A Big Data Approach in Sentiment Analysis of Tweets

Kai Jun Tey
School of Computer Science
University of Nottingham
Nottingham, United Kingdom
hfykt6@nottingham.ac.uk

Yih Seng Liew
School of Computer Science
University of Nottingham
Nottingham, United Kingdom
hfyyl2@nottingham.ac.uk

Yen Kai Lim
School of Computer Science
University of Nottingham
Nottingham, United Kingdom
hfyyl3@nottingham.ac.uk

Wen Jye Chai
School of Computer Science
University of Nottingham
Nottingham, United Kingdom
hfywc5@nottingham.ac.uk

***Abstract*—In recent years, Twitter has emerged as one of the prominent sources for big data extraction in the context of text-based sentiment analysis. Within this paper, two distinct approaches, namely the sequential and distributed computing frameworks have been demonstrated onto a dataset concerning tweets related to the 2020 United States Presidential Election. The motive of this research is to illustrate the common predicament experienced in big data problems, where time and memory consumption are the primary limiting factors in developing an effective big data solution. To depict both computing frameworks, this paper utilises pre-built libraries such as Apache Spark and scikit-learn. To further advocate the research motive, this paper proposed various machine learning techniques such as Random Forest and Naïve Bayes classifiers to study the sentiments present within the tweets. The predictions made by the proposed machine learning models will be closely compared to the actual election result, providing a better insight to the correlation that exists between the candidacy's support rate based on tweets and the actual support rate. As a result of this research, it is proven that the distributed computing framework outperformed the sequential approach in handling big data problems because it provides a greater efficiency when compared to the sequential approach.**

***Index Terms*—Sentiment Analysis, US Election 2020, Big Data**

I. Introduction

Nowadays, social media platforms have become one of the most prominent sources for information extraction in the field of big data. Platforms like Twitter, a microblogging platform, has recently gained its popularity by promoting users to journalise their day-to-day activities in the form of "tweets". These tweets often encompass users' cognition towards a certain occurrence or event either in the past or the present. To ensure tweets are within the defined character limit of 280 characters, users generally include hashtags followed by words or multi-word phrases that implies a specific trending topic. According to Statista, the number of Twitter users worldwide has been steadily increasing from a total of 290.5 million of users in the year 2019 to 329 million users in the year 2022 [1]. Such growth in the number of Twitter users worldwide will indirectly cause an increment to the usages of social media platforms, leads to a surge in volume of structured and unstructured data being established on Twitter [2].Trending topics on Twitter are ranked based on the popularity of keyword combinations that exist within a tweet. Oftentimes, data analysts make use of this big data present within Twitter to assess how public opinion is evolving in trending topics. For instance, the usage of #USElection2020 that represents tweets regarding the 2020 United States (US) Presidential Election had been exploited to understand the public's attitudes towards the US presidential candidates and their respective political parties.

Considered as one of the most significant political events in history, the 2020 US Presidential Election where Joe Biden (JB) of the Democrats attempted to overturn Donald Trump (DT) of the Republicans for his second term of presidency, has become one of the most popular Twitter trending topics throughout the year 2020. The focus of this paper is to implement a series of appropriate classification machine learning (ML) techniques that includes Random Forest (RF) and Naïve Bayes classifiers accompanied by big data solutions onto an unlabeled dataset.

Nevertheless, there exist several challenges when solving problems in this setting. Firstly, the big data presented within this paper will result in an extended training duration of an ML technique. Hence, appropriate big data methodologies will be proposed to ensure efficiency during the training of classification models. Such methodologies will ameliorate the process of conducting analysis on tweets related to the 2020 US Presidential Election. Considering that this dataset is unlabeled, conducting sentiment analysis on the tweets within the big data is laborious. Therefore, this paper will be focusing on utilising big data techniques and natural language processing packages such as the Natural Language Toolkit (NLTK) library in Python, in hopes to produce a classification model that is efficient and effective.

With regards to the literature reviews, it has been suggested that general research in the application of distributed computing framework to conduct sentiment analysis on big data containing tweets regarding the US Presidential Election is still in its infancy. Previous implementations of sentiment analysis using big data approaches are mainly targeted on different areas of interest. Nodarakis et al. (2016) suggested the application of a novel approach for sentiment learning based on a distributed computing methodology using Apache Spark framework on hashtags and emoticons present within tweets [3]. In Elzayady et al. (2018), the authors utilised the

distributed memory abstraction capability offered by Apache Spark to effectively train classification algorithms such as Naïve Bayes, Logistic Regression and Decision Trees to analyse the sentiments of each tweet [4]. Karhan et al. proposed a distributed architecture for conducting sentiment analysis on Twitter data that benefits government agencies to gather public opinion before making decisions [5]. In Sharma et al. (2016), the authors proposed a sequential computing framework for making predictions on India's General State Election in 2016 using sentiment analysis based on tweets in Hindi [6]. Rane & Kumar (2018) suggested a distributed approach to effectively conduct a multiclass sentiment analysis on tweets regarding the six major US Airlines [7].

A. Aims and Objectives

The aim of this study is to investigate the significance of practising big data approaches in sentiment analysis of tweets regarding the 2020 United States Presidential Election.

The key objectives of the study are:

1) To implement a series of sentiment analysis classifiers utilising various machine learning techniques that can perform effectively on big data relating to tweets from the 2020 United States Presidential Election.
2) To compare performances of sequential and distributed computing frameworks when engaging with big data.
3) To study the correlation between the proposed sentiment analysis made on tweets and the actual results of the 2020 United States Presidential Election.

II. Proposed Methodology

A. Lexicon-based Sentiment Analysis

In general, lexicon-based sentiment analysis approach initiate with data cleaning. Invalid data entries are removed to preserve data consistency. Following that, data pre-processing steps were carried out. The tweets that contain both nominees are removed to ensure the subject of the tweet when computing the sentiment polarity score. Besides, only tweets that are written in English are selected as the analysis function does not support multilingual sentiment analysis. Hyperlink, mention, and new line symbols are also removed as research supports that these do not contribute to sentiment analysis. [8] Upon pre-processing, the sentiment of the tweets is computed using the NLTK Valence Aware Dictionary for Sentiment Reasoning (VADER) function. The sentiment score calculation relies on mapping lexical features from tweet to sentiment scores. Finally, the sentiment score of tweets is used to perform polarity classification with the label '-1' as negative opinion; '1' as positive opinion towards the nominee; '0' as neutral opinion. The flowchart of the methodology is portrayed in Figure 1. This methodology was implemented using sequential and distributed computing approaches. The implementation details are scrutinised in the following section.

1) Sequential computing approach: As suggested by the name, all the code was executed sequentially using a single processor without overlapping. To elucidate this point, in this approach, the pre-processing steps were carried out row by row following the indexing order of the dataset. Ergo, the implementation of lexicon-based sentiment analysis using a sequential computing approach is intuitive.

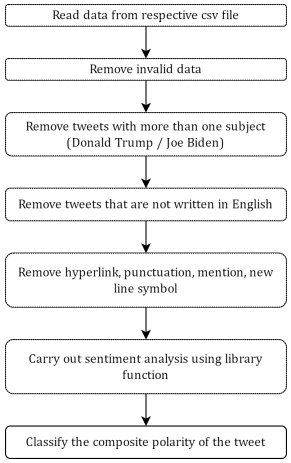

Fig. 1. The methodology flowchart of lexicon-based sentiment analysis

2) Distributed computing approach: Unlike the aforementioned approach, all the code was distributed and executed sequentially using clusters of computing nodes. To elucidate, the dataset is partitioned and distributed into multiple computing nodes and the pre-processing steps are carry out in respective nodes.To maximise the efficiency of distributed computing, the source code of the sentiment analysis function had been revised and adapted to the approach. For example, the in-built pre-processing codes from the sentiment analysis function were extracted and modified to be executed on computing nodes.

B. Machine Learning Based Sentiment Analysis

The training of multi-class classifiers using RF and Naïve Bayes (NB) to predict the sentiment of tweets is done using 2 different approaches, which are sequential and distributed computing approach which be highlighted in the following section.

1) Sequential Computing Approach: For the sequential approach, the dataset for both JB and DT tweets are loaded in parquet format and converted into Pandas dataframe. The label for "Sentiment_Overall" works the same as mentioned in the Lexicon-based Sentiment Analysis section above. To classify pro-JB, anti-JB, pro-DT, anti-DT and neutral tweets, the labels are set to 1,2,3,4 and 0 respectively. Next, functions in CleanText() are executed on the tweets of the merged dataset to remove symbols that will not help with sentiment analysis such as "@, urls, punctuation and emojis". After that, the processed tweets are vectorised using an import from scikit-learn. Vectorising of words is the process of converting words into numbers by mapping words or phrase from vocabulary to a corresponding vector of numbers. Since the "Sentiment_Overall" feature is in the form of an object, it has to be converted into an integer. The dataset is then split

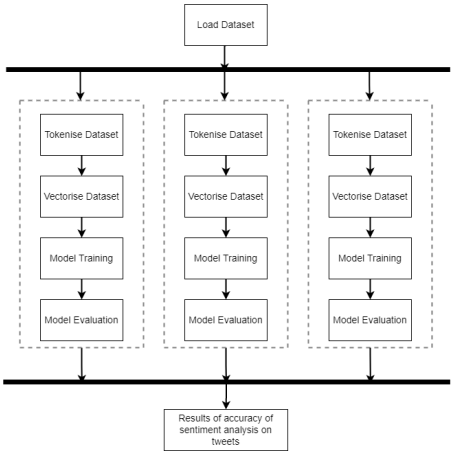

Fig. 2. ML-based Sentiment Analysis under distributed computing framework

into 80% training set and 20% test set. It helps in making an evaluation on the performance of the model by testing trained model with unseen data. The training set is then fit to a RF Classifier imported from the scikit-learn library. RF classifier works by having a large amount of individual decision trees which will all make a class prediction and the class with the most votes will be come the model's final prediction. The RF model is trained with the amount of trees set to 200 and it is used to make predictions on the test set. By comparing the results of the prediction and the expected results, the accuracy of the model is calculated. The same training data is used to train the NB model imported from the same library. This classifier uses Bayes Theorem to make prediction on the probability of a given data point belongs to a particular class.The class with the highest probability is considered the most likely class.

2) Distributed Computing Approach: For distributed computing approach, the same parquet dataset for both Biden and Trump are loaded. The loaded dataset is also split into 32 partitions so that the processing of dataset can be done in distributing computing nodes. Using a User Defined Function (UDF), the same data pre-processing technique in the sequential computing approach is done. However, since Pyspark CountVectorizer requires its input data to first be tokenised which is not required in the scikit-learn CountVectorizer. Tokenisation is the process of splitting a sentence into individual words. This process is carried out on tweets before vectorising. Then the training data is split into 80% training and 20% test data. The training data is fit to a RF Classifier imported from Pyspark with the same number of trees set in the sequential computing approach. Finally, using MulticlassClassificationEvaluator from Pyspark, the accuracy of the model is evaluated. The same training data is fit into NB algorithm from the Pyspark library and evaluated.

C. Identifying Tweet Support of Both Candidates

An analysis of data was done to compare the actual US election result in the year of 2020 based on the "Sentiment_Overall" result labelled on the tweets. Sentiment analysis is a score that represent the sentiment polarity of a tweets. In this work, the lower the scale indicating negative response and vice versa. The positive sentiment which is labelled 1 on tweets in the #Trump will be used as voters that will vote DT on the actual election day and positive sentiment that is labelled 1 on tweets in the #Biden will be used as voters that will vote JB on the actual election day. It is because negative sentiment tweets on either side do not indicate they support the other side. There are chances that they do not support either side or vote either side during the election day. However, positive sentiment on one of the candidates shows that the Twitter user supports that candidate and there is a high chance that he or she will vote for the particular candidate during election. Then, a bar chart was plotted to show the amount of tweet support of both candidates in all US states. The ratio of both tweet support of both candidates in all US states is calculated by dividing DT supporters with JB supporters to determine which candidate won that state. If the ratio is bigger than one, DT wins the state otherwise JB wins the state.

D. Winning Criteria of Election

Besides, the actual presidential result, number of electoral seats of each state and population who voted JB and DT during the US election 2020 are collected from CNN Politic since US election winning rule is determined by whoever won the most electoral seats from each state.. [9] Every state in the US has their number of electoral seats and the number of electoral seats won determines the winner of the election.

E. Analysis of Data

Three types of analysis of the data was done which is comparing the actual election results with the Twitter's sentiment analysis result by assuming that positive sentiments in tweets from #Trump is voting for DT and #Biden is voting for JB, comparing the actual election results with unique users in Twitter that tweeted in #Trump or #Biden by filtering out duplicate tweets by the same users in both hashtags. After that, the correlation between both election results using tweet's sentiment analysis and actual election results using votes are calculated by using Pearson Correlation Coefficient(Eq. 2) of the two's JB supports and DT supports ratio to check whether it is appropriate to use this statistical test.

III. Experimental Set-up

A. Performance Metrics

1) Accuracy: Performance of the trained classifier will be evaluated based on the percentage of accurate predictions. The calculation of accuracy can be derived by dividing the number of correct predictions by the total number of predictions made. From Eq. 1, it shows the equation for calculating accuracy of the classification model.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (1)$$

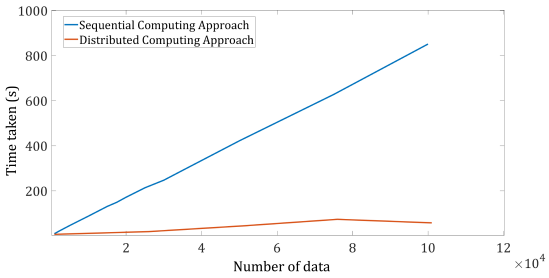

Fig. 3. Time complexity of lexicon-based Sentiment Analysis under different approach

2) Pearson Correlation Coefficient: Pearson Correlation Coefficient also known as the Pearson Product-Moment Correlation Coefficient is an experiment used to measure the strength of a linear association between two variables which is the ratio of tweet support between DT and JB and the ratio of actual election support between DT and JB. [10] The Pearson Correlation Coefficient will attempt to draw a line best fit through the data of two variables and the Pearson Correlation Coefficient value indicates how these data points best fit to the line. The Pearson Correlation Coefficient takes a value from a range of +1 to -1. A 0 value indicates that there is no association between the two variables. A value smaller than 0 indicates that it has negative association and a value greater than 0 indicates that it has positive association.2

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{2}$$

B. Dataset

The dataset used in the proposed work originates from Kaggle by Hui [11]. It comprised of all 2020 US Presidential Election-related tweets made between the date ranging from 15th October to 8th November 2020. All data entries within the dataset have been tagged with their corresponding metadata such as user's location at times when their tweet has been established. These metadata will be used to analyse the overall behaviour of Twitter users in relation to the political positions of both presidential candidates across all states in the US.

IV. Results and Discussion

A. Pre-processing of Tweets

Before cleaning the tweets in the dataset there are 971151 instances in #Trump and 777078 instances in #Biden. After cleaning the tweets there are only 445259 instances left in #Trump and 273246 instances left in #Biden. The instances generally drop about 50% in #Trump and 60% in #Biden every run. The pre-processing of dropping rows of tweets that are not written in English are being compared between a sequential approach and distributed computing approach as in the figure below. It is apparent that the time efficiency on pre-processing of tweets using distributed approach dominant the others by virtue of its architecture. 3

TABLE I
Memory consumption per node of ML model practising various computational framework

	Memory Usages Per Node	
	Random Forest (GB)	**Naïve Bayes (GB)**
Sequential	987	987
Distributed	12.2	0.1389

B. Machine Learning Model-based Sentiment Analysis

1) Memory usage: A comparison between memory usages of both ML models presented within this work has been made where each model has been trained and tested using the entire dataset. Results from the comparison suggested that models implemented in the distributed approach showed superior efficiency in memory usages over models implemented in the sequential approach. Shown in Table I is the amount of memory consumed by each ML model practising different approaches.

Referring to Table I, it is obvious that both models practising sequential approaches have resulted in excessive memory usages. This is due to the nature of sequential approach where execution of workload has been managed solely by the driver node. This scenario has caused a major memory depletion within the compiler that led to a failure of execution of the remaining processes within the pipeline. As for the distributed approach, memory usages for both ML models have demonstrated exceptional competence in handling large volumes of data, hence exhibiting dominance in handling big data problems. One of the main reasons that led to efficiency in the distributed approach is that workload has been split into various partitions, in this case, the partitions have been fixed at 32 partitions. This partitioning of workload portrays the concept of divide and conquer where parallelism in computing can be achieved using various worker nodes present within the computation pipeline. In other words, heavy workload posed within big data problems has been split into chunks of smaller tasks before being directed to worker nodes that facilitates an effective parallel processing of performing training on ML models. Therefore, this exclusive characteristic found within the distributed approach has established that it is indeed the appropriate method in offsetting the heavy, yet time consuming processes offered within big data problems.

Another point that is worth discussing is that the memory consumption for NB model practising the distributed approach showed a significant decrease when compared to RF model implemented in the distributed approach. This drop in memory consumption can be elucidated with the size of both ML models. To contrast, the RF model has a substantially greater model size when compared to the NB model. Therefore, it is clear that a larger model will certainly require more memory capacity to perform its training.

2) Time consumption: After testing and evaluating with both sequential and distributed approaches for handling high volume of data, the process is timed from the loading of

datasets to after the evaluation of the prediction model. By observing Figure 4 and Figure 5 which shows the graph of number of data against the time taken to finish the training and evaluation for RF and NB, it is observed time taken for the sequential approach increases relatively linearly as the number of instances increases.

When there are only 2000 instances, the sequential approach performed better than the distributed approach. The cause of this happening is due to the computational overhead required for distributed computing approach. When size of dataset does not exceed a threshold, the benefits of distributed computing does not offset the computational cost incurred. Thus, resulting in sequential computing approach outperforming distributed computing approach.

However, upon reaching 10000 instances and 30000 instances, the time it takes for each method to finish the process in RF and NB intersect and that distributed approach method can be seen performing better than sequential approach. As the number of data surpasses a certain threshold, the computational cost incurred becomes justifiable and the benefits of distributed computing becomes apparent. This shows that a sequential approach can perform relatively well with smaller datasets, but when the size of the dataset exceeds a certain threshold, a distributed approach proves to be more efficient.

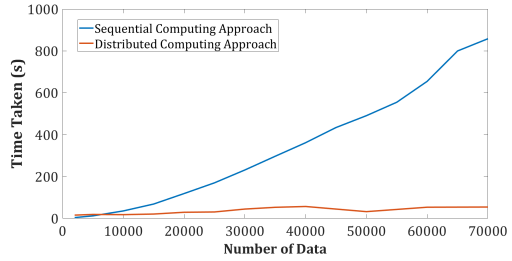

Fig. 4. Graph of number of data against time for Random Forest

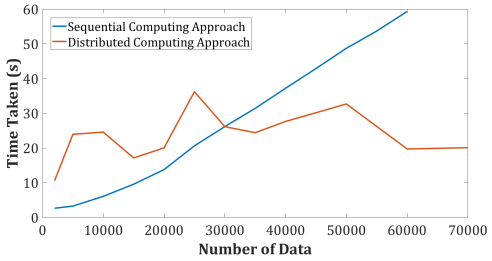

Fig. 5. Graph of number of data against time for Naïve Bayes

3) Comparison of Models' Accuracy: Due to the aforementioned dilemma of excessive memory usage, a fair comparison between the performance evaluation of both ML models practising different computational approaches has been made using a total of 35000 data instances. The reason for using 35000 data instances is because a series of trial and error suggested that any data instances greater than 35000 will exceed the memory constraints offered by the compiler to execute the pipeline. Table II displays the performance evaluations of each ML model on 35000 data instances that practises different computational approaches.

TABLE II
PERFORMANCE EVALUATION OF ML MODEL PRACTISING VARIOUS COMPUTATIONAL FRAMEWORK IN TERMS OF ACCURACY

	Accuracy	
	Random Forest	**Naïve Bayes**
Sequential	0.7420	0.3920
Distributed	0.4800	0.6800

By referring to Table II, it is obvious that the accuracy generated by RF practising sequential approach has a better performance compared to its corresponding contestant practising distributed approach. An inference has been made based on this finding, that is the nature of the distributed approach tends to split dataset into smaller subsets where each of these subsets will be allocated to a worker node for training of the classification model.With that being said, it is obvious that training of classification model within each worker node utilising smaller subset of dataset will certainly influence the ability of each model in making predictions. For the case of Naïve Bayes, accuracy of the distributed approach has outperformed the accuracy presented by the sequential approach.

However, training of ML models using the entire dataset has been conducted using the distributed computing framework, where RF model and Naïve Bayes model have resulted in have resulted in 0.4223 and 0.7100 in prediction accuracy. This observable drop in accuracy in the distributed RF model trained using the entire dataset compared to the distributed RF model trained using 25000 dataset can be elucidated with the case wehre a greater amount of newly encountered words have been supoplied as inpout features into the training of the model. Hence, the greater amount of features presented have caused the difficulty to the model in realising the underlying pattern within the dataset.

C. Result and Discussion of Sentiment Analysis

1) Tweet-based Sentiment Analysis: To begin with, the subset of tweets that are tweeted by US citizens are extracted from the dataset. Then, the filtered data is grouped according to the tweets' sentiment polarity (positive, negative, neutral). The resulting histogram is shown in Figure 6. As portrayed from the figure, there exists a significant difference between the amount of negative sentiment tweets related to respective nominees. Following that, the result is grouped by the state where the tweets are posted from. The result reveals that, in most states, the number of positive tweets that are related to DT is more than JB. Assuming that as election result, the electoral vote of DT is recorded to be 434 and JB is 101.

2) User-based Sentiment Analysis: Since each citizen only holds a single vote in the actual election, the analysis is repeated with an additional filtering of removing tweets that

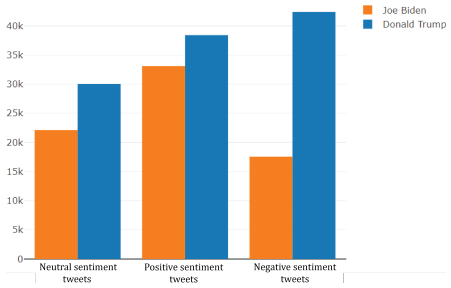

Fig. 6. The histogram of tweets' sentiment on both nominees

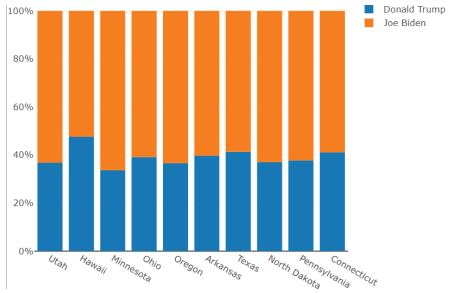

Fig. 7. The histogram of top 10 states users' sentiment on both nominees.

are sent by repeating users. The sentiment histogram is presented in Figure 8 From the histogram, it can be observed that the number of tweets related to DT appears to be lesser as compared to Figure 6. The percentage of positive and negative tweets are almost equivalent. However, the sentiment histogram for JB remains identical as shown in Figure 6. When the positive sentiment tweets are grouped by the state, the result in Figure 7 appears to be contrary to previous analysis as the number of positive tweets related to JB are generally dominant across all states and causes the electoral vote of DT is recorded to be 12 and JB as 523. An additional filtering step is applied on the dataset to remove all 'fake' account that are created during the election period. However, upon analysing, only a negligible number of tweets in the collected dataset satisfies the criteria. Ergo, it does not affect much on the analysis.

3) Correlation between Sentiment Polarity on Twitter and Actual Election Result: Upon calculation, the Pearson Correlation Coefficient of the ratio of tweet supporters between nominees and ratio of actual election supporters appears to be 0.0551 which indicates weak associativity between tweet support and actual election support. In other words, our study concludes that the sentiment polarity of tweets on Twitter is insufficient to make a reliable prediction of the result of the election.

D. Conclusion

In short, our work has conducted a comprehensive sentiment analysis on Twitter concerning the sentiment polarity of Twitter users towards both US election nominees and correlation between the analysis and actual US election result 2020. We have proposed a different sentiment analysis approach

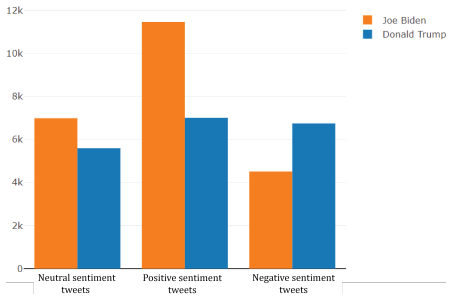

Fig. 8. The histogram of users' sentiment on both nominees

namely the lexicon-based and ML-based approach. Various ML classifiers have been experimented and compared in terms of their performance. Our work suggested that the performance of NB classifier outperformed RF classifier with the accuracy of 0.71. Both these methodologies were implemented in a sequential and distributed computing framework to compare their memory usage per node and their time consumption. In general, distributed computing framework uses lesser memory footprint per node with lower time consumption. Following that, statistical analysis has been carried out to study the result of sentiment analysis and its correlation to actual election results. By using Pearson Correlation Coefficient formula, our study reveals that sentiment polarity on social media does not correlate to the actual US election result in 2020.

REFERENCES

[1] Statista, "Twitter: Number of users worldwide 2020," 1 2022. [Online]. Available: https://www.statista.com/statistics/303681/twitter-users-worldwide/

[2] D. Sehgal and A. K. Agarwal, "Sentiment analysis of big data applications using Twitter Data with the help of HADOOP framework," in *Proceedings of the 5th International Conference on System Modeling and Advancement in Research Trends, SMART 2016*. Institute of Electrical and Electronics Engineers Inc., 4 2017, pp. 251–255.

[3] S. Sioutas, G. Tzimas, N. Nodarakis, and A. Tsakalidis, "Large Scale Sentiment Analysis on Twitter with Spark," 3 2016. [Online]. Available: https://www.researchgate.net/publication/295636747

[4] H. Elzayady, K. M. Badran, and G. I. Salama, "Sentiment Analysis on Twitter Data using Apache Spark Framework," in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, 2018, pp. 171–176.

[5] Z. Karhan, M. Soysaldı, Y. Özben, and E. Kılıç, "Sentiment Analysis On Twitter Data Using Distributed Architecture," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, 2018, pp. 357–360.

[6] P. Sharma and T.-S. Moh, "Prediction of Indian election using sentiment analysis on Hindi Twitter," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 1966–1971.

[7] A. Rane and A. Kumar, "Sentiment Classification System of Twitter Data for US Airline Service Analysis," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 01, 2018, pp. 769–773.

[8] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, p. 107134, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095070512100397X

[9] "Presidential Results," 11 2020. [Online]. Available: https://edition.cnn.com/election/2020/results/president

[10] K. Pearson, "Contributions to the Mathematical Theory of Evolution," Tech. Rep., 1894. [Online]. Available: https://about.jstor.org/terms

[11] Manch Hui, "US election 2020 tweets," 11 2020. [Online]. Available: https://www.kaggle.com/manchunhui/us-election-2020-tweets