# Integration of Deep Learning Techniques for Effective Sentiment Analysis on Stock Market News

**Sakshi Borah [1] and Shallu Bashambu [2]**

Maharaja Agrasen Institute of Technology, Guru Gobind Singh Indraprastha University, New Delhi, India

**Abstract:** Stock market forecasting is very important in the planning of business activities. Stock price prediction has attracted many researchers in multiple disciplines including computer science, statistics, economics, finance, and operations research. Recent studies have shown that the vast amount of online information in the public domain such as Wikipedia usage pattern, news stories from the mainstream media, and social media discussions can have an observable effect on investors' opinions towards financial markets. The reliability of the computational models on stock market prediction is important as it is very sensitive to the economy and can directly lead to financial loss. In this paper, we retrieved, extracted and analyzed the effects of news sentiments on the stock market. Our main contributions include the development of a dictionary-based sentiment analysis model and the evaluation of the model for gauging the effects of news sentiments on the stocks. Using only news sentiments, we achieved a directional accuracy of ~97% in predicting the trends in short-term stock price movement.

*KEYWORDS*—Text Mining, Sentiment analysis, Random Forest, SVM, Stock trends

DOI of the Article: https://doi.org/10.46501/IJMTST0706048

Available online at: http://www.ijmtst.com/vol7issue06.html

As per UGC guidelines an electronic bar code is provided to seure your paper

**To Cite this Article:** Sakshi Borah and Shallu Bashambu. Integration of Deep Learning Techniques for Effective Sentiment Analysis on Stock Market News. *International Journal for Modern Trends in Science and Technology* 2021, *7*, 0706207, pp. 285-290. https://doi.org/10.46501/IJMTST0706048

**Article Info.** Received: 15 May 2021; Accepted: 14 June 2021; Published: 21 June 2021

## INTRODUCTION

In the finance field, the stock market and its trends are very unpredictable in nature. It pulls in specialists to catch the unpredictability and anticipate its best courses of action. Financial specialists and market investigators study the market conduct and plan their purchase or sell methodologies in like manner. As securities exchange creates a huge measure of information consistently, it is hard for a person to think about all the current and past data for foreseeing future patterns of a stock. Chiefly there are two techniques for determining market patterns. One is Technical Analysis and other is Fundamental Analysis. Specialized analysis considers past cost and volume to foresee the future pattern. On the other hand, Fundamental Analysis of a business includes dissecting its monetary information to get a few bits of knowledge. The viability of both specialized and key examination is questioned by the effective market speculation which expresses that financial exchange costs are basically erratic.

This research follows the Fundamental Analysis procedure to find future patterns of a stock by considering news stories about an organization as prime data and attempts to arrange news as good (positive) and bad (negative). In the event that the news sentiment is positive, there are more possibilities that the stock price will go up and in the event that the news sentiment is negative, at that point stock price may go down.

This research is an endeavor to construct a neural network that predicts news polarity which may influence changes in stock trends. In other words, check the effect of news stories on stock costs. We are utilizing Deep Learning and other text mining strategies to check news polarity. And furthermore, have the option to group unknown news, which isn't utilized to construct a classifier. Three distinctive classification algorithms are implemented to check and improve classification accuracy.

Twitter, with more than 500 million users and abundance of 400 million directives for each day, has changed into a goldmine for the relationship to screen their reputation and brands by eliminating and investigating the tendency of the tweets posted by everyone about them, their business divisions, and competitors. Making exact evaluation assessment techniques requires the course of action of examination datasets that can be utilized to survey their presentations.

Throughout the most recent few years, two or three assessment datasets for Twitter doubt assessment have been made wholeheartedly accessible.

## LITERATURE SURVEY

a. Stock price trend forecast is an active research zone, as more precise expectations are straightforwardly identified with more returns in stocks. Along these lines, as of late, huge endeavors have been placed into creating models that can foresee the future pattern of a particular stock or in general market. The vast majority of the current procedures utilize the technical indicators. A portion of the researchers demonstrated that there is a solid connection between news stories about an organization and its stock price changes. Following is discussion on past research on sentiment analysis of text data and different classification techniques.

b. Nagar and Hahsler in their analysis [1] introduced an automated text mining based approach to deal with total reports from different sources and make a News Corpus. The Corpus is separated down to pertinent sentences and dissected utilizing Natural Language Processing (NLP) procedures. A sentiment metric, called News Sentiment, using the count of positive and negative polarity words is proposed as a measure of the sentiment of the general news corpus. They have utilized different open source bundles and tools to build up the news collection and accumulation engine just as the sentiment evaluation engine. They additionally claim that the time variation of News Sentiment shows a solid connection with the real stock value movement.

c. Yu et al [2] present a text mining based structure to decide the sentiment of news articles and show its effect on energy demand. News estimation is evaluated and afterward introduced as a period arrangement and contrasted and changes in energy interest and costs.

d. J. Bean [3] utilizes keyword tagging on Twitter channels about airlines fulfillment to score them for polarity and sentiment. This can give a brisk thought of the sentiment prevailing about airlines and their consumer satisfaction ratings. We have utilized the sentiment detection algorithm based on this research.

e. This examination paper [4] concentrates how the consequences of monetary forecasting can be improved when news articles with various degrees of pertinence to the target stock are utilized simultaneously. They utilized multiple kernels learning techniques for partitioning the information which is extracted from different five categories of news articles based on sectors, sub-sectors, industries etc.

f. News articles are partitioned into the five classes of significance to a targeted stock, its sub industry, industry, group industry and sector while separate kernels are employed to analyze each one. The trial results show that the synchronous use of five news classifications improves the prediction performance in correlation with strategies dependent on a lower number of news categories. The discoveries have demonstrated that the highest prediction accuracy and return per trade were accomplished for MKL when every one of the five classifications of news were used with two separate kernels of the polynomial and Gaussian types used for each news category.

## Dataset

For our problem statement, we have selected a dataset from Kaggle which includes major key events news articles of the Dow Jones Industrial Average (DJIA), hinting at a financial bias for the last couple of years. Our job now is to assign individual sentiments to all the data entries.

## Methodology

### A. System Design

Following system design is proposed in this project to classify news articles for generating stock trend signals.

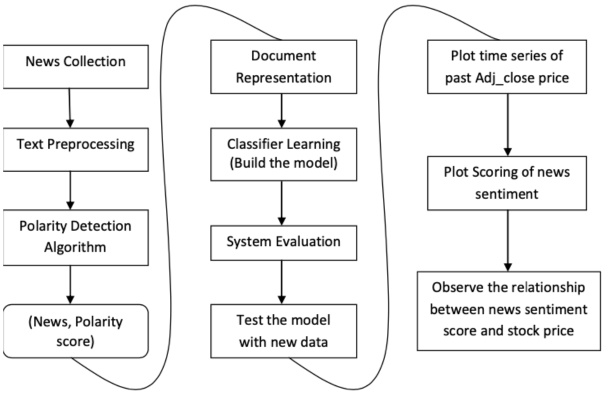

News Collection
Text Preprocessing
Polarity Detection Algorithm
(News, Polarity score)
Document Representation
Classifier Learning (Build the model)
System Evaluation
Test the model with new data
Plot time series of past Adj_close price
Plot Scoring of news sentiment
Observe the relationship between news sentiment score and stock price

Figure 1: System Design

This design can logically be seen as three phases with the first column of blocks in Phase 1, second column as Phase 2 and third column contains blocks in Phase 3. Result of Phase 1 is news articles with a polarity score. This result is given as an input to the Phase 2. In Phase 2, text is converted in TF-IDF vector space so that it can be given to the classifier. Then three different classifiers are programmed for the same data to compare results. At the end of Phase 2, we evaluate the results given by all classifiers and also test for checking classifier performance for new news articles. In Phase 3, we check for relationships between news articles and stock price data. We plot both the data using Python and record the results. In the following sections, each block of the design is explained [5].

### B. Data Pre-processing

Text data is unstructured data. so, we cannot give raw test data to the classifier as an input. Initially, we have to tokenize the report into words to work on word level. Text data contains more boisterous words which are not contributing towards classification. In this way, we have to drop those words. Likewise, text data may contain numbers, more blank areas, tabs, punctuation characters, stopwords and so on. We additionally need to clean the data by eliminating each one of those words [6].

Likewise, in order to ignore words that show up in just a couple of reports, we are considering minimum document frequency which considers words that show up in at least three records. Stemming is likewise imperative to decrease repetition in words. Utilizing the stemming process, all the words are supplanted by its unique adaptation of words. For instance, the words 'created', 'creation', 'creating' are diminished to its stem word 'create'. A portion of the pre-processing is done prior to applying a polarity detection algorithm. Also, some of them are applied after applying polarity detection algorithms.

### *C. Sentiment Assignment Algorithm*

For this we will be using the VADER Sentiment Analysis. VADER (Valence Aware Dictionary for Sentiment Reasoning) is a pre-built sentiment analysis model included in the NLTKpackage [7]. It can give both positive/negative (polarity) as well as the strength of the emotion (intensity) of a text. It is rule-based and relies heavily on humans rating texts via Amazon Mechanical Turk — a crowd-sourcing e-platform which utilizes human intelligence to perform tasks that computers are currently unable to do. These are words or any textual form of communication generally labelled according to their semantic orientation, as either positive or negative, for us. Sentiment Intensity Analyzeris an object included in the larger NLTK library under VADER [8]. The algorithm to calculate the sentiment score of a document is given below:

1. Tokenize the document into word vectors.
2. Prepare the dictionary which contains words with its polarity (positive or negative)
3. Check against each word whether it matches with one of the words from a positive word dictionary or negative words dictionary.
4. Count number of words belongs to positive and negative polarity.
5. Calculate Score of document = count (pos.matches) – count (neg.matches)
6. If the Score is 0 or more, we consider the document is positive or else, negative.

Here, all the data entries have assigned their respective sentiments. The compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1(most extreme positive). In other words,

- positive sentiment: compound score > 0.33
- neutral sentiment: -0.33 < compound score < 0.33
- negative sentiment: compound score < -0.33

### *D. Document Presentation*

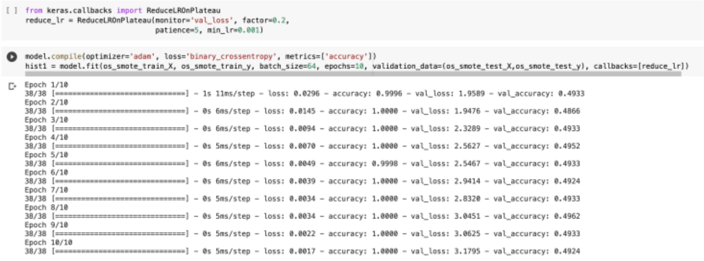
To diminish the intricacy of text archives and make them simpler to work with, the reports must be changed from the full test version to a document vector which describes the contents of the document. To represent text documents, we are utilizing the TF-IDF scheme [9]. The higher TF-IDF value a term gets, the more significant it is. A high value is reached when the term frequency in the given document is high and when there are not many different documents in the collection containing the given term/feature. This term-weighing strategy tends, thus, to sift through normal terms by giving them an extremely low worth.

### *E. System Evaluation*

We separated the data into train and test sets. Likewise, we made an obscure dataset for classifiers to check the precision of the classifier against new data. We will assess each one of the classifiers' performance by checking their accuracy, precision, recall and ROC curve area.

### *F. Neural Network*

- We trained a Neural Network to predict how accurate the data labelling was, but encountered some overfitting. This implies that our data is imbalanced.



```
[ ] from keras.callbacks import ReduceLROnPlateau
    reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2,
                                  patience=5, min_lr=0.001)
```

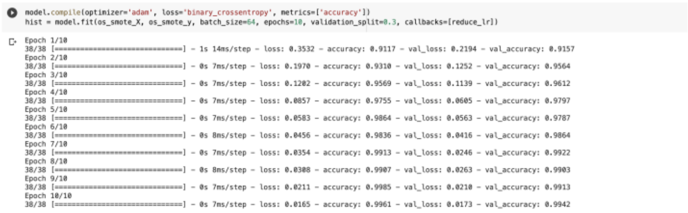
```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
hist1 = model.fit(os_smote_train_X, os_smote_train_y, batch_size=64, epochs=10, validation_data=(os_smote_test_X,os_smote_test_y), callbacks=[reduce_lr])
```

```
Epoch 1/10
38/38 [==============================] - 1s 11ms/step - loss: 0.0296 - accuracy: 0.9996 - val_loss: 1.9589 - val_accuracy: 0.4933
Epoch 2/10
38/38 [==============================] - 0s 6ms/step - loss: 0.0145 - accuracy: 1.0000 - val_loss: 1.9476 - val_accuracy: 0.4866
Epoch 3/10
38/38 [==============================] - 0s 6ms/step - loss: 0.0094 - accuracy: 1.0000 - val_loss: 2.3289 - val_accuracy: 0.4933
Epoch 4/10
38/38 [==============================] - 0s 5ms/step - loss: 0.0070 - accuracy: 1.0000 - val_loss: 2.5627 - val_accuracy: 0.4952
Epoch 5/10
38/38 [==============================] - 0s 6ms/step - loss: 0.0049 - accuracy: 0.9998 - val_loss: 2.5467 - val_accuracy: 0.4933
Epoch 6/10
38/38 [==============================] - 0s 6ms/step - loss: 0.0039 - accuracy: 1.0000 - val_loss: 2.9414 - val_accuracy: 0.4924
Epoch 7/10
38/38 [==============================] - 0s 5ms/step - loss: 0.0034 - accuracy: 1.0000 - val_loss: 2.8320 - val_accuracy: 0.4933
Epoch 8/10
38/38 [==============================] - 0s 5ms/step - loss: 0.0034 - accuracy: 1.0000 - val_loss: 3.0451 - val_accuracy: 0.4962
Epoch 9/10
38/38 [==============================] - 0s 5ms/step - loss: 0.0022 - accuracy: 1.0000 - val_loss: 3.0625 - val_accuracy: 0.4933
Epoch 10/10
38/38 [==============================] - 0s 5ms/step - loss: 0.0017 - accuracy: 1.0000 - val_loss: 3.1795 - val_accuracy: 0.4924
```

Figure 2: Neural Network

- Imbalanced data can be made balanced by using SMOTE [10]. Synthetic Minority Over-Sampling Technique (SMOTE)is performing the same basic task as basic resampling (creating new data points for the minority class) but instead of simply duplicating observations, it creates new observations along the lines of a randomly chosen point and its nearest neighbours. Basically, we are simulating some additional variation in the data (within the established bounds of your minority class), reducing the danger of overfitting (although not eliminating it).

- After applying SMOTE, Logistic Regression is applied again which now shows the accuracies of both 1 and 0 with a decreased accuracy of 1.



```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
hist = model.fit(os_smote_X, os_smote_y, batch_size=64, epochs=10, validation_split=0.3, callbacks=[reduce_lr])
```

```
Epoch 1/10
38/38 [==============================] - 1s 14ms/step - loss: 0.3532 - accuracy: 0.9117 - val_loss: 0.2194 - val_accuracy: 0.9157
Epoch 2/10
38/38 [==============================] - 0s 7ms/step - loss: 0.1970 - accuracy: 0.9310 - val_loss: 0.1252 - val_accuracy: 0.9564
Epoch 3/10
38/38 [==============================] - 0s 7ms/step - loss: 0.1202 - accuracy: 0.9569 - val_loss: 0.1139 - val_accuracy: 0.9612
Epoch 4/10
38/38 [==============================] - 0s 7ms/step - loss: 0.0857 - accuracy: 0.9755 - val_loss: 0.0605 - val_accuracy: 0.9797
Epoch 5/10
38/38 [==============================] - 0s 7ms/step - loss: 0.0583 - accuracy: 0.9864 - val_loss: 0.0563 - val_accuracy: 0.9787
Epoch 6/10
38/38 [==============================] - 0s 8ms/step - loss: 0.0456 - accuracy: 0.9836 - val_loss: 0.0416 - val_accuracy: 0.9864
Epoch 7/10
38/38 [==============================] - 0s 7ms/step - loss: 0.0354 - accuracy: 0.9913 - val_loss: 0.0246 - val_accuracy: 0.9922
Epoch 8/10
38/38 [==============================] - 0s 8ms/step - loss: 0.0308 - accuracy: 0.9907 - val_loss: 0.0263 - val_accuracy: 0.9903
Epoch 9/10
38/38 [==============================] - 0s 7ms/step - loss: 0.0211 - accuracy: 0.9985 - val_loss: 0.0210 - val_accuracy: 0.9913
Epoch 10/10
38/38 [==============================] - 0s 7ms/step - loss: 0.0165 - accuracy: 0.9961 - val_loss: 0.0173 - val_accuracy: 0.9942
```

Figure 3: Application of SMOTE

- Feature Selectionis the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. It reduces overfitting, reduces training time and improves accuracy. Here, we will be using Extra Tree Classifier for extracting the top 10 features for the dataset [11].
- For further verification, XGBoost is applied which shows an accuracy of around 0.73.
- Ensemble model is also applied to calculate and record which statistical models have good F1 scores.

*G. Evaluation*

We tested the algorithms so that we could compare each method against their mean accuracies and observed the following Cross-validation scores [12]:

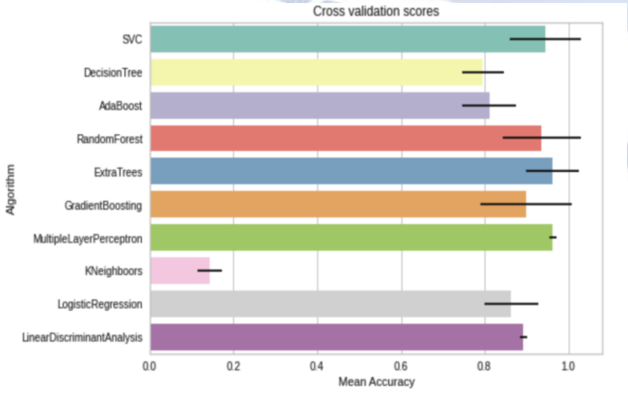


Figure 4: Cross-validation scores

Using hyper-parameter tuning, we conclude the following results:

| STATISTICAL MODEL | ACCURACY |
|---|---|
| XGBoost | 0.73 |
| ADABoost on Decision Tree Classifier | 0.80 |
| Extra Tree Classifier | 0.97 |
| Neural Network | 0.99 |

Figure 5: Result of testing models with polarized data

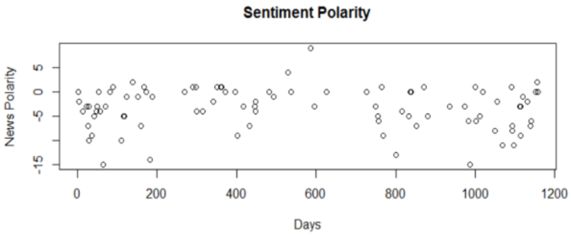


Figure 6: Time-series plot of news-sentiment score

## Conclusion

Discovering future stock trends is a critical task since stock trends rely upon a number of variables. We assumed that news articles and stock prices are related with one another. And also, news may have the ability to fluctuate stock trends. Thus, we thoroughly studied this relationship and inferred that stock trends can be anticipated utilizing news articles and past price history.

As news articles catch sentiment about the current market, we automate this sentiment detection and based on the words in the news articles, we can get a general news polarity. When we come to know that the news is positive, at that point we can express that this news has a good impact on the market, so there are higher odds of stock price increasing. Also, in the event that the news is negative, at that point it might cause the stock price to go down in trend. We utilized a polarity detection algorithm for initially labelling news and making the train set. For this algorithm, a word-reference based approach was utilized. The word references for positive and negative words are made utilizing general and finance specific sentiment carrying words. At that point preprocessing of text data was likewise a difficult task. Based on this data, we executed several classification models and tested under various test scenarios. After comparing their results, we found that Random Forest and Extra Tee Classifiers functioned very well for all experiments going from 94% to 97% accuracy. Accuracy followed by ADABoost on Decision Tree Classifier is likewise significant around 80%. Given any news article, it would be possible for the model to arrive at a polarity which would further predict the stock trend.

## FUTURE WORK

We would like to extend this research by adding data on several more companies and check the prediction accuracy. For those companies where availability of financial news is a challenge, we would be using twitter data for similar analysis. We can also incorporate similar strategies for algorithmic trading.

## REFERENCES

1. Anurag Nagar, Michael Hahsler, Using Text and Data Mining Techniques to extract Stock Market Sentiment from Live News Streams, IPCSIT vol. XX (2012) IACSIT Press, Singapore
2. W.B. Yu, B.R. Lea, and B. Guruswamy, A Theoretic Framework Integrating Text Mining and Energy Demand Forecasting, International Journal of Electronic Business Management. 2011, 5(3): 211-224
3. J. Bean, R by example: Mining Twitter for consumer attitudes towards airlines, In Boston Predictive Analytics Meetup Presentation, 2011
4. YauheniyaShynkevich, T.M. McGinnity, Sonya Coleman, Ammar Belatreche, Predicting Stock Price Movements Based on Different Categories of News Articles, 2015 IEEE Symposium Series on Computational Intelligence
5. Kyoung-jae Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (2013) 307 – 319
6. P. Hofmarcher, S. Theussl, and K. Hornik, Do Media Sentiments Reflect Economic Indices? Chinese Business Review. 2011, 10(7): 487-492
7. R. Goonatilake and S. Herath, The volatility of the stock market and news, International Research Journal of Finance and Economics, 2007, 11: 53-65.
8. Spandan Ghose Chowdhury, Soham Routh, Satyajit Chakrabarti, News Analytics and Sentiment Analysis to Predict Stock Price Trends, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 3595-3604
9. Robert P. Schumaker, Yulei Zhang, Chun-Neng Huang, Sentiment Analysis of Financial News Articles
10. Győző Gidófalvi, Using News Articles to Predict Stock Price Movements, University of California, San Diego La Jolla, CA 92037, 2001
11. L. Breiman, Random forests. Machine Learning, 45(1):5-32, 2001
12. Data Mining Lab 7: Introduction to Cross-Validation Scores (CVS)
13. Kyoung-jae Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (2013) 307 – 319