Unveiling Twitter Sentiments: A Big Data Dive into Twitter Sentiments during the 2020 US Elections

Arnaaz Khan, Arun V.S., Deniz Sagmanli, Harish Mohan Babu, Muhammad Hassan Rizvi, Saanidhya Khurana
COMP4124 Big Data Learning and Technologies
University of Nottingham, Nottingham, UK
Emails: {psxak20, psxav7, psxds8, psxhm11, psxmr4, psxsk14}@nottingham.ac.uk

***Abstract*—This study uses cutting-edge natural language processing and machine learning methods to examine Twitter sentiment related to the 2020 US presidential election. Using only the election-related tweets from Kaggle, we created an extensive dataset. To improve its quality and suitability for sentiment analysis, the dataset underwent a thorough preprocessing process that involved removing stop words, non-standard tokens, special characters, emojis, and hashtags. We used TextBlob to apply a more sophisticated approach to sentiment classification, dividing tweets into positive, negative, and neutral categories. We created a novel custom machine learning model to reclassify naturally complicated neutral sentiments into clear-cut positive or negative categories, increasing the granularity of our sentiment analysis and mitigating the difficulties it presented. Analyses comparing this model to Spark's MLlib showed that it performed better in terms of fault tolerance and scalability and that its mean absolute error accuracy was comparable. The fact that both systems correctly anticipated Joe Biden's win shows how accurate it is to use sentiment analysis on Twitter to forecast voter behaviour. This validation demonstrates the usefulness of our techniques and the possibility of in-depth sentiment analysis in predicting political outcomes. It also opens the door for further research into regional sentiment analysis and the incorporation of a wider range of data sources.**

I. Introduction and Background

Social media platforms become essential forums for political discussion and public discourse in the modern day. Social media sites like Twitter allow for immediate contact and are vital resources for participating in politics.

It is commonly known that social media impacts how political narratives are shaped and how this influences election results. Twitter is a main source of political updates, with about 60% of users participating in political debates on the network [1]. Twitter can potentially be a more accurate predictor of political outcomes than traditional polling techniques, which frequently have biases and delays due to their real-time insights into public reactions to events.

Twitter's importance in the 2020 U.S. Presidential Election is highlighted because it was the main vehicle for voter mobilisation and campaign communication. Social media activity spiked as election day drew nearer, and COVID-19 caused an unprecedented volume of mail-in ballots to postpone the projection of a winner [5], resulting in an increase in election-related Twitter traffic until the results were announced four days later. Sentiment analysis can be used with this data to predict election results and learn more about public opinion. Sentiment analysis in NLP helps to understand dispositions towards a figure or event [4] by classifying emotions and ideas as positive, negative, or neutral [2]. Online sentiment analysis reveals user attitudes and the ways in which social media sites like Twitter impact political movements.

This study investigates the relationship between Twitter sentiments and election outcomes using sentiment analysis. To determine who the more popular candidate was in the 2020 U.S. Presidential election, we take a look at the ratio of positive tweets to all tweets regarding Donald Trump and Joe Biden. We also use machine learning methods to reclassify neutral sentiments. Our method includes training three Logistic Regression models locally, aggregating the results, and using a locally distributed machine learning model. We demonstrate our novel research methodology by comparing our strategy with Spark's MLlib to the sequential approach in terms of mean absolute error, fault tolerance, and scalability.

II. Proposed Methodology

Figure 1 presents a flowchart outlining the proposed methodology for sentiment analysis of Trump and Biden-related tweets. The process begins with data collection, followed by text cleaning and filtering through pre-processing. We employ TextBlob for sentiment analysis, categorising tweets into three categories: neutral, negative, and positive. A driver node coordinates model training and broadcasts the models to worker nodes for tweet classification in a distributed system. The driver node then receives an aggregate of all node results for final analysis. The outcomes are evaluated by comparing sentiment ratios with actual election results, providing valuable insights into the model's scalability and accuracy. This comprehensive approach offers a deeper understanding of the impact of social media sentiment on political movements, enlightening the audience about the intricate dynamics at play.

A. Data Pre-Processing

The data collected from the Kaggle website [7] was divided into two files for the tweets related to the two leading candidates running for the United States presidency in 2020, Donald Trump and Joe Biden. After reading the files, it was noticed that the data for the trump file had approximately just over 2 million rows, while the Biden file had around 1.8 million rows. Both dataset files were pre-processed individually, and the following steps were taken to prepare them for this study.

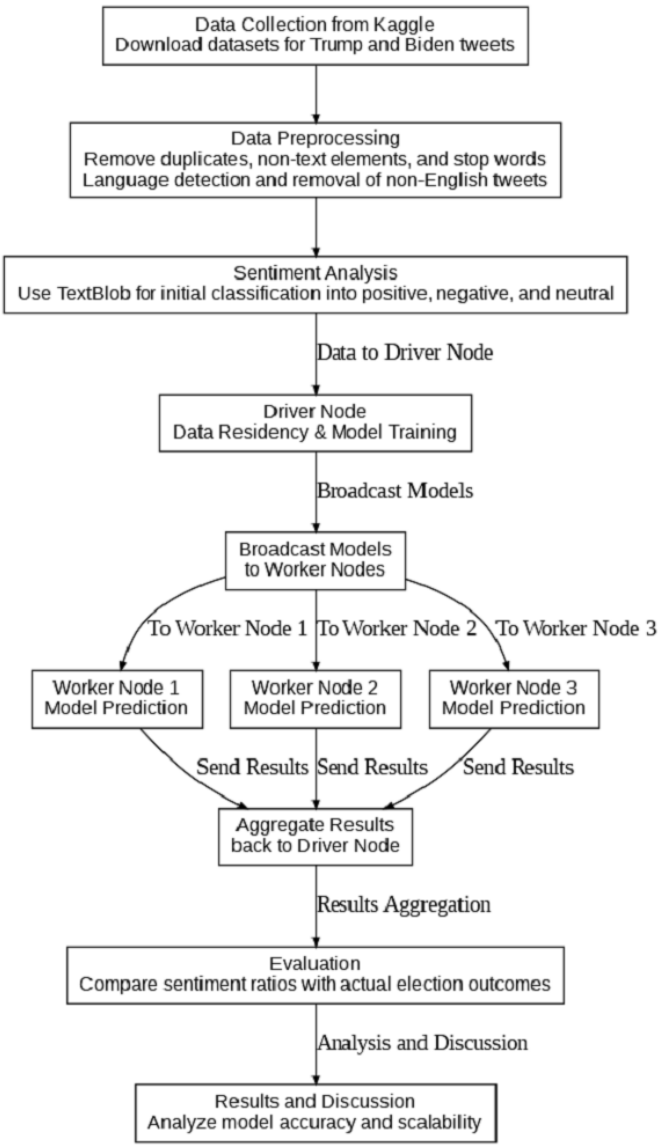

Fig. 1. Proposed Methodology

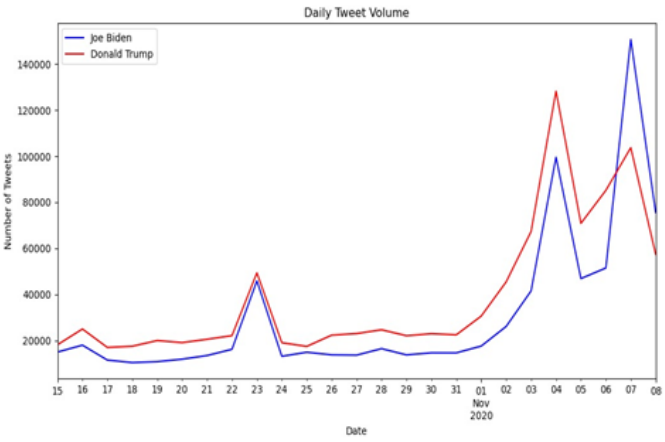

Fig. 2. Proposed Methodology

From October 15, 2020, to November 8, 2020, the line graph in Figure 2 shows the daily volume of tweets sent out for Joe Biden and Donald Trump. It peaks around the election on November 3. Tweets about Trump are often more than Biden's, especially in the crucial hours leading up to electoral Day, demonstrating the calculated use of the social media platform throughout pivotal electoral periods.

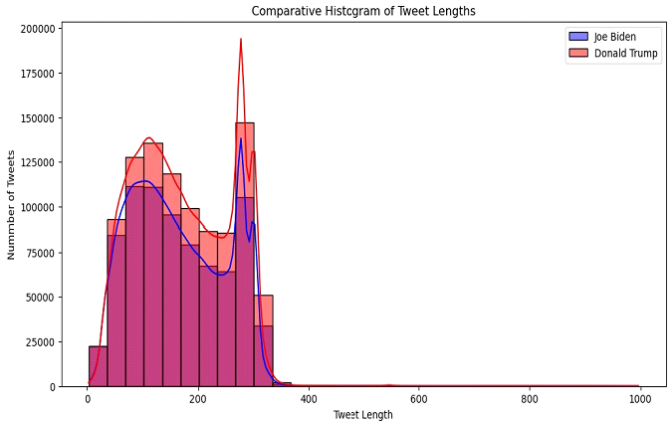

Fig. 3. Proposed Methodology

When comparing the tweet lengths for Donald Trump and Joe Biden, the histogram reveals that both primarily use the 140-character maximum Twitter has historically allowed. Interestingly, tweets about Trump are closer to the updated 280-character limit, suggesting a different way of using Twitter's increased capacity. This can be seen in Figure 3.

The datasets were then assessed for unique values to spot any potential anomalies or redundancies. Using distinct values was the main premise of this technique, as it was thought that retweets did not always imply agreement with the content. The investigation intended to better properly depict individual thoughts and ideas on Twitter by concentrating only on actual user-generated information. Furthermore, tweets were screened to exclude whitespace or missing tweets, guaranteeing that the datasets contained only useful information. The timestamps of every tweet that remained to be confirmed to be in the right format were validated next. Misclassified tweets were appropriately reassigned, and incorrect formats were reset to none.

Text cleaning is an essential pre-processing step in sentiment analysis that refines raw text data by removing non-textual elements like emoticons, URLs, HTML tags, square brackets with metadata, and social media artefacts like mentions, hashtags, and retweets. It also converts all characters in the text to lowercase. Frequently, these components add background noise that hinders the main textual materials, which are emotional clues. The cleaned text is better suited for sentiment analysis when these distractions are methodically removed, making it possible for algorithms to recognise and understand the underlying sentiments. This makes sentiment analysis results more informative and accurate.

Multiple measures were made to refine the datasets for significant insights in the later phases of the pre-processing pipeline. The WordNinja Library separated combined terms in tweets, leveraging English frequency statistics for better readability and text analysis. A language detection mechanism was also created to identify and exclude non-English tweets to ensure high efficacy in sentiment analysis. By eliminating common but semantically meaningless stop words, the sentiment analysis algorithm could concentrate on the most important terms in each tweet. As a result, the analysis's accuracy was improved by the sentiment classification's increased precision and clarity. The length of tweets, word counts, and average word counts were among the metrics calculated to highlight the important textual properties of machine learning models.

An essential component of this research is sentiment analysis, which looks for the underlying feelings and viewpoints in tweets. For this work, the well-known Python package TextBlob was employed. Subjectivity and polarity are two crucial sentiment indicators that are successfully extracted. Subjectivity gives information about how to strike a balance between one's own viewpoint and the truth, with a range of 0 (total objectivity) to 1 (total subjectivity). A tweet's polarity can be used to determine its emotional tone. It can range from -1 (totally negative) to +1 (totally positive). Based on their polarity scores, tweets were classified as "positive," "neutral," or "negative" attitudes using these measures. A methodical examination of the datasets was made easier by this organised format.

B. Model Building

The primary objective of this project is to devise a big data approach to address central research: "Do Twitter sentiments accurately mirror real-life election outcomes?" This research is anchored in the assumption that the candidate with a higher ratio of positive tweets is likely to emerge victorious in the election. To evaluate this hypothesis, TextBlob was deployed to categorise sentiments into positive, negative, and neutral labels, subsequently leveraging these labels to gauge the accuracy of their machine learning (ML) model, measured by mean absolute error (MAE). Initially, the sentiment classification suffices for assessing the research question; however, to enhance the results of whether there is any relation between the real-world election outcomes and Twitter sentiments, a machine learning aspect was introduced in which classifying neutral tweets into positive and negative categories was conducted thereby enriching the ratio calculation. Initially, the analysis focused on determining the ratio of positive tweets to all tweets per candidate. However, the neutral tweets were not accounted for in this calculation. Hence, the decision was made to classify neutrals into positive and negative categories. Therefore, the formula was revised to include the sum of positive and positive-predicted neutral tweets divided by the total number of tweets, providing a more comprehensive assessment for each candidate.

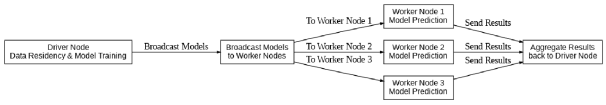

Fig. 4. Proposed Custom Big Data Approach

A sequential MLlib logistic regression model was initiated to explore machine learning aspects to predict sentiment polarity (positive or negative). Logistic regression was chosen for this task due to its effectiveness in binary classification problems, where the goal is to predict the probability of an observation belonging to one of two classes [12]. This method is particularly suitable for sentiment analysis, as it allows for predicting sentiment polarity (positive or negative) based on input features extracted from the text data [13]. Additionally, logistic regression is computationally efficient and less prone to overfitting than more complex models, making it a practical choice for analysing large datasets such as those commonly encountered in sentiment analysis tasks [14]. This entails training and testing the model exclusively on positive and negative sentiment rows, with neutral sentiments introduced as unseen data for prediction. A custom approach in terms of big data was developed, the local approach, in which three logistic regression models were trained concurrently across different nodes and aggregated their predictions through a majority voting mechanism. The novel ensemble approach potentially enhances accuracy and ensures fault tolerance. The sole reason for selecting the local approach was its reduced necessity of in-depth knowledge of individual models. This was chosen to streamline the process and mitigate the complexities of managing multiple models independently. It was also recognised that applying a global approach in future can optimise the process further by minimising data movement and enhancing efficiency [10].

The flowchart in Figure 4 summarises the custom approach, which involves distinct stages where data movement occurs. Initially, the data resides within the driver node, which trains three separate models, each within its designated worker node. Subsequently, these three trained models are broadcast from the driver node to the respective worker nodes. This broadcast mechanism facilitates fault tolerance, ensuring continued operability in the event of node failure, and signifies a significant instance of data movement within the system. Following model predictions conducted independently in each worker node, the resulting predictions are gathered and communicated back to the driver node for aggregation. This process depicts the data flow between the central driver node and the distributed worker nodes, illustrating how the three models are trained and interact within the distributed computing environment.

Furthermore, runtime comparisons were conducted between processing 10% and 20% of the data to assess scalability. The goal is to determine if the runtime for processing 20% of the data is twice that of processing 10%, indicating linear scalability. Comparing runtimes between the processing times for both data sizes suggests a nearly linear scalability of their

approach. This scalability assessment is quantified using the formula for size-up, where the runtime for a given dataset size (n) is divided by the runtime for the entire dataset.

In the concluding stages, the methodology was applied separately to tweets about each presidential candidate, Trump and Biden. By augmenting the initially identified positive sentiments with the predicted positive sentiments, the ratio comparisons were conducted to prove that Twitter sentiments indeed correlate with real-world election results.

III. Experimental Set-up

After doing heavy pre-processing on the datasets, we used TextBlob to get each tweet's subjectivity, polarity, and label, which can be either positive, negative, or neutral. We used these results as the input to our Machine Learning models. Our critical comparison was between a regular MLlib sequential logistic regression model and our custom local approach. We removed some unnecessary columns to ensure the input was the same for both sides of the comparison. For both methods, we divided our primary dataset into two. We used the dataset with just the positive and negative sentiments to train and test our model and teach it what needs to be positive and what needs to be negative. Then, we used our model to predict the sentiment for the second dataset, which only consisted of neutral tweets, to further classify them into positive and negative. We did this because our ratio assumption also needed to include neutral tweets. We calculated the mean absolute error during the training and testing phases, as it is a relevant metric in classification problems. After classifying neutrals into positive and negative, we calculated the proportion of positive tweets using the formula below with an assumption that the candidate with a higher positive versus total tweets ratio would win the election.

$$\text{Proportion of Positive Sentiments} = \frac{P + N_P}{T} \tag{1}$$

where P represents the number of positive tweets, N_P represents the number of positive-predicted neutral tweets, and T is the total number of tweets.

Regarding comparing the sequential MLlib approach and our local approach, we used three scikit-learn logistic regression models on different nodes and aggregated the results using a majority vote system. We used scikit-learn models as MLlib was not optimised enough and kept throwing errors. For instance, we chose the positive if two models predicted positive and one predicted negative. We broadcasted our predictions, and this provided fault tolerance in case any worker node failed. Our main aim was to minimise mean absolute error using three predictions. We trained the models on the driver node and broadcast them to the workers so the trained models would be recovered. We just made predictions for the workers.

After calculating the ratio, we used size-up as our scalability metric to prove that our method was scalable. To measure scalability, we used our model to train and test just 10% of the data and recorded the runtime. We then did the same for 20% of the data. The assumption was that the runtime should linearly relate to the data size.

$$\text{size-up}(n) = \frac{\text{runtime to process } n \text{ data}}{\text{runtime to process all data}} \tag{2}$$

Doing all of this, we compared the accuracy and fault-tolerance of our custom local approach to the sequential MLlib approach. We also aimed to prove that our custom local approach was scalable. Finally, we checked if the resulting ratio of either approach correlated to the actual election results. We ran these experiments separately for Donald Trump's and Joe Biden's tweets.

IV. Results and Discussion

A. Performance Comparision

After the pre-processed TextBlob output was ready, we processed it further. We fed it into Spark MLlib's sequential logistic regression model and our own local Big Data machine learning algorithm.

TABLE I
Performance Comparison of Sentiment Analysis Approaches

Model	Tweets related to Donald Trump		Tweets related to Joe Biden	
	MAE	Positive Tweets%	MAE	Positive Tweets%
MLlib Logistic Regression	0.3606	81.97%	0.2940	85.40%
Custom Big Data Approach	0.3567	81.81%	0.2945	85.40%

It can be observed from **Table I** that both the implemented approaches have comparable accuracy for both datasets. It is important to note that the Trump dataset had significantly more tweets after being cleaned. Considering that our approach had slightly less error for Trump's tweets and that this result did not change over multiple trials, it can be inferred that more data may lead to a higher difference in mean absolute error. Nevertheless, we have not tested this inference due to the lack of extra tweets, but further studies can focus on this.

the Spark MLlib approach, the ratio of positive (including positive-predicted neutrals) + all tweets for Donald Trump turned out to be approximately 0.8197. This number was around 0.8540 for Joe Biden using the same approach. For our local approach, we obtained around 0.8181 for Donald Trump and close to 0.8540 for Joe Biden.

The two approaches almost yield the same proportions of positive tweets; Joe Biden has more positive tweets percentage than Donald Trump. In the Spark MLlib approach, the proportion of positive tweets is 81.97% for Donald Trump and around 85.4% for Joe Biden. Whereas it's 0.8181 for Donald Trump and 85.4% for Joe Biden in the custom local approach. According to our assumption, the candidate with higher positive tweets is likely to win the elections, Twitter data reflected real-life election outcomes. Although one study is insufficient to conclude that Tweet sentiments can accurately

predict the results of every election, they certainly did for this one. Further studies can divide the tweets into the states they were tweeted from and try to see if sentiments correlate to the results in each state.

B. Scalability

We needed to prove that our design was scalable. For the scalability metric, we chose size-up. For both Donald Trump's and Joe Biden's tweets, we separately trained and tested 10%, 20%, and then the entire data set.

TABLE II
PERFORMANCE METRICS FOR SCALABILITY TESTING

Performance metrics	**Donald Trump Dataset**			**Joe Biden Dataset**		
	10%	**20%**	**100%**	**10%**	**20%**	**100%**
% of Data	10%	20%	100%	10%	20%	100%
Runtime (in sec)	22	41	46	35	11	39
Size-up	0.478	0.891	-	0.897	0.282	-

As demonstrated in **Table II**, the runtime for both candidates was almost linearly related to the data size, confirming the scalability of our solution.

For Trump, training and testing 10% of the data for Trump took 22 seconds, 20% took 41 seconds, and the entire data took 46 seconds. Size-up(10%) was 0.478, and size-up(20%) was 0.891 when we ran our experiment on tweets about Donald Trump. Regarding Joe Biden, 10% of the data took 35 seconds to train and test, while 20% took 11 seconds, and the entire dataset took 39 seconds. Size-up(10%) yielded 0.897 and size-up(20%) yielded 0.282. The inconsistency in Joe Biden's runtime was most likely due to the heavy workload on the cluster due to other people using it at that time. It can be concluded that according to the size-up metric, runtime was almost linearly related to the data size. This proved that our solution is indeed scalable.

C. Fault Tolerance

We have observed that fault tolerance was genuinely supplied by our new strategy. There would be no need to retrain a model in the event of a node failure because the trained logistic regression models were broadcast among all nodes. Furthermore, to avoid having to re-predict any node failure, we additionally broadcast the forecasts that were made at each node. Our technique outperformed the sequential approach by a lot since it is fault-tolerant.

Reducing the amount of data moved across the network was one of our objectives. Since broadcasting necessitates transferring data, there must be a trade-off between fault tolerance and minimum data movement. The driver node is where we initially trained the models. For the predictions to come true, we subsequently broadcast them to the appropriate workers. While the predictions were being made, there was no data movement. However, we further broadcast each node's predictions as a precautionary measure, making them accessible to other nodes. We finally combined them and brought them back to the driver. Moving the data was also necessary for the collection process. We maintained our design more simplistic but rigid even though it could have prevented certain data movement and increased the likelihood of node failure in our system.

We could train numerous models instead of just one by using a local method, which helped us develop a divide-and-conquer strategy. The primary benefit was that it did not necessitate an in-depth understanding of the logistic regression model. It was, therefore, easier to implement. That is not to argue that there are no limitations when employing a local strategy. Given that we had to train numerous models and partition the workflow, a global solution could have been more effective regarding runtime and data movement. Thus, the number of partitions affects how well a local solution performs. A global approach would have acted more like a sequential model without being affected by the number of partitions.

Because it was an ensemble model, our technique was innovative because it was a Random Forest with Logistic Regression or Random Forest Regression. Our research indicates that this solution has not yet been applied in a big-data way.

Our model can be further enhanced by adding TextBlob to perform the initial data processing with a substantially shorter runtime, and by using a global approach with TextBlob for data pre-processing rather than relying on scikit-learn to reduce the runtime. We did not pay attention to this because the scope of our Big Data technique included the additional categorisation of neutral tweets as either positive or negative to include them in our computation of the positive-to-all tweets ratio for each candidate.

V. CONCLUSIONS

We thoroughly analysed Twitter opinions around the 2020 US presidential election using cutting-edge machine learning techniques to explore the intricate relationship between electoral dynamics and online debate. We gleaned insightful information from a vast corpus of tweets by carefully pre-processing the data and conducting sentiment analysis. This allowed us to illuminate any possible relationship between Twitter's feelings and election results. Data pre-processing to guarantee data integrity, sentiment analysis to divide tweets into positive, negative, and neutral sentiments, and machine learning models to further classify neutral sentiments and examine sentiment ratios for each candidate were all essential components of our methodology.

Our experiments yielded some remarkable results. Despite the inherent difficulties brought about by the volume and noise of social media data, our machine-learning model showed similar levels of sentiment categorisation accuracy. Our proprietary local technique and the Spark MLlib sequential logistic regression model both showed comparable results in terms of mean absolute error. Ultimately, our solution proved to be scalable. Additionally, real-world election results closely matched our analysis of sentiment ratios for each candidate using Twitter data, indicating that Twitter sentiments did represent more considerable public opinion and political trends.

This highlights the promise of social media analytics, albeit with some restrictions and cautions, as a tool for predicting election results and assessing public sentiment.

Specifically, our unique local approach—which uses fault-tolerant techniques and ensemble learning—exhibited better scalability and fault tolerance compared to conventional sequential models. Although it came with inevitable trade-offs related to data mobility, its divide-and-conquer strategy worked well for processing massive amounts of data. There are still opportunities for development and investigation. To gain a more thorough understanding of electoral dynamics, future studies could improve machine learning algorithms, investigate sentiment analysis at a more regional level (such as provincial sentiment analysis), and incorporate more data sources. Furthermore, developments in big data technologies and approaches present opportunities to improve model performance and optimise data processing pipelines. Finally, our research adds to the expanding corpus of work on social media analytics and its applications to electoral analysis and political discourse. We have unearthed essential insights into the complex relationship between Twitter attitudes and actual events by utilising cutting-edge machine learning algorithms and significant data approaches, opening the door for more research in this intriguing field.

REFERENCES

[1] Le, H., Boynton, G., Shafiq, Z., Srinivasan, P. *A postmortem of suspended twitter accounts in the 2016 US presidential election*. In: 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2019; pp. 258–265. IEEE.

[2] Feldman, R. *Techniques and applications for sentiment analysis*. Communications of the ACM. 2013;56(4):82–9.

[3] Liu, B. *Sentiment analysis and opinion mining*. Synthesis Lectures on Human Language Technologies. 2012;5(1):1–167.

[4] Mukherjee, S., Bhattacharyya, P. *Feature specific sentiment analysis for product reviews*. In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 475–487, 2012. Springer.

[5] Gupta, V., Piryani, R., Singh, V. K., Ghose, U. *An analytical review of sentiment analysis on twitter*. Advances in Computing, Control & Communication Technology. 2016;1:219–25.

[6] Ali, R. H., Pinto, G., Lawrie, E., et al. *A large-scale sentiment analysis of tweets pertaining to the 2020 US presidential election*. Journal of Big Data. 9, 79 (2022). https://doi.org/10.1186/s40537-022-00633-z

[7] kaggle.com. *US Election 2020 Tweets*. [online] Available at: https://www.kaggle.com/manchunhui/us-election-2020-tweets.

[8] Hazarika, D., Konwar, G., Deb, S., Bora, D. J. *Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing*. Annals of Computer Science and Information Systems. https://doi.org/10.15439/2020km20.

[9] TextBlob. *TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation*. [online] Readthedocs.io. Available at: https://textblob.readthedocs.io/en/dev/.

[10] Basgall, M. J., Hasperué, W., Naiouf, M., Fernández, A., Herrera, F. *An Analysis of Local and Global Solutions to Address Big Data Imbalanced Classification: A Case Study with SMOTE pre-processing*. Communications in Computer and Information Science, pp. 75–85. https://doi.org/10.1007/978-3-030-27713-0_7.

[11] Ogdol, J. M. G., Samar, B.-L. T., & Cataroja, C. *Binary Logistic Regression based Classifier for Fake News*. URL: https://www.researchgate.net/publication/332341870_Binary_Logistic_Regression_based_Classifier_for_Fake_News.

[12] Hastie, T., Tibshirani, R., Friedman, J. H. *The Elements of Statistical Learning*. Springer. 2nd Edition.

[13] Pang, B., & Lee, L. *Opinion Mining and Sentiment Analysis*. Foundations and Trends® in Information Retrieval, 2(1–2), 1–135. DOI: 10.1561/1500000001

[14] Harrell, F. E. *Regression Modeling Strategies*. Springer Science & Business Media.