's report, quantitative funds run by automated systematic trading techniques, also known as investing robots or bots, are the fastest growing type of funds. According to JPMorgan Chase & Co. reports, human discretionary investors currently account for just 10% of the trading volume. This project aims to project the flow of stock prices based on running scenarios and events which affect the industries' economic performance using XGBoost (Classification) and LSTM (Regression) models. A sub-project of its own will also be developed, an automated web scraper to collect any companies' daily stock data and news pertaining to it.

II. EXISTING SOLUTION

Stock Market price prediction is done with the use of machine learning using various features and algorithms for a long time now. SVM is widely used for text classification and sentiment analysis. Data acquired from SVM and historical stock market data can be used together to forecast stock market prices. Xie Y and Jiang H. (2019) [11] applied this concept to forecasting Chinese stock market prices. Some research papers use Naive Bayes Classifier, Neural Networks, and multiple other technologies to achieve the aim. Detailed technical analysis and their achieved performance is briefly discussed in Literature Review.

Plenty of work has been done in the field of Stock market prediction as well as Sentiment Analysis in past years. Deep learning and neural networks provide a very good advantage in the prediction process. Though, currently, there is no notable work done on Stock Price prediction using Ensemble learning or XGBoost as it is a comparatively very new algorithm and is currently in its comparatively nascent phase. Stock market prices can be considered as time series data and therefore the fact cannot be neglected that previous data points (prices) impact a current data point, conventional models and machine learning algorithms cannot use the previous data points in prediction and Neural Networks and Deep Learning algorithms can take only a few data points into account for the prediction of the current data point because of the vanishing gradient. The Time Series prediction model LSTM can help deal with time series data because of their additional input which takes the output of the previous data point with a weight factor of 1. Therefore, it can take any number of previous data points into account for the prediction of the current data point..

III. LITERATURE REVIEW

Many papers have studied the connection between the Stock market price trends and movement and the people's sentiments gathered from different sources, some of the relevant studies are mentioned in this section.

TABLE I. DETAILED SUMMARY OF EXISTING LITERATURE, THEIR APPLIED TECHNOLOGIES AND OUTCOME

Literature	Data Used	Method	Extracted Features from Textual Data	Results
Nayak et al. [1]	Twitter, Yahoo Finance, News	Boosted Decision Tree, Logistic Regression, SVM, Sentiment Analysis	Positive and Negative Sentiment	Accuracy for Bank Sector – 0.548, for Mining Sector – 0.76, for Oil Sector – 0.769
Nemes and Kiss [2]	Economic news,	TextBlob, NLTK-VADER	Positive, Negative, Neutral	Concluded that the sentiments

978-1-6654-9056-6/22/$31.00 @2022 IEEE

	Yahoo Finance	Lexicon, RNN and BERT		affect the stock market. Did not develop a predictive model
Vijh et al.[3]	Yahoo Finance	Aritificial Neural Network, Random Forest	No sentiment analysis carried out	Lowest RMSE – 0.42 Highest RMSE – 3.40
Martin [4]	Twitter, CAC40 french stock data	Neural Network	Aggregate Sentiment Score	Accuracy Score - 80%
Zhang et al. [5]	Financial news data, Shanghai Composite stocks data and Xueqiu data	Artificial Neural Networks	Positive polarity and Negative Polarity	Accuracy Score - 60%
Zhang et al.[5]	Hang Seng300 Index and News and Posts from Sina Weibo	Sentiment dictionary and double layer Recurrent NN	Keywords of two types - Positive and Negative	MAE - 0.625 MAPE - 9.381 RMSE - 0.803
Shastri et al.[6]	Daily News headlines and daily Apple's stocks data	Multi-level perceptron artificial neural network (ANN)	Sentiment score	MAPE - 8.21 Accuracy Score - 98%
Kolasani and Assaf [7]	Tweets and Historical Data from Yahoo Finance	Support Vector Regression	Positive and Negative Sentiments	Accuracy Score – 83% Lowest RMSE – 1.37
Khedr and Yaseen[8]	Daily index data of 3 random NASDAQ companies and Financial News	K Nearest Neighbours and Naive Bayes	Positivity, Negativity and Equal sentiments	Accuracy Score - 90%
Li et al. [9]	Forum posts of investors and Daily CSI300 stocks index data	Naive bayes and LSTM	Positive, Negative and Neutral Sentiments	Accuracy Score - 87.86%

IV. Proposed Methodology

In this research paper, we have three main subsections for our proposed methodology which are as follows.

A. Sentiment Analysis

VADER Lexicon is useful to classify polarity which includes positive, negative, neutral, compound (normalized values of three attributes before), and the strength of these polarities as well. VADER Lexicon uses a dictionary that associates lexical features with emotion intensity levels which is also known as sentiment scores. VADER calculates the sentiment score by aggregating the emotion intensity of every word in the text. VADER also has an understanding of the context of the words in the text like "did not love" as a negative sentiment. It can also understand the weight of punctuation and capitalization like "ENJOY". The only drawback of the VADER lexicon is that it does not determine if the sentence is subjective or objective, meaning it cannot tell if the sentence provided to it is a fact or an opinion.

To find the subjectivity and objectivity of the headlines, a classification model was created using the PyTorch neural networks module. It was fed the pre-classified data of 5000 subjective and 5000 objective sentences retrieved from the Cornell University data library [12] to train and test. The data is converted into the vector form using the GloVe (Global Vectors for Word Representation) data file [13] which is a collection of words and their vector forms. The model consists of three 1D Convolutional layers having ReLu as their output activation function. Followed by a single dropout layer having 0.5 dropout probability.

Once the model is good enough to classify the unseen data it is exported and used to classify the news headlines' subjectivity and objectivity. Every single day we have multiple headlines and therefore the combined subjectivity and objectivity of all the headlines is calculated by an aggregate average method and is assigned to the day's Sub_Obj score which can be used as a parameter for the prediction model.

B. XGBoost for Classification

XGBoost, also known as "Extreme Gradient Boosting", where the term "Gradient Boosting" was first mentioned in the paper by Friedman(2001)[14] named "Greedy Function Approximation: A Gradient Boosting Machine". The aim of any boosting algorithm is to build multiple prediction trees sequentially in a manner that every tree tries to reduce the errors of the tree created before it. Each new tree gets the error values of its predecessor and learns from them and then updates these error values. Therefore, trees that are going to be built after the current tree will learn from the updated values of the residual errors. The single trees also known as the base learners in this algorithm are weak learners with high bias and low predictive power. The predictive power is only a little bit better than random guessing. Every base learner contains some important information for prediction which allows the boosting algorithm to create a strong learner by merging all of the base learners. Bias and variance both are pretty low in this final strong learner.

Ensemble techniques like Random Forest use bagging techniques where all the base learner trees are grown and maximized to their full height but XGBoost uses Boosting ensemble technique in which the base learners are trained with very fewer splits. These trees are very short in height and can be easily interpreted. Validation techniques such as K-fold cross-validation can be useful to find optimal parameters like the gradient boosting learning rate or a number of trees and iterations.

Below is a comparison of different models for implementation of Random Forests with default hyperparameter values against XGBoost.

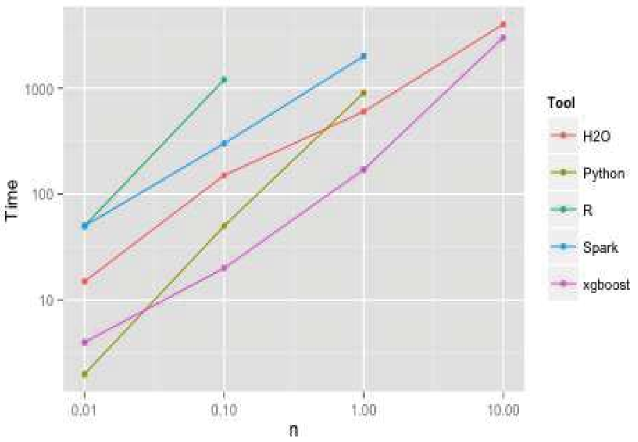


Fig 1. Performance comparison of XGBoost and Random Forest Algorithm on different Languages (H2O, Python, R, Spark)

The XGBoost prediction model was enhanced by optimizing the hyperparameters n_estimators and max_depth. For this purpose, an iterator was used with standard and most used values of these parameters and was run on the test data set to find the optimal values of these parameters.

To find the impact of all the extracted features on the prediction we plot a bar graph using Plotly and feature importance matrices for XGBoost and Random Forest Classifiers. Snippets of the results are shown in Figure 2.

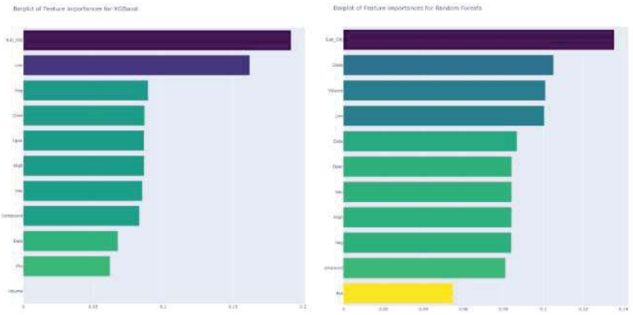


Fig 2. Graphical representation of important features for prediction in XGBoost(Left) and Random Forest Classifier(Right)

Figure 2. shows that the Subjectivity and Objectivity (Sub_Obj) of news headlines play a major role in stock market price trend prediction for both XGBoost and Random Forest prediction models. Sub_Obj is followed by low stock prices of the previous day (low) for XGBoost and by closing stock prices of the previous day (close) for the Random Forest model. Both models are trained in different manners. Therefore, the impact of different features is different on prediction. For example, the Volume of previous day stock has almost no impact in the prediction making of XGBoost but is the third most impactful feature in decision-making of the Random Forest model.

C. *LSTM for Regression*

LSTM stands for "Long Short Term Memory". It is a unique kind of Recurrent Neural Network that can learn long-term dependencies. LSTM cells by themselves are similar to neural networks which can have past iterations' information in the larger neural network. LSTM cells have an output edge that loops back as an input with a maximum weight value of one. This assures that these cells at every feed-forward iteration have the information of all the past steps. As mentioned earlier, the edge which loops back has a weight value of one, memories of older steps won't distinguish as steps increase as they would in classic Recurrent Neural Networks. Because of these attributes, LSTM models can be very useful to be working with time series data. By using some of the old stored states Recurrent Neural networks and LSTM can process the current information as well as the information which occurred one, ten, or a thousand steps ago.

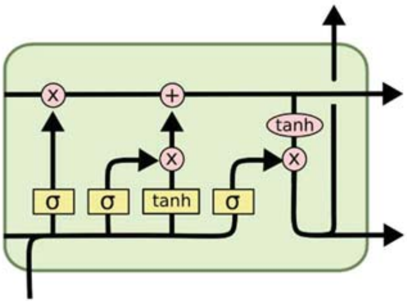


Fig 3. Descriptive image of a LSTM cell

The input layer will have shape (history_points, No. of highlights), since each information point is an exhibit formed as a structure [history_points × No. of features]. The architecture of the model will have a dropout layer to forestall overfitting and afterward a couple of dense layers to merge the entirety of the LSTM information. Figure 4. shows the LSTM model architecture.

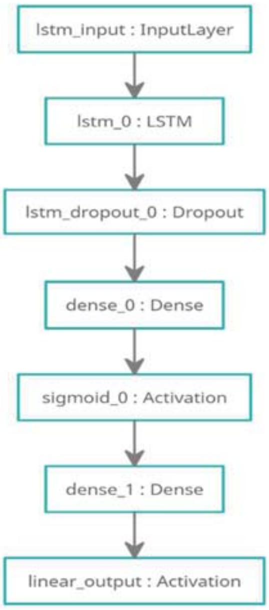


Fig 4. Structural graph of LSTM Model

V. Design and Data Flow

Figure 3. shows the workflow of the project, Data Crawler collects the required data then news headlines and market prices are merged and the sentiments (Positivity, negativity, Neutral, Subjectivity, Objectivity) are calculated. Basic pre-processing tasks like dimensionality reduction, data visualization, and normalization are also done in the pre-processing part before feeding the data to the prediction models. Processed data is then fed to the Classification(XGBoost) and Regression(LSTM) models which predict the flow of stock prices ("Raise" or "Fall") and stock prices respectively. After the prediction, the same data is fed to other similar and competitor models to do the comparisons of the results.

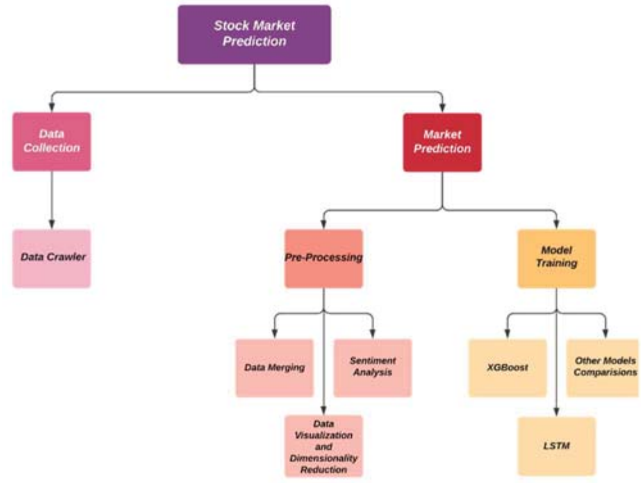


Fig 5. Flow diagram of workflow of the system

News data contains the Top 25 daily news headlines and stocks data contains daily stock prices data. News data is modified by extracting sentiments from headlines which are Positivity(Pos), Negativity(Neg), Neutral(Neu) and Subjectivity, and Objectivity Score(Sub_Obj). The modified news data is then merged with the daily stocks data to create input data for the prediction models. Figure 6 shows the flow of data for the system.

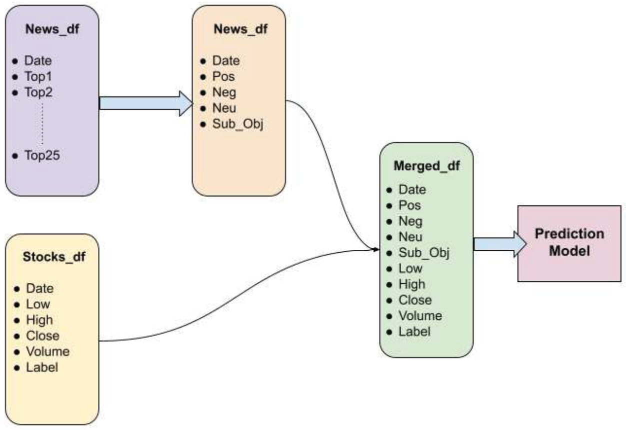
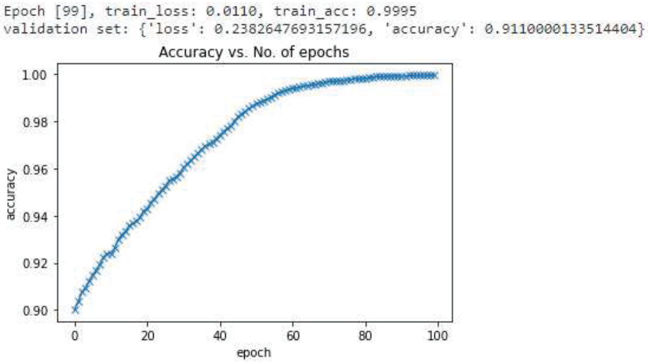


Fig 6. Explanation of data dictionaries and data flow

VI. Results

A. Sentiment Analysis

The Subjective/Objective classifier was trained on 10,000 sentences retrieved from the Cornell University data library with a 70:30 test/validation split. The trained model achieved 99% accuracy on training data and 91% accuracy on unseen validation data. Figure 7 shows the training graph of the Subjectivity/Objectivity classification model.



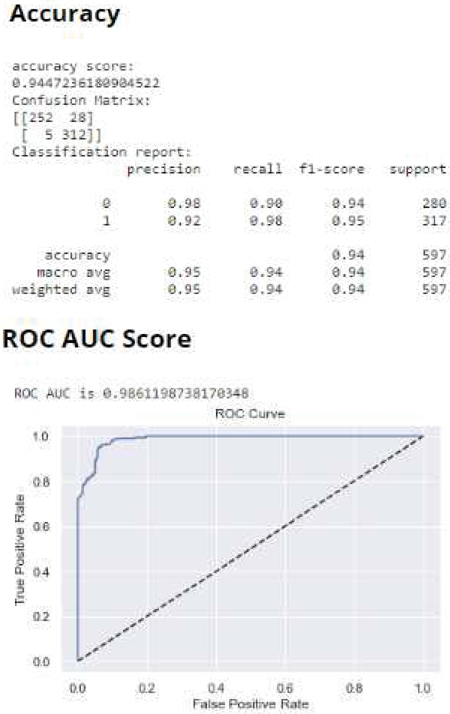
```
Epoch [99], train_loss: 0.0110, train_acc: 0.9995
validation set: {'loss': 0.2382647693157196, 'accuracy': 0.9110000133514404}
```

Fig 7. Training and Validation accuracy of Subjectivity/Objectivity Classifier

The accuracy of the model was not affected significantly and it stayed almost the same. But the number of required estimators in XGBoost reduced from 1000 to 450. This means the training time is reduced significantly as the model must train 55% fewer models. Subjectivity and Objectivity of the news headlines did not improve the accuracy by much but made the decision-making process easier for the model.

B. XGBoost

The XGBoost classification model achieved a 0.9447 accuracy score on test data and a 0.9861 ROC AUC score on 30% of test data.



Accuracy

```
accuracy score:
0.9447236180904522
Confusion Matrix:
[[252  28]
 [  5 312]]
Classification report:
              precision    recall  f1-score   support

           0       0.98      0.90      0.94       280
           1       0.92      0.98      0.95       317

    accuracy                           0.94       597
   macro avg       0.95      0.94      0.94       597
weighted avg       0.95      0.94      0.94       597
```

ROC AUC Score

```
ROC AUC is 0.9861198738170348
```

Fig 8. Test results in form of Accuracy Score (Top) and ROC AUC score (Bottom) of XGBoost classifier

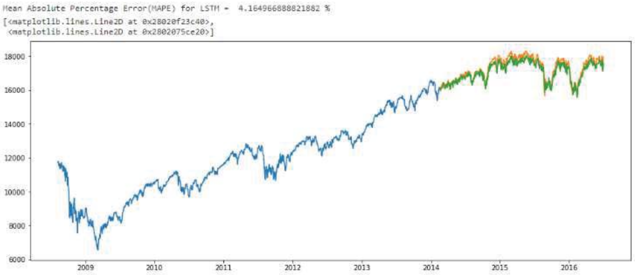
Some classic models were trained with the same data and their results were compared with XGBoost as well. Table 2 shows that most of the standard classification models performed poorly. Decision Tree Classifier and Random Forest Classifier performed comparatively very well with accuracy scores of 0.889 and 0.938 respectively. XGBoost outperformed every other classification model with an accuracy score of 0.945.

TABLE II. Comparison of accuracy scores of different classification models

Model	Accuracy Score
Logistic Regression	0.542
K-Nearest Neighbours	0.458
Decision Tree Classifier	0.889
Gaussian Naïve Bayes Classifier	0.536
Support Vector Machine	0.534
Random Forest Classifier	0.938
XGBoost	0.945

C. *LSTM*

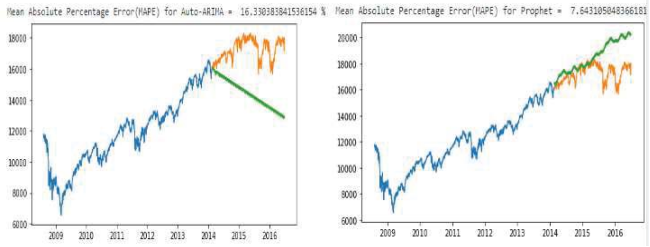
As seen in Fig 9. The LSTM regression model achieved a 4.164% Mean Absolute Percentage Error meaning that the LSTM model can predict the actual price with a 4.164% marginal difference.



```
Mean Absolute Percentage Error(MAPE) for LSTM =  4.164966888821882 %
[<matplotlib.lines.Line2D at 0x28020f23c40>,
 <matplotlib.lines.Line2D at 0x2802075ce20>]
```

Fig 9. Results of LSTM regression model including Mean Absolute Percentage Error (MAPE) and comparison graph of predicted (Green) and actual (Orange) values

Other time series prediction models Auto-Arima and FB Prophet were trained and tested on the same data. Auto Arima achieved MAPE 16.33% and predicted values and actual values were very separated in the comparison graph. FB Prophet achieved 7.64% MAPE and predicted values were closer to actual values in one direction but when the actual values changed direction the predicted values were very much inaccurate. Figure 10. Shows the results of Auto-Arima and FB Prophet time series regression models.



```
Mean Absolute Percentage Error(MAPE) for Auto-ARIMA =  16.330383841536154 %
Mean Absolute Percentage Error(MAPE) for Prophet =  7.643105048366181
```

Fig 10. Mean Absolute Percentage Error (MAPE) and Comparison graphs of predicted (Green) and actual (Orange) values of Auto-ARIMA(Left) and FB Prophet (Right)

VII. Summary

Stock market prices depend on various features. In this research project, we used one of the many factors, i.e., news sentiments. We extracted sentiments like positivity, negativity, neutral, subjectivity, and objectivity and used them as input to our prediction models along with the daily stock market data. For classification, XGBoost gave the most accurate results among the other standard algorithms and models. For regression, we used a sequential model of multiple LSTM cells and were able to predict the precise prices of the stocks and achieved better results than other well-known time series models and regression models. Subjectivity and Objectivity scores of the news headlines proved to be the most impacting feature for the prediction models followed by the negativity score of the headlines. As a future work, a fake news classifier can be incorporated at the front of the data pipeline to prevent any data fallacy from being fed to the algorithm. Along with that, other factors that affect the stock market can be studied and used to predict more accurate results.

References

[1] A. Nayak, M. M. M. Pai, and R. M. Pai, “Prediction Models for Indian Stock Market,” *Procedia Computer Science*, vol. 89, pp. 441–449, 2016, doi: 10.1016/j.procs.2016.06.096.

[2] L. Nemes and A. Kiss, “Prediction of stock values changes using sentiment analysis of stock news headlines,” *Journal of Information and Telecommunication*, vol. 5, no. 3, pp. 375–394, Feb. 2021, doi: 10.1080/24751839.2021.1874252.

[3] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, “Stock Closing Price Prediction using Machine Learning Techniques,” *Procedia Computer Science*, vol. 167, pp. 599–606, 2020, doi: 10.1016/j.procs.2020.03.326.

[4] V. Martin, “Predicting the French Stock Market Using Social Media Analysis,” *2013 8th International Workshop on Semantic and Social Media Adaptation and Personalization*, Dec. 2013, **Published**, doi: 10.1109/smap.2013.22.

[5] X. Zhang, S. Qu, J. Huang, B. Fang, and P. Yu, “Stock Market Prediction via Multi-Source Multiple Instance Learning,” *IEEE Access*, vol. 6, pp. 50720–50728, 2018, doi: 10.1109/access.2018.2869735.

[6] M. Shastri, S. Roy, and M. Mittal, “Stock Price Prediction using Artificial Neural Model: An Application of Big Data,” *ICST Transactions on Scalable Information Systems*, vol. 0, no. 0, p. 156085, Jul. 2018, doi: 10.4108/eai.19-12-2018.156085.

[7] 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. 08, no. 04, pp. 309–319, 2020, doi: 10.4236/jdaip.2020.84018.

[8] A. E. Khedr, S. Salama, and N. Yaseen, “Predicting Stock Market Behavior using Data Mining Technique and News Sentiment Analysis,” *International Journal of Intelligent Systems and Applications*, vol. 9, no. 7, pp. 22–30, Jul. 2017, doi: 10.5815/ijisa.2017.07.03.

[9] Jiahong Li, Hui Bu and Junjie Wu, "Sentiment-aware stock market prediction: A deep learning method," 2017 International Conference on Service Systems and Service Management, 2017, pp. 1-6, doi: 10.1109/ICSSSM.2017.7996306.

[10] Sun, J. (2016, August). Daily News for Stock Market Prediction, Version 1. Retrieved 09/03/2021 from https://www.kaggle.com/aaron7sun/stocknews.

[11] Y. Xie, “Stock Market Forecasting Based on Text Mining Technology: A Support Vector Machine Method,” *Journal of Computers*, pp. 500–510, 2017, doi: 10.17706/jcp.12.6.500-510.

[12] Bo Pang, Lilian Lee (2004, June). Subjectivity Dataset, Version 1. Retrieved from http://www.cs.cornell.edu/people/pabo/movie-review-data/rotten_imdb.tar.gz

[13] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. Retrieved from http://nlp.stanford.edu/data/glove.6B.zip

[14] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine.." Ann. Statist. 29 (5) 1189 - 1232, October 2001.