TweetPulse: A Power BI-Driven Interactive Sentiment Analysis Dashboard for Brand Monitoring

Ajith M Joshy
Department of Computer Science and Engineering
Saintgits College of Engineering
Kerala, India
ajith.csa2125@saintgits.org

Jerry Narakathara Thomas
Department Computer Science and Engineering
Saintgits College of Engineering
Kerala, India
jerry.csa2125@saintgits.org

Anitta Mary Jose
Department Computer Science and Engineering
Saintgits College of Engineering
Kerala, India
anitta.csa2125@saintgits.org

Talit Sara George
Department Computer Science and Engineering
Saintgits College of Engineering
Kerala, India
talit.sg@saintgits.org

***Abstract*—Today we live in a world where everything has become data and is keeps on growing at a fast rate. In various industries, data has become important for a variety of applications. Brands when they release a product or before releasing the product want to know public opinions on how satisfied the customers are with their brand or their products. Sentiment analysis has become one of the most important and crucial tool and it analyze data from sources where public data is available for understanding public sentiments and it has been the fast growing area. Social media platforms has a rich source of data where companies use it for doing sentiment analysis. This paper reviews recent advancements in sentiment analysis techniques, approaches and comparing different models used. The paper also reviews a proposed real-time system that collects recent data and classifies sentiments and then view various trends and insights in an interactive dashboard, which can help companies to monitor their brand and perform various analytics and take timely decisions based on the insights.**

***Index Terms*—Sentiment analysis, Natural language processing, Social media analytics, Machine learning, Real-time analysis, Twitter data mining, Data visualization,**

I. Introduction

In recent years, sentiment analysis has emerged as a powerful tool for understanding human emotions and opinions in digital text [1]. As businesses, governments, and researchers seek information about public opinion, sentiment analysis offers an efficient way to process large volumes of data from various online sources. By categorizing opinions as positive, negative, or neutral, sentiment analysis enables decision makers to gauge public sentiment, adjust strategies, and even predict trends.

Social media platforms, particularly Twitter [2], provide a unique opportunity for sentiment analysis due to their massive, real-time data streams and public accessibility. Twitter users instantly share reactions to events, brands, and policies, creating a continuous and dynamic flow of opinions. This makes Twitter an ideal data source for studying public mood and sentiment on various topics, from politics and entertainment to product feedback.

The evolution of sentiment analysis techniques has been driven by advances in machine learning and natural language processing (NLP) [3]. Modern NLP models, such as BERT and other transformer-based architectures, have improved the accuracy of sentiment analysis, even in cases where the language is informal, contains slang, or includes nuanced expressions. These improvements have made it possible to apply sentiment analysis not only for static data but also in real-time, which is valuable for applications such as monitoring brand reputation, tracking public reactions during events, or identifying social trends as they emerge.

However, sentiment analysis on platforms such as Twitter also poses significant challenges [4]. The brevity and informality of tweets, combined with the frequent use of emojis, abbreviations, and non-standard spellings, require sophisticated preprocessing and robust models. Moreover, analyzing sentiment in real-time demands efficient data handling and visualization methods, as well as the ability to derive insights that are both accurate and actionable.

This study reviews the approaches, technologies, and techniques used for sentiment analysis and presents a real-time sentiment classification system. The paper is categorized into different sections as Section I gives the introduction of the paper, Section II gives the review of papers, Section III talks about the proposed system, Section IV shows how the system is implemented, Section V shows how models are evaluated and the system is tested, Section VI talks about the limitations of the system, Section VII discuss the results obtained from analysis, Section VIII concludes and outlines future work.

II. Related Work

The sentiment analysis can help in product improvement, marketing strategy development and brand image management. It helps brands like Huawei monitor and analyze customer sentiment on social media platforms in an automated way [5]. Support Vector Machine is used in classifying sentiment of user comments related to 'Huawei Mate60' as the results indicate SVM is more effective in obtaining an accurate sentiment classification compared to other four models, Logistic Regression, K-Nearest Neighbors and Naive Bayes and XGBoost.

Brand reputation management and marketing strategy has become more and more important in the digital age [6]. The brands can comprehend and respond to changing public sentiment by using information retrieval techniques, such as calculating, textual similarity and using machine learning methods to analyze product similarity for competitive analysis. The objective is not only to monitor public sentiment towards a brand, but also to provide valuable insights by comparing it with similar brands in the market. Through user-friendly visualizations and tools, the brands stay abreast of public opinion and make informed decisions amid the dynamic landscape of consumer sentiment and competition, providing practical solution for brand reputation management and marketing strategy.

In social media platforms there are many hate speeches posted by the users. This hate speech can be detected using the system [7] that streams real-time Twitter data of a trending topic and classifies it into two classes: hate speech and non-hate speech. The model used in the system is based on Long Short Term Memory (LSTM) model, using term frequency inverse document frequency (TF-IDF) vectorization, and compares its performance with other machine learning and deep learning models such as Support Vector Machine (SVM), Naive Bayes (NB), Logistic Regression (LR), XGBoost (XGB), Random Forest (RF), K-Nearest Neighbor (kNN), Artificial Neural Network (ANN), and Bidirectional Encoder Representations from Transformers (BERT). The LSTM model achieved an accuracy of 97% in detecting hate speech, outperforming the other models.

During COVID-19 pandemic, a total of 999,978 data related to Weibo posts from January 1 to February 18, 2020 was collected to understand public sentiment and concerns in China during eary stages of the pandemic [8]. A fine-tuned Bert model was used to classify the sentiment of the posts as positive, negative and neutral, achieving an accuracy of 75.65%. Then topic modelling was applied using TF-IDF to extract key themes discussed in negative sentiment posts. The analysis revealed that people were concerned about the origin of virus, symptoms, impact on work and school, and the public health control measures. These findings could help governments make effective public health decisions during a crisis.

The field of sentiment analysis can be improved significantly by introducing AI-powered approach to this field. It showed that this approach [9] navigates the complexities of sentiment identification and classification in the digital age and also improves decision-making and quality assurance. It also integrates traditional statistical methodologies, such as machine learning models, with advanced deep learning architectures to capture the intricacies of sentiment in a dynamic, multilingual, and culturally diverse digital environment.

There are many techniques to extract topics from the dataset. In this study, [10], MABED and OLDA are the two techniques used for this purpose along with LR and SVM models to address if the topic has positive or negative impact on the event. The overall best score for sentiment analysis of events is obtained when combined MABED with LR.

Naive Bayes algorithm is used in the proposed system, which has got an accuracy of 83%. This system [11] allows users to input keyword, and the system extracts data related to the keyword from Twitter. These data are then classified as positive and negative, and the results are displayed in a pie chart for the users.

E-Commerce has become an important source for user data and an application for sentiment analysis in which users post their opinions about the products [12]. A web based E-commerce application was developed that collects data from Amazon.com and Support Vector Machine(SVM) model was chosen for sentiment classification. The model was trained using NLTK corpus library that contain sentences having negative and positive polarity scores. Flask RESTful API, a web framework is used for making the application RESTful. The application allows the users to submit their review about the product and the submitted reviews are summarized which is then classified as positive or negative polarity in the form of 1 and 0 respectively. Using sentiment analysis in online shopping websites can help customers make educated decision about the product.

III. Proposed System

The proposed system as shown in Fig 1, is designed to collect recent data from Twitter. The data is collected using Twikit, a scraper that interacts with the Twitter API designed only for Twitter. The system streams live tweets using the hashtag #Apple, allowing instant analysis of public sentiment. This hashtag is used specifically for this case study as the system is designed for brand monitoring. So here the system monitors Apple brand. After collecting the data, the data is moved to the preprocessing stage where NLTK techniques such as removing numbers, spaces, emojis, stopword removal, are applied to remove noise from the data (including tokenization and normalization). The feature engineering stage extracts key features using Bert embeddings and Vader to convert the data into numerical form, the form which is only understood by machine learning model. The XGBoost model is used for sentiment classification to predict and classify sentiments as positive, negative, or neutral. The predicted sentiments and various other attributes of the tweets such as timestamp of the tweets, unique tweet id, location of the tweets, tweet text, hashtags and keyword of each tweet are stored in MySQL database. The database is used for efficient storage and retrieval of data that are currently analyzed.

The user interface is the Power Bi dashboard which is visible to the user. It is a web application and it provides dashboard where various analytics and visualizations of data can be performed. This dashboard is connected to the database for transferring the collected data and then visualize various trends and patterns using the data. This application provides different visuals with which various insights can be obtained that helps the user in making timely and correct decisions for their brand, ensuring a reliable and efficient sentiment analysis system operating in real-time. The dashboard and the back-end is connected by FastAPI server. This server is a web framework used for building APIs with python and helps in building applications quickly and efficiently. The dashboard provides a button feature which is provided with FastAPI url and on clicking it sends the request to server. The server recieves and accepts this request which help in automatic pipeline processing.

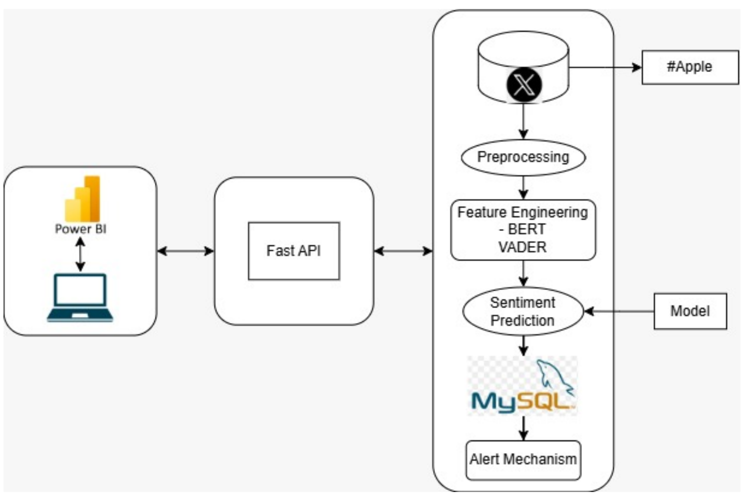

Fig. 1. System Architecture

IV. System Implementation

- **Data Collection**: Fetches recent tweets using hashtag #Apple via Twikit, a python library for interacting with Twitter API. The scraper is configured to authenticate through the Twitter account and saved the cookies once, and then load the saved cookies on subsequent data collection for preventing account ban. Along with tweet, other attributes of each tweet are also fetched such as created_at(timestamp of tweet), location(location of tweet), tweet_id(tweet id), hashtags and keywords. It also filters the tweets by collecting only tweets in english language and discards duplicate tweets, retweets and replies.
  In intial implementation, during a span of 3 months, a total of 4125 tweets was collected. This data for used for training and testing ML models. After the implementation of the back-end and integration of UI and back-end, tweets were collected again for testing the back-end and the system as a whole. The sentiment of tweets were predicted using the best performed model and then visualizing those predicted tweets in the dashboard using different visuals, Power Bi application provides. During the initial implementation, csv files were used for storing tweets.
- **Data Preprocessing**: Processes and cleans the collected tweet to convert it into a form suitable for analysis by removing unnecessary elements(links, stopwords, emojis, space and numbers) and performed tokenization and lemmatization.
- **Feature Engineering** Converts the preprocessed raw tweets from previous module into features that is in numerical form which can be understood by ML models using feature engineering techniques. The two techniques used are Bert and Vader.
  The first feature engineering technique using BERT embeddings. It initializes a BERT tokenizer and model (bertbase-uncased) to process batches of text data, extracting BERT-based feature vectors. The extract_bert_features_batch function tokenizes and feeds batches of text into BERT, capturing the CLS token embeddings, which represent each text's semantic features. These features are then stored in the DataFrame and standardized using StandardScaler for consistency in model training. The standardized BERT features are saved to a file, allowing for easy reuse.
  The second feature engineering technique is VADER for sentiment analysis on text data. First, it initializes the VADER sentiment analyzer. The get_vader_sentiment function applies VADER to each text, classifying sentiment as 'positive', 'negative', or 'neutral' based on the compound score threshold. This function is then applied to the processed_text column, creating a sentiment column with the sentiment labels. For model compatibility, the sentiment labels are encoded into numeric values in the sentiment_encoded column.
- **Class Distribution** Checks the distribution of sentiment classes in the dataset to understand class balance and then visualizes the distribution using a Seaborn count plot, displaying the frequency of each sentiment class. There was class imbalance in the dataset where majority class was positive and minority class was negative. This does not improve our models, so resampling technique is applied for balancing the dataset.
- **Splitting of dataset into Training and Testing** For any ML models two inputs are required for training. Here supervised ML models are used, so a label is required

in order for models to learn. So the first input, that is X input, is the feature set by combining features from vader and bert. Then the second input is the labels, that is, y. First, it concatenates these feature sets from Vader and Bert embeddings into a single matrix feature set, X, and scales them for consistency. y is the target variable(label) for training machine learning models. Both Vader and Bert embeddings are in different ranges, so StandardScaler is used in order to standardize them. This ensures all features are on the same scale preventing any single feature from dominating the learning process due to its larger magnitude. Then both inputs are splitted as train and test data. Here test_size=0.2 mean 80% of the dataset is used for training and 20% is used for testing.

- **Resampling the dataset** This addresses class imbalance in the training data by using oversampling and undersampling techniques. First, it uses SMOTE (Synthetic Minority Over-sampling Technique) to increase the minority class samples. Then, it applies RandomUnderSampler to reduce the majority class samples, ensuring all classes are balanced, having same number of samples.
- **Training Machine Learning models** Trained four ML models, Logistic Regression, Support Vector Machine, Random Forest and XGBoost on the resampled training data. After training, the model's performance is evaluated using performance metrics (accuracy, precision, recall, and F1score for each sentiment class) and a confusion matrix to assess its performance across the sentiment categories (negative, neutral, and positive).
- **Data Storage**: The predicted sentiment results and other attributes of tweets such as created_at(time tweet was created), location(location of tweet, text(tweet), tweet_id(tweet id), tweet_count(Count of tweets) and extracted keywords and hashtags from the tweets are stored in MySQL database.
- **Dashboard**: The dashboard is used as user-interface for the user. The MySQL database is connected to dashboard for transferring data from database to dashboard and this data is used for visualizing various trends and patterns.
- **Trend Detection**: Different trends can be visualized in dashboard such as sentiment(positive, negative and neutral), hashtags and keywords trends. These trends are detected and notified using an alert mechanism when the particular trend crosses certain threshold.

V. Evaluation

This section shows how we evaluated our ML models and tested our system.

A. Evaluation Metrics

To assess model performance, we used the following metrics:

- Accuracy: Measures the overall percentage of correctly classified instances.
- Precision: Indicates the accuracy of positive predictions for each sentiment class.
- Recall: Reflects the model's ability to correctly identify all instances of a given sentiment.
- F1 score: A balance between precision and recall, particularly important for imbalanced classes.
- Confusion matrix: Provides a detailed breakdown of true versus predicted classifications for each sentiment, helping us understand common misclassification patterns.
  - True class labels(True Positives and True Negatives): Instances that were correctly predicted are presented along the diagonal of the matrix.
  - False class labels(True Positives and True Negatives): Instances that were incorrectly predicted and that are present in off-diagonal elements.

B. Model Performance and Comparison

Table 1 show that Random Forest and XGBoost performed better among the four models. The model's can be further understood using confusion matrix as shown in Fig 2, Fig 3, Fig 4, Fig 5. The models are then further evaluated by applying cross validation technique and found that the results are consistent, so XGBoost is selected as the best performing model.

TABLE I
PERFORMANCE COMPARISON OF ML MODELS

Algorithm	Accuracy	Precision			Recall			F1-Score		
		Pos	Neg	Neu	Pos	Neg	Neu	Pos	Neg	Neu
Logistic Regression	0.93	0.78	0.83	0.80	0.91	0.92	0.91	0.99	0.96	0.98
Support Vector Machine	0.94	0.87	0.93	0.90	0.93	0.93	0.93	0.98	0.96	0.97
Random Forest	0.95	0.91	0.93	0.92	0.96	0.92	0.94	0.96	0.99	0.97
XGBoost	0.99	0.95	0.98	0.96	0.99	0.99	0.99	1.00	1.00	1.00
Weighted Avg	-	0.60	0.57	0.58	0.68	0.65	0.66	0.63	0.60	0.62

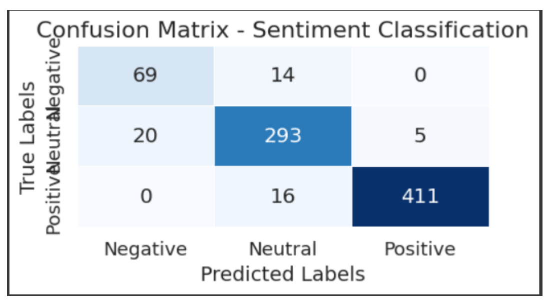

Fig. 2. Confusion Matrix of Logistic Regression

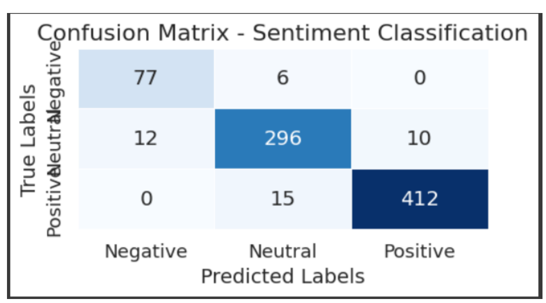

Fig. 3. Confusion Matrix of Support Vector Machine

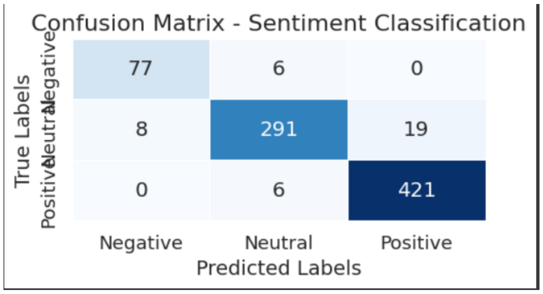

Fig. 4. Confusion Matrix of Random Forest

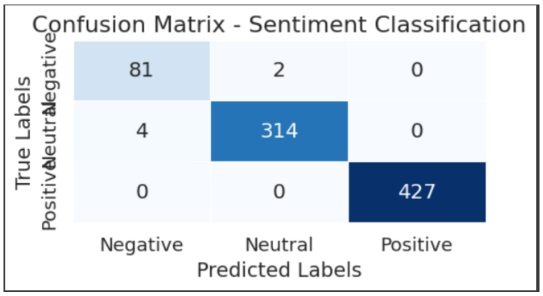

Fig. 5. Confusion Matrix of XGBoost

VI. Limitations of the System

The Twikit scraper itsef has limitations in data collection. In 15 minutes, it can collect upto 50 tweets and the rate limit resets every 15 minutes. The system does not run automatically and needs to run manually. The system does not allow one to input other hashtags and if needed to collect data related to other hashtags then one needs to modify the python code with which we have implemented data collection. In addition, the threshold for the alert mechanism is also implemented in the back-end and one cannot modify the threshold. Resolving these limitations can significantly improve the system.

VII. Results and Discussion

In this section, the insights obtained from different visuals used in dashboard shown in Fig 6, are explained.
The positive, negative and neutral sentiment percentages of the total collected tweet is shown in Fig 7. It helps in understand in overall how customers are reacting toward the brand, if are

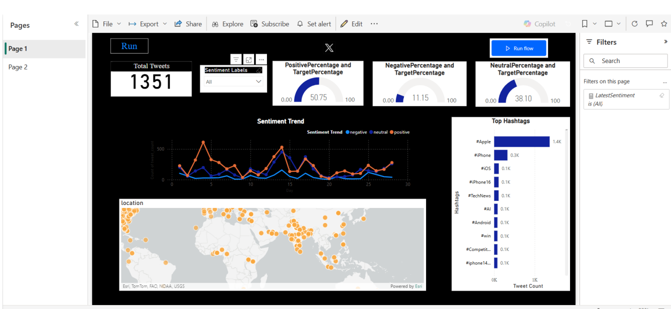

Fig. 6. Brand Monitoring Dashboard

they satisfied or dissatisfied.
The trend of different sentiment categories is shown in Fig 8. It shows how different sentiments vary throughout the timeline. The xaxis shows represents each day and y-axis represents the count of collected data. The sentiment variation also depends on how much data is collected each day, as tweets are not collected the same amount everyday.
The Top hashtags as shown in Fig 9, tell the hashtags that have been used by the users the most. The count of hashtags are presents in the decreasing order from the top. It gives an idea on how to reach maximum people using these hashtags.
Fig 10, shows the location of the users whose tweets are collected. Sometimes, it displays the name of the country or state and does not go deep into the address of the users. This location feature is provided by the dashboard.
A visual providing the keywords that are used the most by users are provided in Fig 11. The keywords that are in large size show the keyword that has been used the most by users, and the keywords that are least in size show that they are used the least by users, and the count of each keywords can be known by putting the mouse cursor on top of the keyword.

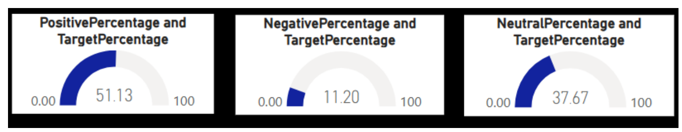

Fig. 7. Percentage of Positive, Negative and Neutral Sentiments

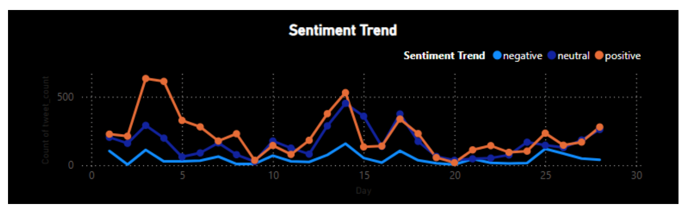

Fig. 8. Sentiment Trend

VIII. Conclusion and Future Work

This paper explored various techniques and approaches for sentiment analysis. By reviewing existing research, have identified several challenges such as noisy data, difficulty in handling sarcasm, data collection and limitations in real-time processing and also found some drawbacks in these papers. With these challenges and drawbacks, the proposed system was designed and implemented as a system that operates in real-time and collects tweets using # Apple, processes the data using advanced NLP techniques, and classifies sentiments using supervised machine learning models that address all the drawbacks.
The results showed that the XGBoost model as the best performed model and is used for predicting sentiments for tweets. Although the system performs well, there is always room for improvement. Improving data collection of this system as there are some limitations in collecting data and also

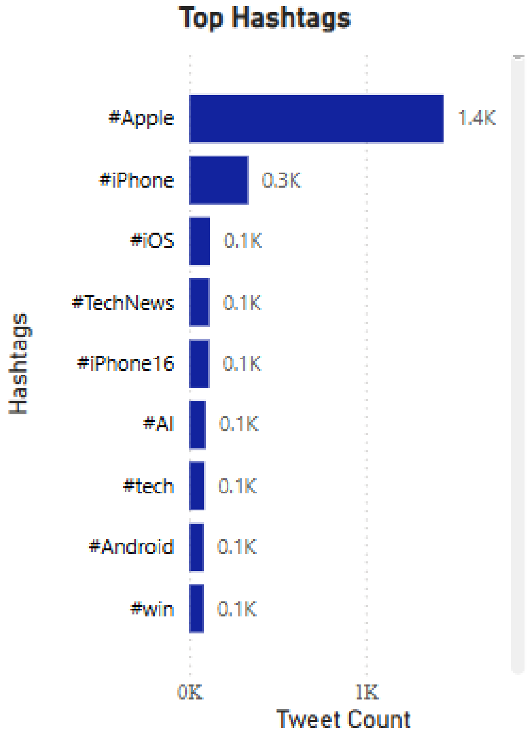

Fig. 9. Trending Hashtags

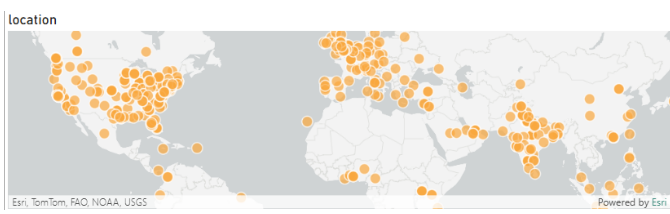

Fig. 10. Location of Tweets

Fig. 11. Word Cloud

this was preferred over Twitter API due to high cost for the Twitter API, addressing data collection from multiple sources, integration of deep learning models, expanding to multiple languages, could further improve our system. Ultimately, this paper highlights how Power Bi can significantly help improve the brand monitoring system and also the various features it provides. This paper contributes to the growing field of sentiment analysis tailored for brand monitoring.

References

[1] N. Raghunathan and K. Saravanakumar, "Challenges and issues in sentiment analysis: A comprehensive survey," *IEEE Access*, vol. 11, pp. 69 626–69 642, 2023.

[2] M. Abu Kausar, A. Soosaimanickam, and M. Nasar, "Public sentiment analysis on twitter data during covid-19 outbreak," *International Journal of Advanced Computer Science and Applications*, vol. 12, 01 2021.

[3] A. Motz, E. Ranta, A. Sierra Calderon, Q. Adam, F. Alzhouri, and D. Ebrahimi, "Live sentiment analysis using multiple machine learning and text processing algorithms," *Procedia Computer Science*, vol. 203, pp. 165–172, 08 2022.

[4] M. Rezwanul, A. Ali, and A. Rahman, "Sentiment analysis on twitter data using knn and svm," *International Journal of Advanced Computer Science and Applications*, vol. 8, 01 2017.

[5] X. He, "Sentiment classification of social media user comments using svm models," in *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, 2024, pp. 1755–1759.

[6] A. Ingole, P. Khude, S. Kittad, V. Parmar, and A. Ghotkar, "Strategic brand sentiment analysis for competitive surveillance," in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 2024, pp. 1114–1119.

[7] S. Roy, A. Roy, P. Samui, M. Gandomi, and A. Gandomi, "Hateful sentiment detection in real-time tweets: An lstm-based comparative approach," *IEEE Transactions on Computational Social Systems*, vol. PP, pp. 1–10, 01 2023.

[8] T. Wang, K. Lu, K. P. Chow, and Q. Zhu, "Covid-19 sensing: Negative sentiment analysis on social media in china via bert model," *IEEE Access*, vol. 8, pp. 138 162–138 169, 2020.

[9] A. Radaideh and F. Dweiri, "Sentiment analysis predictions in digital media content using nlp techniques," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 11, 2023. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2023.0141128

[10] A. Petrescu, C.-O. Truică, and E.-S. Apostol, "Sentiment analysis of events in social media," in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2019, pp. 143–149.

[11] A. S, "Real time twitter sentiment analysis using natural language processing," *International Journal of Engineering Research and*, vol. V9, 07 2020.

[12] J. Jabbar, I. Urooj, W. JunSheng, and N. Azeem, "Real-time sentiment analysis on e-commerce application," in *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, 2019, pp. 391–396.