Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 3920325, 13 pages
https://doi.org/10.1155/2021/3920325

Research Article

Real-Time Twitter Trend Analysis Using Big Data Analytics and Machine Learning Techniques

Anisha P. Rodrigues, Roshan Fernandes, Adarsh Bhandary, Asha C. Shenoy, Ashwanth Shetty, and M. Anisha

Department of Computer Science and Engineering, NMAM Institute of Technology, Nitte, Karkala, India

Correspondence should be addressed to Roshan Fernandes; roshan_nmamit@nitte.edu.in

Received 28 July 2021; Accepted 4 October 2021; Published 25 October 2021

Academic Editor: Rajesh Kaluri

Twitter is a popular microblogging social media, using which its users can share useful information. Keeping a track of user postings and common hashtags allows us to understand what is happening around the world and what are people's opinions on it. As such, a Twitter trend analysis analyzes Twitter data and hashtags to determine what topics are being talked about the most on Twitter. Feature extraction and trend detection can be performed using machine learning algorithms. Big data tools and techniques are needed to extract relevant information from continuous steam of data originating from Twitter. The objectives of this research work are to analyze the relative popularity of different hashtags and which field has the maximum share of voice. Along with this, the common interests of the community can also be determined. Twitter trends plan an important role in the business field, marketing, politics, sports, and entertainment activities. The proposed work implemented the Twitter trend analysis using latent Dirichlet allocation, cosine similarity, K means clustering, and Jaccard similarity techniques and compared the results with Big Data Apache SPARK tool implementation. The LDA technique for trend analysis resulted in an accuracy of 74% and Jaccard with an accuracy of 83% for static data. The results proved that the real-time tweets are analyzed comparatively faster in the Big Data Apache SPARK tool than in the normal execution environment.

1. Introduction

Twitter is a popular social networking site where millions of people tweet every second about various topics related to society, politics, sports, entertainment, and many more. The standard syntax followed by Twitter users while tweeting involves hashtags, retweets, and user mentions. Hashtags are words or phrases which are prefixed with "#," and user mention means mentioning other people, companies, brands, or precisely other Twitter users in the tweet by using the "@" symbol at the beginning of their user name. There is a restriction of 140 characters on the length of any tweet which allows users to post tweets quickly. At the same time, users all across the globe can tweet about anything happening or their thoughts at any given time of the day. Tweets thus help people to understand how others feel about different ongoing events, government policies, sports tournaments, etc. Brands can analyze tweets to know people's sentiments towards their products. Government and politicians get an idea of how people are responding to the different policies, acts, and amendments. During elections, Twitter plays a vital role in campaigning too. For a given day or a span of days, any topic can be made trending by the repeated use of the same hashtag. Thus, Twitter trends play an important role in the process of decision-making by different organizations and companies. The main motivation for the Twitter trend analysis is to identify the recent trends happening across the world using big data machine learning techniques. This will help to analyze what has happened in the past and what may happen in the future. It helps to track customer trends and interests especially what customers like, what their behaviors are, and how this changes over the time.

In the proposed work, the tweets are collected using Twitter API and applied counting methods and different machine learning algorithms to identify trending topics on

Twitter. Twitter API provides a standard way to read and write Twitter data. This API provides a set of methods that can be used to communicate with the application. To process a huge volume of tweets instantaneously, we have used SPARK streaming. SPARK is a big data tool that can be effectively used to deal with a large volume of data in a short time. Hashtag counting and noun counting are the two basic methods that count the hashtags and nouns in tweets, respectively, to determine which particular word is trending. Topic modeling technique latent Dirichlet allocation (LDA) is used, which groups the tweets into clusters of topics based on keywords. Cosine similarity measures how similar two or more documents are and groups the tweets accordingly. K means clustering and Jaccard similarity also help us to classify tweets into clusters. By using SPARK streaming, we were able to identify real-time trends more quickly as compared to a normal execution environment. We have performed an analysis of the time taken to execute the programs on static data and real-time data collected using SPARK. We have also included analysis for May 2021 which shows us the output obtained using different techniques and helps us to conclude that all algorithms run efficiently and give accurate trends.

1.1. Contributions of the Proposed Work. By carefully analyzing many works of literature in the field of Twitter data analysis, we have concluded that the majority of researchers have contributed towards Twitter sentiment analysis than trend analysis. Few researchers who contributed to trend analysis have used LDA and clustering techniques using SPARK. The main contribution of the proposed work is to perform the Twitter trend analysis. This includes applying the various techniques for Twitter trend analysis and comparing the results using various evaluation parameters. The techniques used are hashtag counting, noun counting, cosine similarity, Jaccard similarity, LDA, and K-means. These techniques are applied to static Twitter data as well as real-time streaming data and compared the results. We obtained better results in terms of execution speed for real-time Twitter trend analysis using SPARK.

2. Related Work

To identify sentiments in tweets, lexicon-based methods and polarity multiplication have been used [1]. NLP techniques like tokenization, removal of stopwords, and stemming are used for preprocessing. The lexicon method is simpler and has lower accuracy compared to machine learning. Hence, machine learning techniques must be used for analyzing tweet sentiments and trends. Machine learning algorithms like Naïve Bayes, SVM, and KNN were used for sentiment analysis [2, 3]. Out of the three Naïve Bayes was found to achieve the highest accuracy, i.e., 80.9% followed by KNN with an accuracy of 75.58%. Latent Dirichlet allocation which is a topic modeling algorithm was used to analyze tweets and extract useful information from them [4]. Using LDA, a large number of tweets are processed as a collection of documents, where each document is associated with a collection of topics. Each topic is associated with a set of words, and each document has a different proportion of topics based on the frequency of words that appear in each topic. The same method was used by Negara et.al [5] to process a large number of tweets and divide them into 4 clusters, namely, economic, sports, military, and technology. LDA algorithm was found to have optimal performance for Sports tweets with an accuracy of 98% which is better than LSI topic modeling.

Shahreen et al. [6] have used the machine learning and neural network approach for the text analysis. SVM was used for text analysis, and weight optimizers like Limited-memory BFGS, Stochastic gradient descent, and Adam were used to attaining maximum accuracy using neural networks. They obtained an accuracy of 95.2% with SVM and 97.6% using a neural network. Hidayatullah et al. [7] performed topic modeling on a dataset obtained from the official Twitter account of traffic management center in Java to create a topic model regarding traffic information. Hasan et al. [8] have planned the analysis in two phases. In phase 1 after data acquisition, researchers have preprocessed the data stream using tokenization and stop word removal; then, they have clustered using improved fuzzy C-means clustering and adaptive particle swarm optimization. They have examined Twitter data streaming using an Apache SPARK engine. In phase 2, the data is preprocessed, and they have classified Higgs data using modified SVM, and Higgs data streaming is examined using an Apache SPARK engine. The computational analysis shows that it achieved better results compared to existing methods in terms of F-score, precision, ROC curve, and accuracy. Garg and Kaur [9] have explained the analysis of Twitter data using components of Cloudera distribution of Hadoop. The objective is to assign polarity to each tweet. Map reduce and Apache SPARK frameworks were used for sentiment analysis. The result shows that Apache SPARK is better than MapReduce. Saad and Yang [10] have performed sentiment analysis of Twitter data using ordinal regression. The preprocessed tweets are run using different machine learning algorithms. These algorithms reveal the polarity of tweets. The algorithms used were support vector regression, decision tree, random forest, and multinomial logistic regression of which decision tree showed the highest accuracy.

Hasan et al. [11] used machine learning techniques to perform sentiment analysis. Polarity calculation and sentiment analysis were performed using Text Blob, Sentiwordnet, and W-WSD and then classified using Naive Bias and SVM. It gives a comparison of techniques of sentiment analysis by applying supervised machine learning algorithms like Naïve Bayes and SVM. Huq et al. [12] have also performed sentiment analysis on Twitter data using machine learning algorithms. They have used SVM (support vector machine), and sentiment classification algorithm (SCA) was built using KNN (K Nearest Neighbour). The performance of both was compared, and SCA is found to be better than SVM. Jianqiang et al. [13] found that convolutional neural networks are better for sentiment classification of tweets. An RBF kernel SVM and LR exploiting unigram and bigram features (BoW) were also used. For twitter sentiment analysis, DCNN using pretrained word vectors was found to have good performance. Ahmed and Rodríguez-Díaz [14] have performed sentiment analysis on online customer reviews. Here, text selection, text collection, text processing, sentiment analysis, and regression analysis techniques were used. This project analyzes the customer experience and helps to meet customer demands. Predeveloped lexicons were used to determine positive and negative signs as there is no dynamic element to guide feelings. Rathod and Barot [15] researched the same field to predict public opinion on ongoing events by analyzing tweet sentiments using machine learning classifiers like SVM, Naïve Bayes, logistic classifier, and KNN classifier. SVM was found to be the best classifier with the least mean square error for the classifications. Garg et al. [16] have identified the trending pattern in Twitter using SPARK. These patterns were obtained by collecting tweets on a real-time basis and identifying trending hashtags at the same time. It was implemented using a big data technology SPARK streaming. This helps companies to know about their brand awareness and customer needs. To handle a large number of tweets from Twitter on a real-time basis, SPARK framework has been used. Sentiment analysis and opinion mining of tweets have been done using the same [17]. Machine learning techniques can be extended to classify the fake reviews and fake news [18, 19]. The text classification is improved using the two-stage text feature selection algorithm [20]. Big data Hadoop framework is used to classify the product reviews based on aspects [21].

This research article proposed Twitter trend analysis using hashtag counting, noun counting, cosine similarity, Jaccard similarity, LDA, and K-means. These techniques are applied to static Twitter data as well as real-time streaming data and compared the results. The proposed work obtained better results in terms of execution speed for real-time Twitter trend analysis using SPARK tool.

The rest of the content is organized as follows. Section 3 discusses the proposed methodology. Section 4 gives the detailed results and analysis, and Section 5 highlights the conclusion and future scope in this research work.

3. Proposed Methodology

The proposed methodology includes the various steps, namely, collecting the static and real-time tweets from the Twitter and to perform the trend analysis. The proposed technique uses both static tweets and also real-time tweet trend analysis. Initially, the tweets need to be preprocessed for further analysis. Later, various machine learning techniques are applied on these static and real-time tweets to analyze the trends. Figure 1 depicts the proposed architecture for the real-time Twitter trend analysis.

This model is aimed at analyzing the trending topics in Twitter by using different approaches. Initially, the tweets are collected and preprocessed for further analysis. This preprocessed data is then analyzed using various methods like hashtag counting, noun counting, cosine similarity, Jaccard similarity, LDA, and K-means techniques. The performance of each algorithm is evaluated. The results are then analyzed to obtain the trending topic. We have also used SPARK framework to analyze real-time tweets. Using real-time streaming by SPARK, we have streamed tweets in real-time from Twitter and produced trending results faster. The various components involved in the proposed work are discussed in this section.

3.1. Data Collection. Tweets are collected using Twitter API. Tweets belonging to different domains like sports, health, economy, politics, and social, which were tweeted between January 15, 2021, and June 30, 2021, are collected. We have collected as many as 20,000 tweets. The dataset follows the JSON format. While streaming data through SPARK, we used a TCP socket as a data source to which tweets were written. SPARK will read and process the data from the socket.

3.2. Data Preprocessing. The collected tweets were stored locally in JSON file. This data is preprocessed by the following steps:

(1) Converting emoticons present in the tweets to text

(2) Removing hyperlinks (https/url) present in each tweet

(3) The tweets are made ready to be processed by removing punctuations and white spaces

(4) Removing stop-words

(5) *Performing Stemming.* Stemming is the process of removing suffixes in a word and retaining only the root word. For example, eating will become eat after stemming

(6) *Performing Lemmatization.* Lemmatization is similar to stemming where the output after the lemmatization process is called "lemma." In lemmatization, the reduced form of the words is found to be more meaningful when compared to the results of stemming

The preprocessed data is then fed as input to various algorithms to track the trending topics. The various techniques used were hashtag counting, noun counting, cosine similarity, Jaccard similarity, LDA, and K-means clustering.

3.3. Hashtag Counting. Hashtag counting is a primitive and simple method of predicting a trending topic. The collected dataset is subjected to counting, based on the number of times the hashtag appears in the dataset its trend value is set. Then, this value is used to get the top trends corresponding to the processed dataset. Here, the hashtag with the highest count is said to be a trending hashtag.

3.4. Noun Counting. In this method, the tweet contents are tagged with corresponding parts of speech. Tweet contents are categorized as nouns, verbs, adverbs, adjectives, and so on. Now, we detect the trend by counting the repeated nouns. The noun with the highest count is said to be trending.

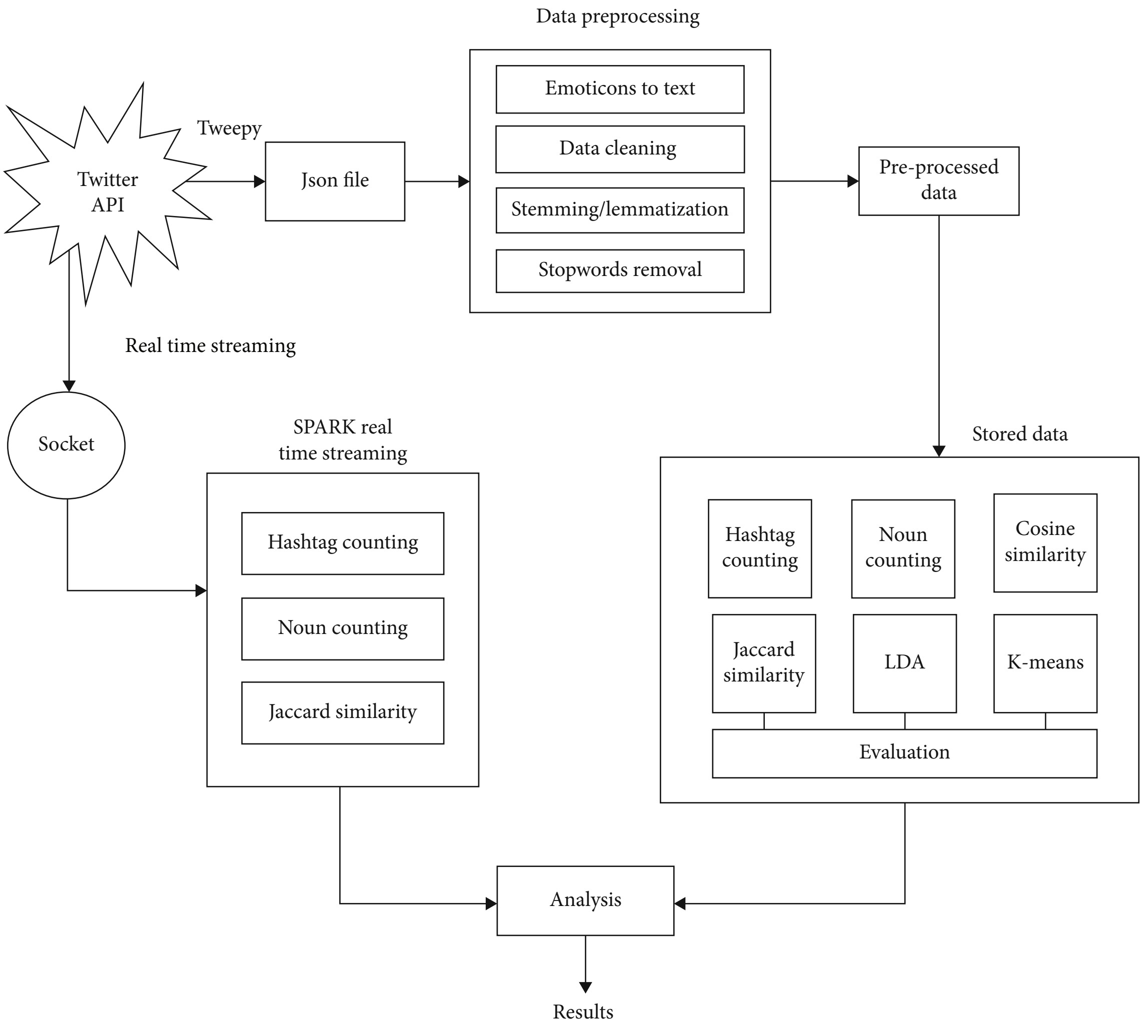

Figure 1: Proposed architecture for real-time tweet trend analysis.

3.5. Clustering Using Latent Dirichlet Allocation (LDA). Topic modeling techniques can be used to analyze Twitter trends based on the tweet text. The goal of this type of analysis is to find the different hidden topics in the dataset of tweets and then to determine the trending topic based on the number of tweets for each topic. LDA is one of the topic modeling algorithms specially designed for text data. This technique considers each document as a mixture of some of the topics that the algorithm produces as a final result. The topics are the probability distribution of the words that occur in the set of all the documents present in the dataset. For the Twitter trend analysis, the dataset can be considered as the set of documents where each document will be a tweet.

For example, consider the following three tweet texts:

(1) "What a champion. Simply the best. So calm, so sure in a run-chase. No big celebration, no theatrics, just a job finished. Superstar of the game."

(2) "IPL is postponed because of COVID. It is sad but safety is first."

(3) "Day 486 of lockdown, no effective vaccine rollout, restaurants are only doing takeaways and honestly this is all taking away my happiness."

The preprocessing of these tweet texts will give keywords as follows: ['champion', 'best', 'job', 'run-chase, 'celebration', 'superstar', 'game'], ['IPL', 'postpone', 'COVID', 'safety', 'first'] and, ['lockdown', 'vaccine', 'restaurant', 'takeaway', honest', 'happy'].

Each keyword array will be considered as a document, and LDA will try to find the hidden topics based on the probability distribution of keywords. We observe that the above tweet texts are related to sports and the COVID-19 pandemic. Initially, the algorithm will assign each word in the document to a random topic out of *n* number of topics. As we already know theoretically, the above tweets consist of two topics; the algorithm may assign the first word that is "champion" for topic 2 (COVID-19). We know this assignment is wrong, but the algorithm will try to correct this in the future iteration based on two factors that are how often the topic occurs in the document and how often the word occurs in the topic. As there are not many COVID-19-

related terms in tweet 1 and the word "champion" will not occur many times in topic 2 (COVID-19), so the algorithm may assign the word "champion" to the new topic that is topic 1 (sports). With multiple such iterations, the algorithm will achieve stability in topic recognition and word distribution across the topics. Finally, each document can be represented as a mixture of determined topics; in the example, under consideration, tweet 1 is 100% topic 1, tweet 2 is 70% topic 1 and 30% topic 2, and tweet 3 is 100% topic 2. The number of topics and other tuning parameters can be altered to get better results in terms of clear topics.

3.6. Trend Analysis Using Cosine Similarity. Cosine similarity is a standard of measurement used to determine how much similar the records are regardless of their size. In terms of mathematics, it is a measurement of the cosine of the angle between two vectors plotted in multidimensional space. In this context, the two vectors are dictionaries (with the key being word and value being the count of that particular word) of those two documents. When we plot these two vectors in a multidimensional space, where each dimension corresponds to the keys in the dictionary (i.e., words in the document) and corresponding values represent how far is the point from that dimension, the cosine similarity calculates the angle between those two vectors, not the Euclidean distance. The cosine similarity metric is beneficial because even when two documents with the similar resemblance in word count but are far apart by the Euclidean distance because of the size (e.g., the word "baseball" occurred 100 times in the first document and 10 times in the second document), but they could have had a minor angle between them. The lesser the angle, the greater the similarity as we know the cosine of the angle increases as the angle decreases. Given two vectors $\vec{a}$ and $\vec{b}$, the angle between those two vectors is calculated by the equation (1) [22]:

$$\theta = \cos^{-1}\left(\left(\vec{a}\ .\ \vec{b}\right)/\left(\left|\left|\vec{a}\right|\right| * \left|\left|\vec{b}\right|\right|\right)\right) \cdots, \tag{1}$$

where

$$\vec{a}\ .\ \vec{b}\ = \sum_{1}^{n}(ai * bi). \tag{2}$$

On Twitter, hashtags are not mandatory for any tweet, so some tweets may not be having the hashtag attached. If we directly apply the hashtag counting algorithm for analysis on such data, the algorithm will simply ignore the tweets without hashtags thereby making the analysis inaccurate. Hence, in the proposed work, we divided the dataset into two sections as tweets with hashtags and tweets without any hashtags. Each tweet in the first section is stored as a document and labeled with the respective hashtag thereby creating a document for each available hashtag. Now for each tweet in the second section, we try to introduce the missing hashtag as one among the many available options in the stored document set based on the cosine similarity between the tweet under consideration and documents. This way we can make sure that each tweet will be attached with relatable hashtags and thereby considered by the hashtag counting technique.

3.7. Trend Analysis Using Jaccard Similarity. Jaccard similarity can be used to get the similarity coefficient of the tweet text and the predefined clusters and then can be classified based on the score obtained. Jaccard similarity algorithm works using the set intersection and union operations as shown in equation (3) [22].

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \cdots. \tag{3}$$

Consider a tweet text "Arranging ambulance, oxygen, and beds at the hospital during this COVID19 pandemic was not at all easy." Now after preprocessing the text, we get "Arranging ambulance oxygen bed hospital pandemic easy" as keywords. If we have two predefined clusters, namely, health and sports as "hospital ambulance doctors medicine COVID19 vaccine bed cough lungs pandemic oxygen pandemic" and "cricket match championship ipl football pandemic suspended umpire loss win," respectively, then we can represent all these as follows.

$$\begin{aligned} \text{Tweet}_{\text{text}} &= [0, 1, 2, 3, 4, 5, 6], \\ \text{Health}_{\text{related}_{\text{words}}} &= [4, 1, 7, 8, 9, 10, 3, 11, 12, 2, 5], \\ \text{Entertainment}_{\text{related}_{\text{words}}} &= [13, 14, 15, 16, 17, 5, 18, 19, 20, 21], \end{aligned} \tag{4}$$

where {'Arranging': 0, 'ambulance': 1, 'oxygen': 2, 'bed': 3, 'hospital': 4, 'pandemic': 5, 'easy': 6, 'doctors': 7, 'medicine': 8, 'COVID19': 9, 'vaccine': 10, 'cough': 11, 'lungs': 12, 'cricket': 13, 'match': 14, 'championship': 15, 'ipl': 16, 'football': 17, 'suspended': 18, 'umpire': 19, 'loss': 20, 'win': 21}.

For the given text intersection with health, *n* sports will be {1, 2, 3, 4, 5} and {5}, respectively. Hence, the Jaccard coefficient for health and sports will be 5/18 and 1/17. As the score of similarity for the health cluster is higher, the tweet will be classified as health related to tweet.

3.8. Trend Analysis Using K-Means Clustering. Interests of Twitter users vary from user to user; some may tweet more about social events, some are much into politics, and some tweet more about sports. Twitter users' behavior or interests and in turn likes and dislikes can be analyzed based on the number of tweets they tweet on different various topics. By using the results of Jaccard similarity, we can cluster Twitter users into multiple categories with the help of K-means clustering. K-means algorithm attempts to cluster the given dataset into *k* number of nonoverlapping groups, such that every data point in the dataset belongs to a unique cluster. Cluster formation is done such that maximum similarity is maintained within a cluster, and different clusters are as far as possible from each other. Euclidian distance is used to achieve the clustering goal. For example, if we consider some tweets for four Twitter accounts related to the health and sports category as [1], [1, 2], [3, 4], and [4, 5], respectively, where [a,b] represents *a* number of health-related tweets and *b* number of sports-related tweets from a user. K-means algorithm follows below simple steps in a loop until it meets converging conditions.

(1) Find the coordinates of the centroid

(2) Calculate the distance of each object to the centroid

(3) Assign the objects to a cluster based on minimum distance

3.9. Real-Time Streaming Using SPARK. A good analysis needs a large amount of data. The more is the data, the much better will be the analysis. It is very important to cover a large volume of tweets across the globe on various topics from different people to get accurate trends while analyzing Twitter trends. Thankfully, Twitter provides all the support we need to get tweets for such analysis. But if we choose to write a program for collecting the tweets and then preprocess, store, and finally apply algorithms on the stored data to find out the trends, a lot of time and resources will be wasted when we can do the same task with the help of real-time streaming and SPARK. Thanks to Twitter again, which will support streaming the Twitter data. A TCP socket in a system will be used instead of a file to hold the incoming Twitter stream. If a SPARK session is connected to this same TCP socket, it will read incoming data as soon as it will be written to the socket. This powerful combination can be used to enhance the results of the algorithms that have mentioned earlier in the paper. The advantage here is that the model will not wait until we are done collecting the required amount of tweets. Every time Twitter data is written to the socket, SPARK will immediately start processing it. With the structured streaming support in SPARK, the result will get updated as the incoming data get processed. So even though SPARK produces results in batches, every batch will be having the result corresponding to the data streamed until that point in time.

In our analysis, we have used a TCP socket as a data source. Tweets are collected on real-time basis using Twitter API and tweepy writes it to this data source. Pspark processes tweets in batches. The program runs on SPARK for the specified time interval where in tweets are streamed in batches and output is also obtained in batches. Depending on what is being tweeted about the most at a given time, the numbers will keep changing in every batch. In this way, we were able to stream data continuously on a real-time basis. Compared to static data, larger number of tweets could be collected, and it also provides accurate trends at any given time of a day.

4. Results and Discussion

The experimental results of different techniques and algorithms used for trend analysis provide us insights on which method is best suited for real-time analysis and gives accurate results. The outputs of these techniques have been presented and analyzed in the form of graphs, and tables and a close match have been found between the different results obtained. We have performed trend analysis on static and dynamic data. For static analysis, data is collected beforehand and stored in a file. Basic counting methods and machine learning algorithms are applied to this stored data to identify the trends. In the case of dynamic analysis, data is streamed on a real-time basis, and analysis is performed at the same time by using SPARK structured streaming. This has allowed us to process a large number of tweets and obtain accurate trends. For the experimental analysis, we have used a sample dataset having 20000 static tweets. For real-time trend analysis, we have extracted live tweets.

4.1. Static Twitter Data Analysis. The first and the basic method we have used to predict what is being talked about a lot on Twitter is counting the hashtags. Hashtags being the key elements of tweets are used widely by people to express their opinions and as supporting elements of the tweet content. For the experimental analysis, we have used a sample dataset having 20000 tweets where in "#COVID19" has been found to be used the highest number of times, which is 87.

Hashtag counting does not consider the actual tweet content to predict the trend. To overcome this drawback, we have used the noun counting method which identifies the nouns in all the tweets and hence tells us which nouns have been used repeatedly. At first, we have used part-of-speech tagging to tag the words in the tweets with their corresponding part of speech. Next, the words tagged as "nouns" were collected and counted to find the noun that has been used frequently.

As seen in the result, the word "vaccine" is found to be used the highest number of times. The results obtained using the two counting techniques are related and strongly comply with the actual scenario too. Hence, we decided to use these two techniques for real-time analysis as well. Results of Twitter trends using hashtag counting and noun counting are shown in Figures 2 and 3, respectively.

A sample dataset with 6000 tweets has been used to get the results of the designed LDA model. The preprocessed dataset stored in a data frame was given as input to the model. The execution of the algorithm for a different number of topics (k) produced different results. To find an optimal number of topics for the given dataset, the coherence value has been calculated for each value of k. This measure helps us to figure out how coherent the topics are, in other words how well the recognized topics support each other. Figure 4 shows the coherence value for different k values.

Choosing the right k value is not straightforward always, and there is no such standard way to do that. Either we can manually try to tune the k value based on the topic interpretation or we can consider the k with a larger coherence value. In the above sample, $k = 3$ can be taken as an optimal number of topics and further can be improvised by modifying the other parameters such as alpha, beta, or even number of iterations. The following is the result of LDA after tuning k values.

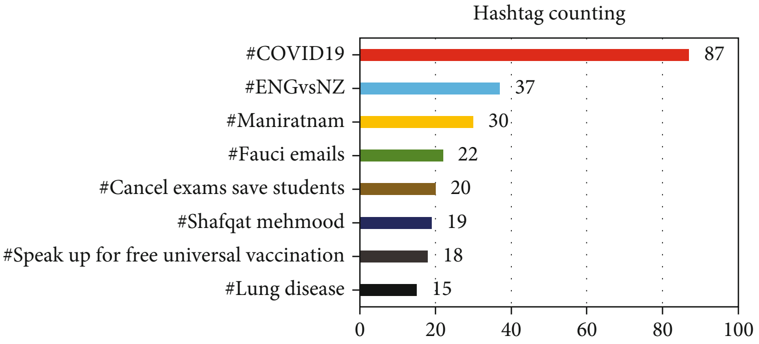

Figure 2: Twitter trends using hashtag counting.

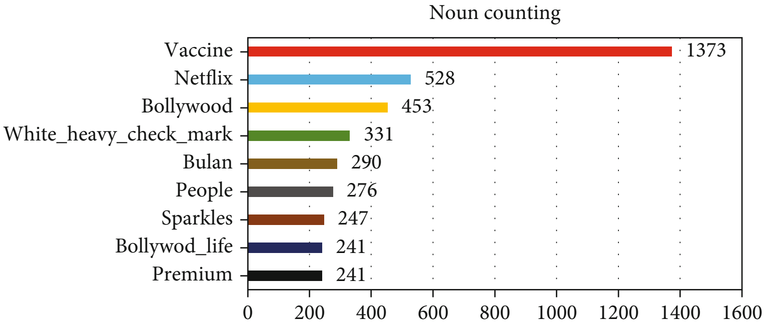

Figure 3: Twitter trends using noun counting.

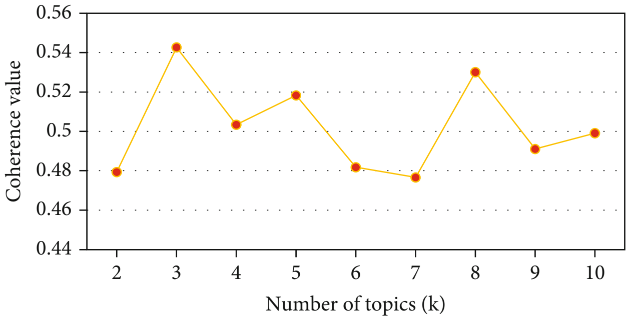

Figure 4: Number of topics vs. coherence value.

Table 1: Topic assignment for the tweets in the data set.

Tweet number	Topic distribution
tweet2186	99% Bollywood
tweet1782	98% Netflix
tweet2640	99% Bollywood
tweet5518	99% COVID-19
tweet1718	26% Bollywood, 73% Netflix
tweet5725	99% COVID-19
tweet1051	99% Bollywood
tweet936	99% COVID-19
tweet2522	99% Bollywood
tweet2679	77% COVID-19, 22% Netflix

[(0,
'0.056∗"vaccine" +0.020∗"month" +0.016∗"covid" +0.010∗"netflix" +0.009∗"hold" +0.008∗"day" +0.008∗"solo" +0.007∗"php" +0.007∗"amp" +0.006∗"people"'),
(1,
'0.048∗"bollywood" +0.013∗"movie" +0.009∗"netflix" +0.006∗"amp" +0.006∗"film" +0.005∗"time" +0.005∗"actor" +0.005∗"good" +0.004∗"song" +0.003∗"sushant"'),
(2,
'0.039∗"netflix" +0.014∗"bulan" +0.012∗"premium" +0.012∗"jual" +0.010∗"spotify" +0.009∗"viu" +0.008∗"canva" +0.008∗"legal" +0.008∗"youtube" +0.007∗"garansi"')]

The output contains 3 topics with id topics 0, 1, and 2, a close look at these clustered topics can give some insights on what that topic represents. In the above case, we can say that topic 0 is the COVID-19 pandemic, topic 1 is Bollywood, and topic 2 is Netflix. Table 1 gives the topic assignment for each tweet in the dataset.

If we approximate the topic distribution by assigning the most probable topic in the distribution for each tweet, we get 2307 tweets related to COVID-19, 2100 tweets about Bollywood, and 1593 tweets on topic Netflix. Figure 5 shows the analysis.

Next, we used cosine similarity to group the tweets into documents based on different topics and then measured the cosine of the angle between the documents by considering them as vectors. First, it creates documents that have hashtags and then label that particular document with that hashtag. For the tweets which have no hashtag, it will

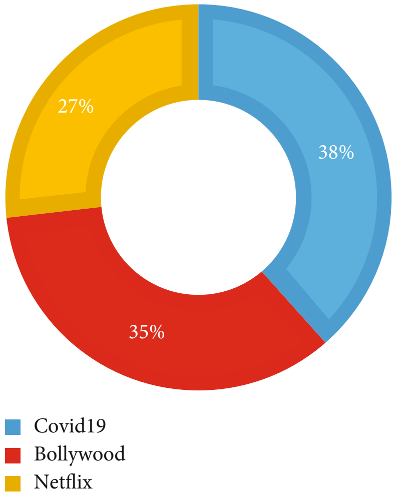

Figure 5: Trend analysis using LDA.

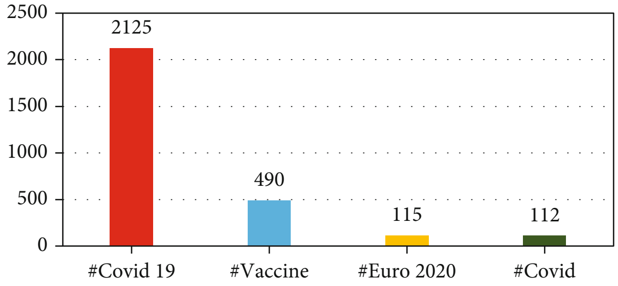

Figure 6: Top trends using cosine similarity.

calculate cosine similarity with TF-IDF to all the documents that were created earlier, and one with the highest similarity is labeled with the hashtag, and the count of that hashtag is also incremented. Here, #covid19 is trending in the generated documentation for the dataset that we have collected. This is depicted in Figure 6. When compared with Figure 2, the following graph in Figure 6 shows the huge rise in the hashtag counts after the usage of cosine similarity to generate hashtags for the tweets without any tags attached. The accuracy of this model in terms of introducing relatable hashtags at the missing value is calculated to 0.7397 which is approximately 74%.

The collected Twitter data has been classified by the model, designed using the Jaccard similarity classification algorithm. It shows that health-related tweets are more in number when compared to other categories with 60% of the collected tweets being health tweets. This is depicted in Figure 7.

The performance of the model was determined by the accuracy and Jaccard score. To find the accuracy, a dataset is prepared with tweets including their actual category which is assigned manually. Later, tweets in the dataset were given as input to the model, and the predictions done by the model are compared against the actual results stored in the dataset.

The confusion matrix for the above results is shown in Table 2.

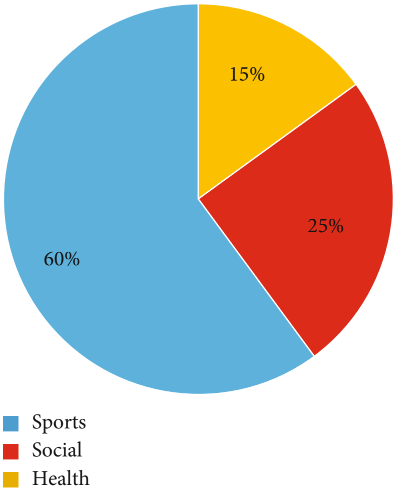

Figure 7: Trend analysis using Jaccard similarity.

Table 2: Confusion matrix for Jaccard similarity.

Expected\predicted	Social	Health	Sports
Social	112	20	33
Health	16	283	6
Sports	25	21	153

Based on the above results, the model accuracy is calculated to 0.8316 approximately 83%. Similarly, the Jaccard score will be 0.7117(average = "micro") and 0.68551 (average = "macro").

4.2. Trend Analysis Using K-Means Clustering. The model has been designed in such a way that it can group users into various categories based on the results of Jaccard similarity. For the sample data set, four categories were predefined, namely, economy, health, social, and culture. At first, each tweet in the dataset was classified using Jaccard similarity, and the number of tweets tweeted by each user id was calculated.

If we cluster users based on their interest in the social and economic sectors, we get the result that shows that a group of users show limited involvement in the social sector with less than 100 tweets and not much interest in the economic sector either. There is one more group of people who tweets on both sectors but show high interest in the social sector when compared to the economy. This is shown in Figure 8.

Silhouette outline can be adapted to fix the degree of separation among clusters. The optimal number of clusters will be decided based on the silhouette coefficient.

$$\text{Silhouette coefficient} = \frac{b^i - a^i}{\max\left(b^i, a^i\right)} \cdots, \tag{5}$$

where a^i is the average length from all score points in the corresponding cluster and b^i is the average length from all

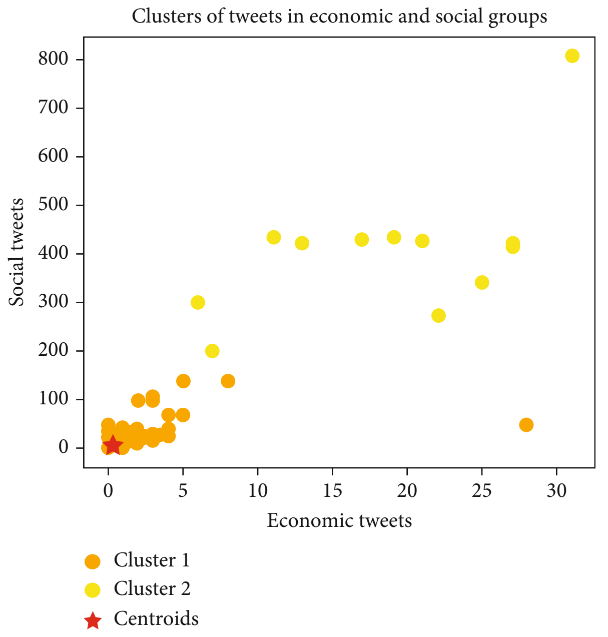

FIGURE 8: Results of clustering after applying on economic and social tweets.

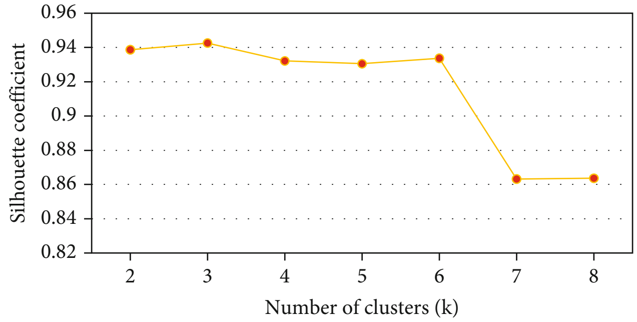

FIGURE 9: Number of clusters vs. silhouette coefficient.

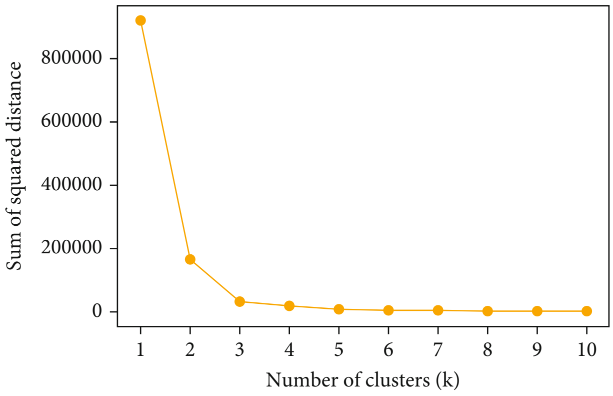

FIGURE 10: Finding the best value of k using the elbow technique.

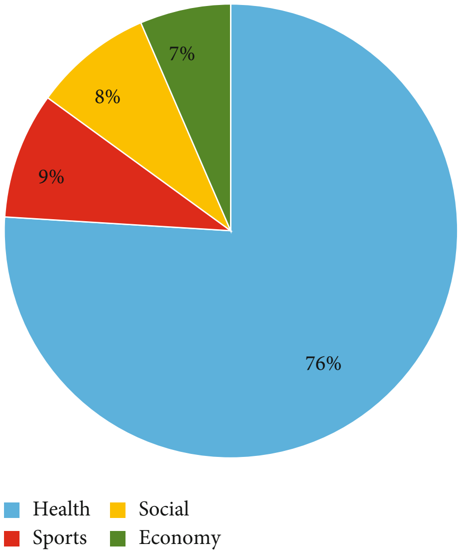

FIGURE 11: Real-time tweet trend distribution using Jaccard similarity.

score points in the nearest cluster. The coefficient can get values within the interval [-1, 1]. If it is 0, the unit is very close to the nearby clusters. If it is 1, the unit is notably apart from the nearby clusters. If it is -1, the unit is attached to the incorrect clusters. Hence, we look for the k value with a higher coefficient value. For the dataset under consideration, we can say that $k = 2$ or $k = 3$ is not a bad choice which has a silhouette coefficient of 0.938 and 0.942, respectively. But for $k = 7$ or $k = 8$, we can observe that there is a decrease in the coefficient value. This is depicted in Figure 9.

An alternative method for Silhouette analysis is the elbow method. The elbow method helps in deciding a good match of k value for a given dataset as per the sum of squared distance between data points and corresponding clusters' centroids. The optimal k value is the spot where SSE forms an elbow and starts to flatten out. This is shown in Figure 10.

The above graph shows the formation of the elbow at $k = 2$ and $k = 3$. $K = 2$ is chosen as the optimum number for clusters of the given dataset as the SSE curve forms elbow at that point.

4.3. Real-Time Twitter Data Analysis Using SPARK.

To process a large number of tweets in a fast manner, we have used SPARK streaming. SPARK is a big data tool that enables fault tolerance parallelism in the data processing. The results obtained using SPARK were rightly matching with the real-time Twitter trends. We have applied hashtag counting to find popular hashtags, noun counting to obtain the most prominent words in the tweets, and Jaccard similarity to group the tweets into different categories like health, economy, sports, and social. We collected tweets for one month (May 2021) to analyze and compare the output of different techniques. By using Jaccard similarity, we were able to group the tweets into different categories and obtain a pie chart for the distribution shown in Figure 11.

Figures 12–14 depict the real-time sports, health, and social trends using the hashtag counting technique. Figures 15–17 depict the real-time sports, health, and social trends using the noun counting technique.

The above graph shows that for each separate category obtained in Jaccard similarity, similar words are found to be trending when both hashtag counting and noun counting

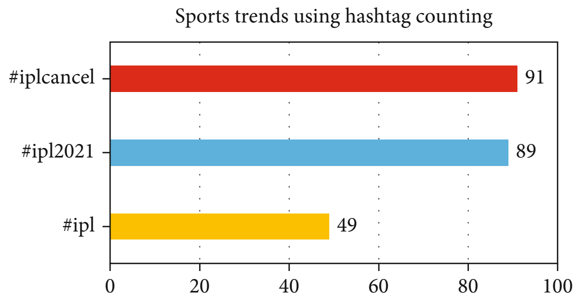

Figure 12: Real-time sports trends using hashtag counting.

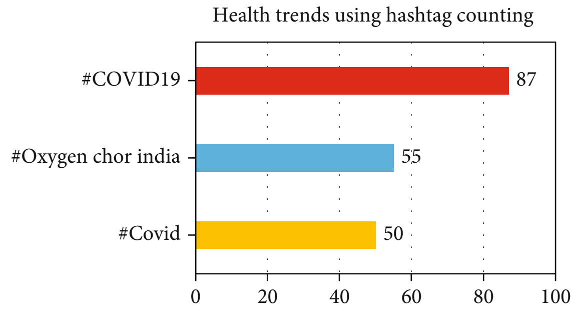

Figure 13: Real-time health trends using hashtag counting.

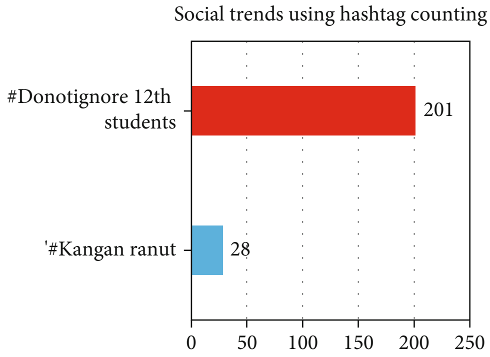

Figure 14: Real-time social trends using hashtag counting.

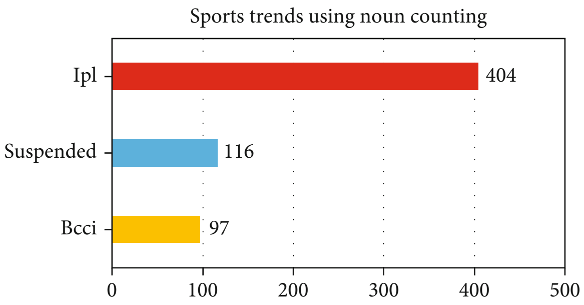

Figure 15: Real-time sports trends using noun counting.

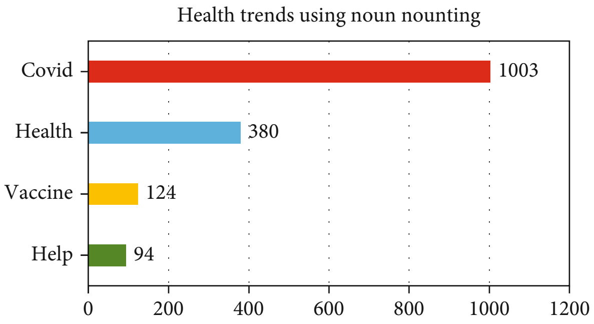

Figure 16: Real-time health trends using noun counting.

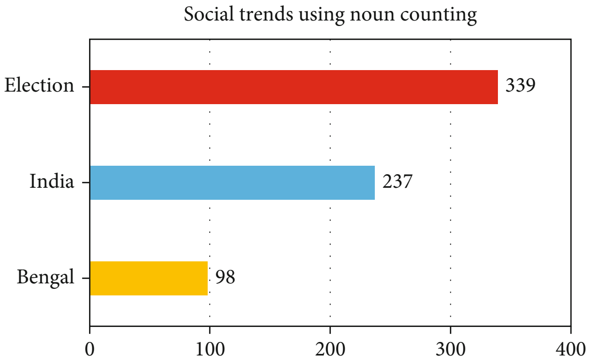

Figure 17: Real-time social trends using noun counting.

methods are applied. We can track the counts for certain handpicked nouns using noun counting technique for a particular interval of time. Figure 18 depicts the trend plot for few handpicked nouns based on the Twitter activities for 5 days of interval in June of 2021.

Similarly, we can even track the trends for few selected hashtags as well using hashtag counting techniques. Figure 19 depicts the trend chart of some of the popular hashtags for 3 days period in June of 2021.

Figure 20 shows the variation in the volume of real-time tweets related to health, economy, and social for five days session in June of 2021 using the Jaccard similarity method. Table 3 gives the comparison of Twitter trend analysis using SPARK and without using SPARK in terms of execution time required for the hashtag counting method (in seconds).

Figure 21 shows the execution time comparison between two cases with and without using SPARK for real-time and stored tweet trend analysis using hashtag counting technique, respectively.

In the graph shown in Figure 21, we can see that for a smaller number of tweets, SPARK is taking relatively higher time but as the number of tweets increases, we can see the difference and need for using SPARK. With real-time streaming and using SPARK as we can generate the results in batches, the response time will be much better compared to the program without using stream and SPARK. Table 4 gives the response time in seconds while applying different analysis techniques on the stored datasets without using SPARK tool.

Figure 22 depicts the comparative execution time in seconds, for the real-time trend analysis by streaming with

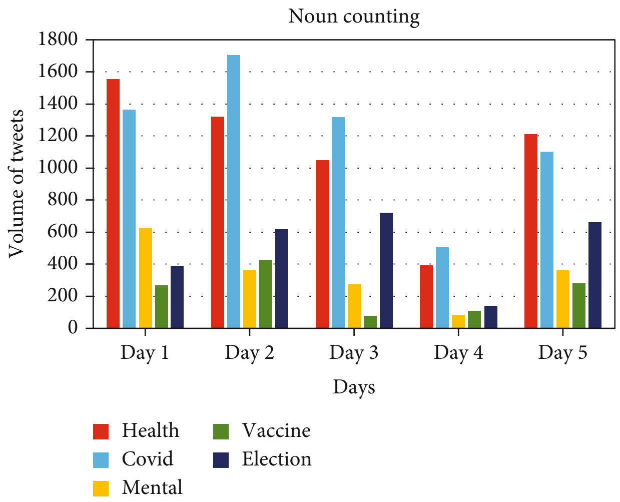

FIGURE 18: Real-time Twitter trend analysis for 5 days in June using noun counting.

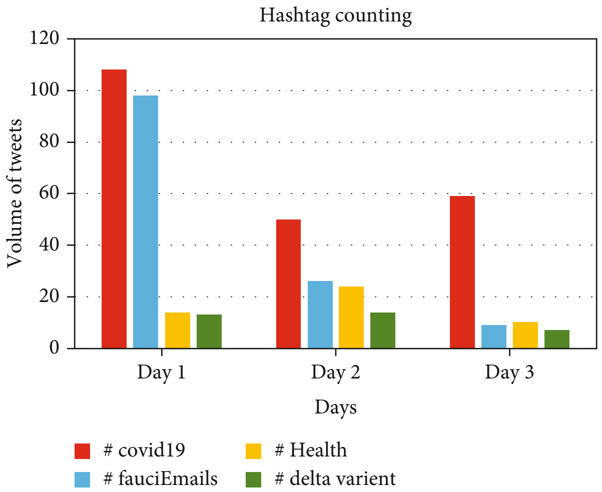

FIGURE 19: Real-time Twitter trend analysis for 3 days in June using hashtag counting.

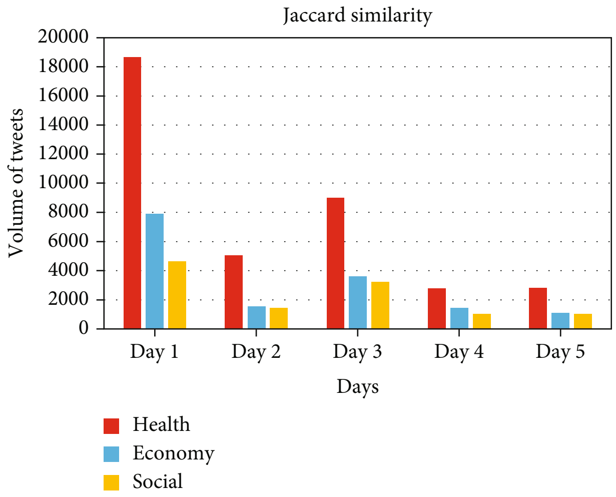

FIGURE 20: Real-time Twitter trend analysis for 5 days using Jaccard similarity.

TABLE 3: Comparison of execution time (in seconds) with and without using SPARK for hashtag counting technique.

Number of tweets processed	Total execution time using SPARK	Total execution time without using SPARK
457	51.49812531	28.36422133
1075	101.5530577	66.67746949
1509	151.5708518	432.3796902
1955	201.9629259	651.3878441
2316	231.7777178	806.4659975
2615	271.9080777	817.5746803
4814	302.536587	1656.805031

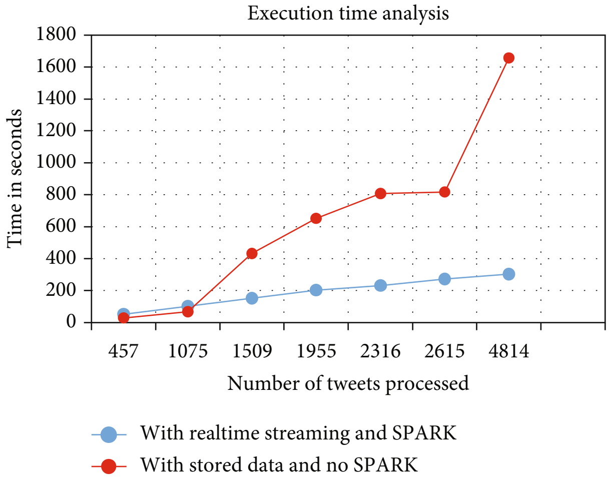

FIGURE 21: Execution time analysis.

TABLE 4: Response time (in seconds) without using SPARK.

Number of tweets in the dataset	Response time in hashtag counting	Response time in noun counting	Response time in Jaccard similarity
10	1.222835064	2.911193	2.761557
30	2.047077417	3.765497	3.606873
50	4.008028526	5.937976	5.719659
100	6.942538977	8.9464	8.678112
500	33.71384072	36.51038	35.83625
1000	66.13316321	70.19669	68.6782
1200	82.5823097	86.82858	85.56904

SPARK and hashtag counting, noun counting, and Jaccard similarity without using SPARK.

The graph in Figure 22 shows that SPARK will take around 40 seconds to start the response to the stream and produces the first batch, and then, it keeps on updating the result as the input stream keeps coming. The constant time shown here depends on the system specification on which we are running the program and also the speed of the internet connection to the system because of the streaming of tweets from Twitter. This response time can still be

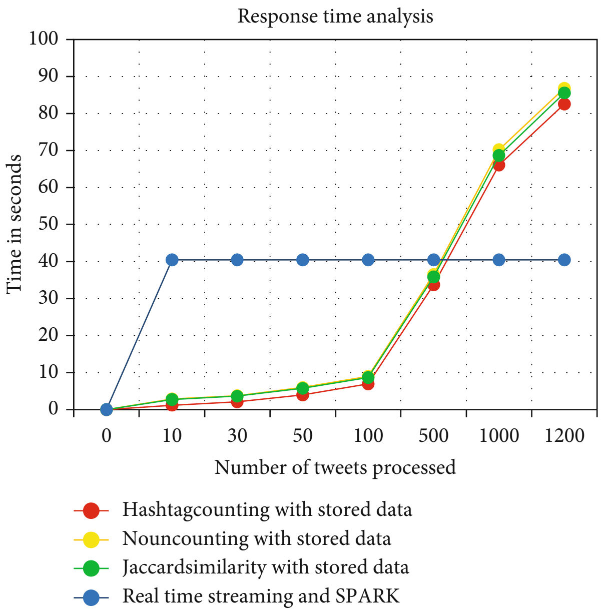

Figure 22: Comparative analysis of the execution time.

decreased using machines with powerful processors. Without using SPARK, the response will be the result itself hence have to wait until the program executes completely which is not preferable always as Twitter is a platform where so many people will tweet on so many topics in very little time.

4.3.1. API Used. We have used Twitter API to get access to the Twitter data, using Twitter API we can programmatically retrieve the data. In order to get access to the Twitter API, you will need to follow the following steps.

Step 1. Apply for twitter developer account and wait until we receive approval. Generally, Twitter provides two levels of access: one is "Standard," and another one is for "Academic research." The proposed work chosen academic research option.

Step 2. Once our account is approved, we will be able to generate or find the twitter API access credentials which are discussed below:

API key. This is basically a user name that allows you to make request to the Twitter to get access for the data.

API key secret. This is the password for your API key.

Access token. This token represents the associated Twitter account.

Access token secret. This also represents the associated Twitter account.

Bearer token. This token represents the application for which you are using the Twitter data.

Since we are building application in Python, its package manager pip provides a library called "tweepy" which is used to connect programmatically and get access to the Twitter API using the credentials that we will get in Step 2 and then download or stream data in real time.

5. Conclusion

Twitter is one of the major platforms with a large number of users worldwide. People, their interest, their opinion, likes, dislikes, events, sports tournaments, politics, movies, and the music everything are part of it. Analyzing such a rich data content platform and observing trends in it definitely will be beneficial. Analyzing Twitter trends helps to know what people are more interested in and thus helps business organizations or brands to improve their sales, political parties to understand people's emotions and needs, movie industries to get valid feedback for their performances, and much more. In this article, we have proposed some of the possible techniques that can be used to analyze Twitter trends from brute force counting techniques to topic modelling and machine learning clustering techniques. Choice of the technique depends on the purpose of analysis, the amount of data expected to be covered in the analysis model, and even the expected output formats. As everyone expects the model to be run faster and smoother, we also have proposed model development using real-time streaming and SPARK which is a big data analytics tool. The LDA technique for trend analysis resulted in an accuracy of 74% and Jaccard with an accuracy of 83% for static data. The results proved that the real-time tweets are analyzed comparatively faster in the Big Data Apache SPARK tool than in the normal execution environment.

6. Future Work

As future work, the proposed models can be modified to develop a trend analysis system that tracks the trends in a particular geographical location. This will help business organizations to target the right people in the right places to build their brand values. As the application and demand for Twitter trend analysis are always rising, the proposed techniques with few modifications can be used to fit most of the requirements.

Data Availability

The JSON data used to support the findings of this study are included within the supplementary information file.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Supplementary Materials

The supplementary file consists of 20,000 tweets collected from Twitter for the experimental analysis. The file is in the text format, and the tweets are in the JSON format. The real-time analysis is performed on live streaming of tweets and hence not stored offline. This data set may be used by the researchers to further carryout more experiments and obtain better results. *(Supplementary Materials)*

References

[1] M. Mashuri, "Sentiment analysis in twitter using lexicon based and polarity multiplication," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, pp. 365–368, IEEE, 2019.

[2] M. Wongkar and A. Angdresey, "Sentiment analysis using naive Bayes algorithm of the data crawler: Twitter," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, pp. 1–5, IEEE, 2019.

[3] R. Sharma, *Twitter Sentiment Analysis*, 2019, https://github.com/sharmaroshan/Twitter-Sentiment-Analysis.

[4] S. Yang and H. Zhang, "Text mining of Twitter data using a latent Dirichlet allocation topic model and sentiment analysis," *International Journal of Computer and Information Engineering*, vol. 12, no. 7, pp. 525–529, 2018.

[5] E. S. Negara, D. Triadi, and R. Andryani, "Topic modelling Twitter data with latent Dirichlet allocation method," in *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pp. 386–390, IEEE, 2019.

[6] N. Shahreen, M. Subhani, and M. M. Rahman, "Suicidal trend analysis of Twitter using machine learning and neural network," in *2018 international conference on Bangla speech and language processing (ICBSLP)*, pp. 1–5, IEEE, 2018.

[7] A. F. Hidayatullah and M. R. Ma'arif, "Road traffic topic modeling on Twitter using latent Dirichlet allocation," in *2017 international conference on sustainable information engineering and technology (SIET)*, pp. 47–52, IEEE, 2017.

[8] R. A. Hasan, R. A. I. Alhayali, N. D. Zaki, and A. H. Ali, "An adaptive clustering and classification algorithm for twitter data streaming in Apache Spark," *Telkomnika*, vol. 17, no. 6, pp. 3086–3099, 2019.

[9] K. Garg and D. Kaur, "Sentiment analysis on Twitter data using Apache Hadoop and performance evaluation on Hadoop MapReduce and Apache Spark," *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, , pp. 233–238, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2019.

[10] S. E. Saad and J. Yang, "Twitter sentiment analysis based on ordinal regression," *IEEE Access*, vol. 7, pp. 163677–163685, 2019.

[11] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for twitter accounts," *Mathematical and Computational Applications*, vol. 23, no. 1, p. 11, 2018.

[12] M. R. Huq, A. Ali, and A. Rahman, "Sentiment analysis on Twitter data using KNN and SVM," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 19–25, 2017.

[13] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for Twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.

[14] A. Z. Ahmed and M. Rodríguez-Díaz, "Significant labels in sentiment analysis of online customer reviews of airlines," *Sustainability*, vol. 12, no. 20, pp. 1–18, 2020.

[15] T. Rathod and M. Barot, "Trend analysis on Twitter for predicting public opinion on ongoing events," *International Journal of Computing Applications*, vol. 180, no. 26, pp. 13–17, 2018.

[16] P. Garg, R. Johari, H. Kumar, and R. Bhatia, "Trending pattern analysis of Twitter using spark streaming," in *International Conference on Application of Computing and Communication Technologies*, pp. 3–13, Springer, Singapore, 2018.

[17] N. D. Zaki, N. Y. Hashim, Y. M. Mohialden, M. A. Mohammed, T. Sutikno, and A. H. Ali, "A real-time big data sentiment analysis for Iraqi tweets using spark streaming," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1411–1419, 2020.

[18] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Generation Computer Systems*, vol. 117, pp. 47–58, 2021.

[19] H. Khan, M. U. Asghar, M. Z. Asghar, G. Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, "Fake review classification using supervised machine learning," *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV*, , pp. 269–288, Springer International Publishing, 2021.

[20] G. SRIVASTAVA, P. K. R. MADDIKUNTA, and T. R. GADEKALLU, *A Two-Stage Text Feature Selection Algorithm for Improving Text Classification*, ACM Transactions on Asian and Low-Resource Language Information Processing, 2021.

[21] A. P. Rodrigues, N. N. Chiplunkar, and R. Fernandes, "Aspect-based classification of product reviews using Hadoop framework," *Cogent Engineering*, vol. 7, no. 1, p. 1810862, 2020.

[22] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," in *International conference on intelligent data engineering and automated learning*, pp. 611–618, Springer, Berlin, Heidelberg, 2013.