SENTIMENT ANALYSIS OF COVID-19 VACCINATION IMPACT ON TWITTER TWEETS USING NLP SUPERVISED LEARNING AND RNN CLASSIFICATION COMPARISON

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Tanvir Ahmed

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

November 2022

Fargo, North Dakota

North Dakota State University
Graduate School

Title

SENTIMENT ANALYSIS OF COVID-19 VACCINATION IMPACT ON TWITTER TWEETS USING NLP SUPERVISED LEARNING AND RNN CLASSIFICATION COMPARISON

By

Tanvir Ahmed

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Simone A. Ludwig
Chair

Dr. Saeed Salem

Dr. Maria de los Angeles Alfonseca-Cubero

Approved:

11/14/2022
Date

Dr. Simone A. Ludwig
Department Chair

ABSTRACT

Twitter provides a platform for exchanging information and opinions on global concerns like the COVID-19 epidemic. During the COVID-19 pandemic, we used a collection of around 16,180 tweets to derive inferences regarding public views toward the vaccine impact once immunizations became widely available to the community. We use natural language processing and sentiment analysis techniques to uncover information regarding the public's perception of the COVID-19 vaccine. Our findings demonstrate that people are more pleased about taking COVID-19 shots than they are about some of the vaccines' side effects. We also look at people's reactions to COVID-19 safety measures after they have received the immunizations. In terms of maintaining safety precautions against COVID-19 among the vaccinated population, good attitude outnumbers negative emotion. We also estimate that around 48 percent of individuals have a neutral attitude, 36 percent have a positive opinion, and around 16 percent have a negative opinion towards vaccination. This research will help policymakers better assess public reaction and plan vaccination campaigns, as well as health and safety measures, amid the current global health crisis.

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Simone A. Ludwig, my research advisor, and my family members who supported me in the process of achieving my goals. I am also grateful to Dr. Saeed Salem and Dr. María de los Ángeles Alfonseca-Cubero for their guidance and time to serve in the supervisory committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Machine learning has revolutionized data science and ushered in a new era of technical achievement in recent decades. Some of the real-world machine learning applications that are sweeping the globe include image identification, sentiment analysis [1], product recommendations, spam/fraud detection [2] social media features, and so on. The number of people using social media has been rapidly increasing, especially in the last decade. Over the preceding year, Facebook, Twitter, YouTube, LinkedIn, and Pinterest all witnessed significant growth. With 2.8 billion monthly active users [3] Facebook is the most popular social media platform, while Twitter has roughly 300 million monthly active users [3]. Twitter is rapidly gaining popularity throughout the world and is experiencing tremendous growth. Certain users use the Twitter interface to support various opinions, such as a medium for fighting, political missions, and information dissemination, and it is playing an increasingly important role in societal development.

The Coronavirus has been one of the most popular topics on Twitter since January 2020, and it has continued to be researched to this day. There have been 3.57 million verified fatalities and 171.19 million confirmed COVID-19 cases as of June 1, 2021 [4]. Since COVID-19 immunization began to be scaled up, the situation has improved. As more proof of vaccination's good effects on transmission becomes available, public trust will grow [4] Considering this, evaluating public opinion or emotion is critical for pushing individuals to get the COVID-19 vaccine.

Governments have traditionally relied on surveys to gauge public opinion; however, these surveys frequently have flaws such as small sample sizes, closed questions, and granularity in space and time [5] We argue that social media data can be used to obtain more real-time insights

into public sentiments and attitudes with significant spatiotemporal granularity to overcome these limitations. Because social media data is largely unstructured, it may be used to extract subjects and feelings using proven artificial intelligence (AI) techniques such as machine learning and, deep learning (DL) [6].

The incredible increase in society's reliance on social media for information, as opposed to traditional news sources, and the volume of data offered, has resulted in a greater emphasis on the use of natural language processing (NLP) and AI technologies to aid text analytics [7]. This data covers a wide range of social phenomena, including cultural dynamics, social trends, natural disasters, and public health, as well as topics that are widely discussed and opinions expressed on social media. This is due to its inexpensive cost and ease of use, as well as the social network's personal connectivity. Professional opinion leaders (and state actors) are increasingly using social media to enhance their message through network effects. Many businesses utilize social media to advertise their products, brands, and services [8]. As a result of the reviews and experiences shared by end users, an information-rich reservoir is created, and this information is stored as text, making open communication platforms and social media important information sources for researching issues involving rapidly changing public sentiment [9]. Because there hasn't been a global pandemic in over a century, this is a unique opportunity to investigate a global issue.

The field of natural language processing (NLP) and its application to social media analysis has grown at a breakneck pace. However, utilizing NLP-methods to deduce a text's underlying meaning remains a difficult task. Even the latest NLP tools have been found to be "susceptible to hostile texts" [10, 11]. As a result, it is critical to gain a better grasp of the limitations of text categorization methods, as well as related machine learning (ML) algorithms.

It's also crucial to see if there's a way to circumvent these restrictions by combining various technologies and applying a synergistic concept. As a result, the technique may help develop AI applications in human communication and in extracting insights from texts. Sentiment analysis enters the picture, with the goal of establishing efficient algorithmic techniques for the automatic extraction of the writer's sentiment from the text. Relevant works are centered on tracking the sentiment valence (or polarity) of single utterances, typically in the form of short text posts that are laden with subjectivity and uncertainty [12]. The use of social media language, such as Twitter, is not normalized, and its utterances tend to break vocabulary and grammar rules; they are unstructured, syntactically quirky, and often quite casual. Users employ made-up terms and jargon in their posts, and they regularly add URLs. They also use abbreviations, nonstandard punctuation, improper spelling, emoticons, slang, idioms, and abbreviations. Context-aware ways to utilize ambiguity are either nonexistent or inefficient due to the lack of facial expressions, visual, and tone-of-voice clues.

Sentiment analysis is the process of classifying subjective opinions from text, audio, and video sources [13] to determine polarities (positive, negative, and neutral), emotions (anger, sadness, and happiness), or states of mind (interest vs disinterest) toward target topics, themes, or aspects of interest [14]. A related approach, known as stance detection [15], provides a stance label (favorable, against, or none) to a post on a certain specified target, which may or may not be referenced to or the subject of discussion in the post. Currently, such methodologies are underutilized in health-care research. Drawing on AI-enabled social media analysis to enhance public policy research has a lot of untapped promise.

The focus of this study is on public tweets on the COVID19 vaccine impact. We scrap tweets based on vaccination-related phrases and the many vaccine names available on the market

for covid. We are attempting to ascertain public perceptions of vaccination effects, particularly in the United States. People are more likely to acquire the covid vaccination because of these consequences. Furthermore, people have had a variety of reactions to vaccination, some of which are favorable and others of which are bad [3]. We analyzed the data to generate a comprehensive picture of the COVID19 vaccine effects in the United States based on people's opinions.

In this study, sentiment analysis is used to get a sense of how individuals felt about the COVID-19 immunization. The overall objectives of this study can be represented as follow-

(a) We looked at tweets on nine different types of COVID-19 vaccines to see how people felt about them. This research is valuable in determining whether people are hesitant to vaccinate due to the potential negative effects of vaccinations. Furthermore, this reaction demonstrates people's interest in and readiness to receive vaccinations because of the immunization campaign.

(b) We compiled a list of tweets that mentioned vaccination in conjunction with other health-related topics. We can learn about people's opinions on how they are following health standards after getting vaccinated by looking at the public mood on these tweets.

2. RELATED WORK

Several studies on evaluating the Twitter dataset on various themes during the COVID-19 outbreak have been published [16-18]. Only a few research [19, 20] have investigated Twitter data connected to COVID-19 immunization. During the COVID-19 epidemic, Glowacki et al. [21] used text mining to discover addiction issues. They compiled a list of 14 common subjects from public tweets including the phrases "addiction" and "covid," as well as debate on those issues. Only 3301 tweets are included in their sample. They want to find out what people are saying about addiction on Twitter during the COVID epidemic, but they are not doing sentiment analysis on addiction because of the pandemic. The authors used Twitter data connected to "Mask" in [16]. They discovered that the number and polarity of mask-related tweets grew dramatically between March 17 and July 27, 2020.

Xue et al. [22]employed Latent Dirichlet Allocation (LDA), a machine learning technique, to detect common unigrams, bigrams, salient topics and themes, and attitudes in four million tweets on COVID-19 using 25 distinct hashtags from March 1 to April 21, 2020. They divided the sentiments into eight categories using the NRC Emotion Lexicon: anger, anticipation, fear, surprise, sorrow, pleasure, disgust, and trust. Pano and Kashef [17] used VADER (Valence Aware Dictionary for Sentiment Reasoning) which is a sentiment analysis measure the intensity of an emotion to do sentiment analysis on tweets about bitcoins during the COVID-19 epidemic. For linking the emotion ratings of the tweets with bitcoin price, they examined 13 different text preparation algorithms. During the COVID-19 epidemic, Bhagat et al. [18]used TextBlob to do sentiment analysis on online education by extracting 154 articles from online news and blogging platforms. Their findings suggest that over 90% of the articles are favorable, with blogs generally being more positive than newspaper stories.

Chen and Dredze [23] were the first to use Twitter to examine vaccine-related imagery. The purpose was to track the spread of pictures used in vaccine-related tweets and apply a logistic regression model to predict whether they were retweeted. The authors have provided a labeled dataset that may be used to classify photos based on their emotion Villavicencio et al. [19]conducted a sentiment analysis of COVID-19 immunization tweets in the Philippines for their research. The authors utilized the RapidMiner data science program to identify English and Filipino language tweets (993 tweets) with 81.77 percent accuracy, revealing Filipino opinions regarding COVID-19 vaccinations. Chaudhri et al. [20]have investigated if individuals are in favor of getting vaccinated against COVID-19. Their findings reveal that people had a slightly favorable attitude toward obtaining COVID-19 vaccination doses on average. However, the authors employed a relatively small number of tweets in their research, only 900. They did not say how they chose those tweets or what factors they considered when harvesting them. The article also fails to mention the timeline for scraping the tweets.

Only the research in [19, 20] are linked to our work in any way. However, Villavicencio et al. [19]'s research is limited to tweets from the Philippines, whereas we gather tweets from all around the world. As a result, we have roughly 1.2 million tweets, but they only looked at 993 of them. This study likewise uses the Nave Bayes model to predict categorization, whereas we categorize tweets using a lexicon-based classifier and the freely accessible tools TextBlob and VADER Villavicencio et al. [19] manually annotated the training data, that is, they gave sentiment labels for the training data to predict the test data. We do not guess at the sentiment labels; instead, we use well-known sentiment analysis algorithms to determine them. As a result, we are unable to make any accuracy comparisons with Villavicencio et al. [19]. The Twitter data

collection criterion and timeline are missing from [20], which are required if we wish to compare our results to theirs.

Using the ensemble approach in R, the Centers for Disease Control and Prevention (CDC) [24]forecasts the cumulative mortality for COVID-19 four weeks ahead of time. The CDC predicts weekly death/cumulative deaths, daily hospitalization, and weekly new COVID-19 cases using this model. They do not, however, incorporate any vaccination data for the projection of the vaccination situation in the United States (state and national). [25]uses ensemble learning of the well-known regression algorithms in WEKA to forecast COVID-19 fatalities and cases in the 15 nations of South and Central Europe. We cannot test how well their classifier [26] would operate on the vaccine dataset because the dataset and specific implementation are not sufficiently explained. Other research has looked at the stock market [27-29], company sales [30, 31], temperature [32], weather [33], energy use [34, 35], power [36, 37], and so on. In the United States, we couldn't identify any research that demonstrates a projection based on vaccination data. As a result, we are unable to compare the accuracy of our model to previous efforts.

3. METHODOLOGY

3.1. Preliminaries

In this part, we go through the many Python libraries we utilized in our research, as well as the evaluation metrics, sentiment analysis methodologies, and performance measures for time series forecasting modeling.

There are two major approaches to sentiment analysis. Those are: -

- Supervised machine learning or deep learning approaches.
- Unsupervised lexicon-based approaches.

We use the first strategy since we have a manually labeled dataset.

To analyze our forecasting model, we used the following error and accuracy criteria. Several assessment criteria, such as sensitivity, specificity, and accuracy, were employed to assess the performance of our model. These were produced from the confusion matrix and applied to the classifier assessment [38, 39], as illustrated in Equations (1) through (4).

$$\text{Accuracy} = (TP + TN)/ (TP + FP + TN + FN) \quad (1)$$

$$\text{Sensitivity} = TP/ (TP + FN) \quad (2)$$

$$\text{Specificity} = TN/ (TN + FP) \quad (3)$$

$$\text{F-1 Score} = 2TP / (2TP + FP + FN) \quad (4)$$

Here:

TP = number of positive examples correctly classified

TN = number of negative samples correctly classified

FN = number of positive observations incorrectly classified

FP = number of negative samples incorrectly classified

In this part, we will go through our Twitter dataset and how we gathered it. We discuss our data pre-processing processes for sentiment analysis. We also show the methodologies for feature development, training, and testing for our machine learning forecast model for COVID-19 immunization in the United States. In this area, we also discuss the computational tools and environment.

3.2. Computational Tools/Libraries

For the sentiment analysis, we employed a variety of Python modules. Tweepy [40] was used to scrape tweets and collect data from Twitter. We utilized NLTK [41] for data preparation. We utilized CountVectorizer [42], TF-IDF[43], Word2vec [44], and Doc2Vec [45] to extract features.

3.3. Environment

Experiments in this study were carried out using a personal computer with an Intel(R) Core (TM) i7-8750H CPU running at 2.20GHz and 2.21GHz, 16 GB of RAM, 1 TB hard drive, and 64-bit Windows 11 OS.

3.4. Sentiment Analysis

We present our methods for performing sentiment analysis on Twitter data connected to COVID-19 immunization in this part. Figure 1 depicts the twitter authentication, while Figure 2 depicts the schematic diagram for the various phases of our sentiment analysis approach on COVID-19 vaccination-related tweets.

3.4.1. Twitter Data Collection

Using the Python module Tweepy [40], we collected 16,180 original tweets using the Twitter API [46]. 'covaxin','sinopharm','sinovac','moderna', 'pfizer', 'biontech', 'oxford', 'astrazeneca','sputnik', 'vaccinations', 'vaccine', 'vaccines', 'immunization', 'vaccinate',

'vaccinated', 'vaccinations'. Figure 3 shows more information on the keywords. We also exclusively gathered tweets in English, and we utilized NLTK to analyze the data. Figure 1 depicts our Twitter data collecting pipeline. By sampling 1% of all public tweets in near real time, Twitter's API enables access to 1% of all public tweets. Although concerns about skewed or imbalanced data from a 1% sample of all tweets may emerge, it has been demonstrated that sentiments identified in tweet samples retrieved using the API and the whole twitter dataset represent the same sentiment percentage with very little difference (1.8%) [47]. We only made the tweet IDs matching to the collected tweet text publicly available in accordance with the Twitter content redistribution policy [48, 49].

Figure 1: Twitter Authentication

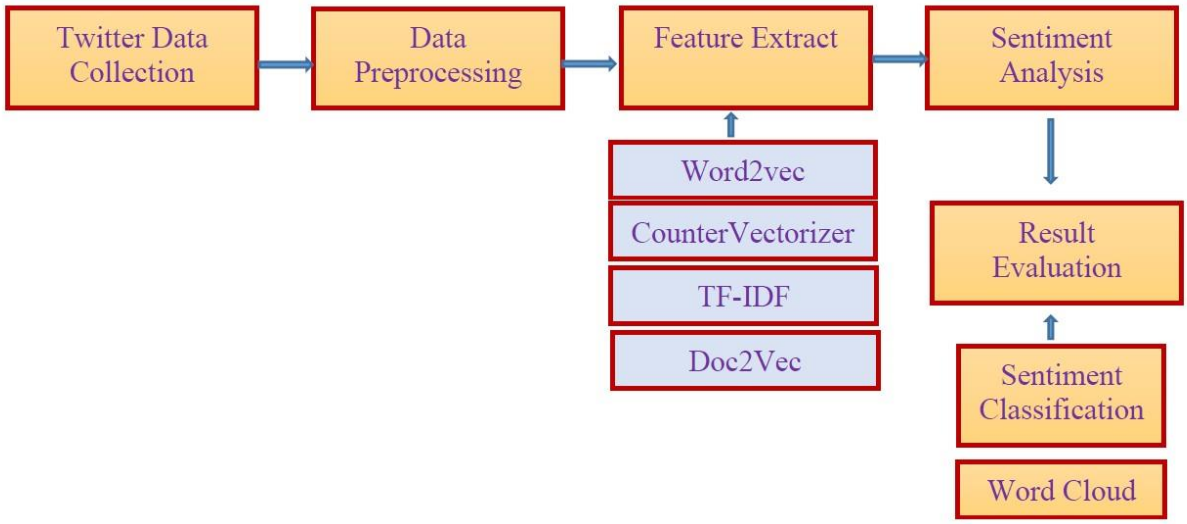


Figure 2: Schematic Diagram

	tweet_id	timestamp	user_id	text	hashtags	sentiment
0	1.460000e+18	2021-11-08T03:49:15+00:00	1.290000e+18	Same folks said daikon paste could treat a cyt...	['PfizerBioNTech']	2
1	1.460000e+18	2021-11-08T02:00:44+00:00	3.480000e+08	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	['coronavirus', 'SputnikV', 'AstraZeneca', 'Pf...	2
2	1.460000e+18	2021-11-08T01:43:37+00:00	3.495831e+07	it is a bit sad to claim the fame for success ...	['vaccination']	2
3	1.460000e+18	2021-11-08T01:40:23+00:00	1.670000e+08	#CovidVaccine \n\nStates will start getting #C...	['CovidVaccine', 'COVID19Vaccine', 'US', 'paku...	1
4	1.460000e+18	2021-11-08T01:36:25+00:00	1.720000e+08	while deaths are closing in on the 300,000 mar...	['PfizerBioNTech', 'Vaccine']	1
5	1.460000e+18	2021-11-08T00:59:08+00:00	1.050000e+18	Trump announces #vaccine rollout 'in less than...	['vaccine']	1
6	1.460000e+18	2021-11-08T00:33:46+00:00	5.518660e+07	UPDATED: #YellowFever & #COVID19 #Immunity...	['YellowFever', 'COVID19', 'ImmunityPassports'...	1
7	1.460000e+18	2021-11-08T00:25:24+00:00	1.570000e+09	Coronavirus: Iran reports 8,201 new cases, 221...	['Iran', 'coronavirus', 'PfizerBioNTech']	1
8	1.460000e+18	2021-11-07T23:59:15+00:00	3.380000e+09	.@Pfizer will rake in billions from its expens...	['CovidVaccine']	0
9	1.460000e+18	2021-11-07T23:04:55+00:00	1.782277e+07	The trump administration failed to deliver on ...	['COVIDIOTS', 'coronavirus', 'CovidVaccine']	0

Figure 3: Dataset

After utilizing the Twitter API, the tweet_id, timestamp, user_id, text, and hashtags are obtained. By reading each tweet, we manually entered the level value, which is the Sentiment column, into the dataset. Positive sentiment is categorized as 2, neutral sentiment is categorized as 1, and negative sentiment is categorized as 0. The most common is neutral, and the least common is negative. There are 2545 negative sentiments, 7865 neutral sentiments, and 5770 positive sentiments. Figure 4 depicts the dataset's sentiment distribution.

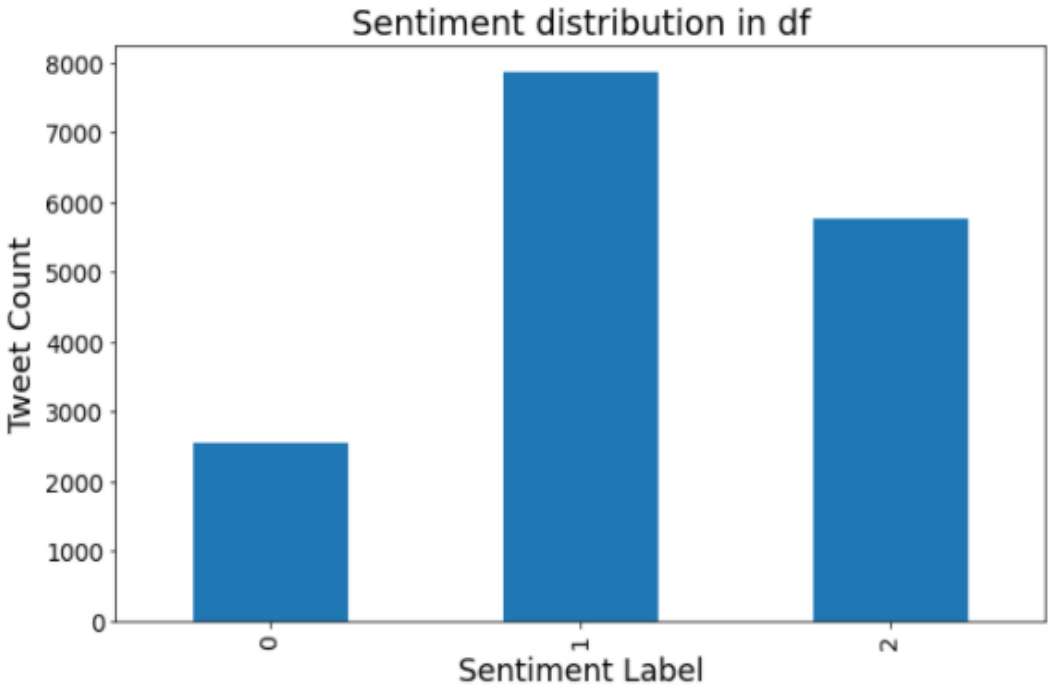


Figure 4: Sentiment Distribution

3.4.2. Pre-Processing of Data

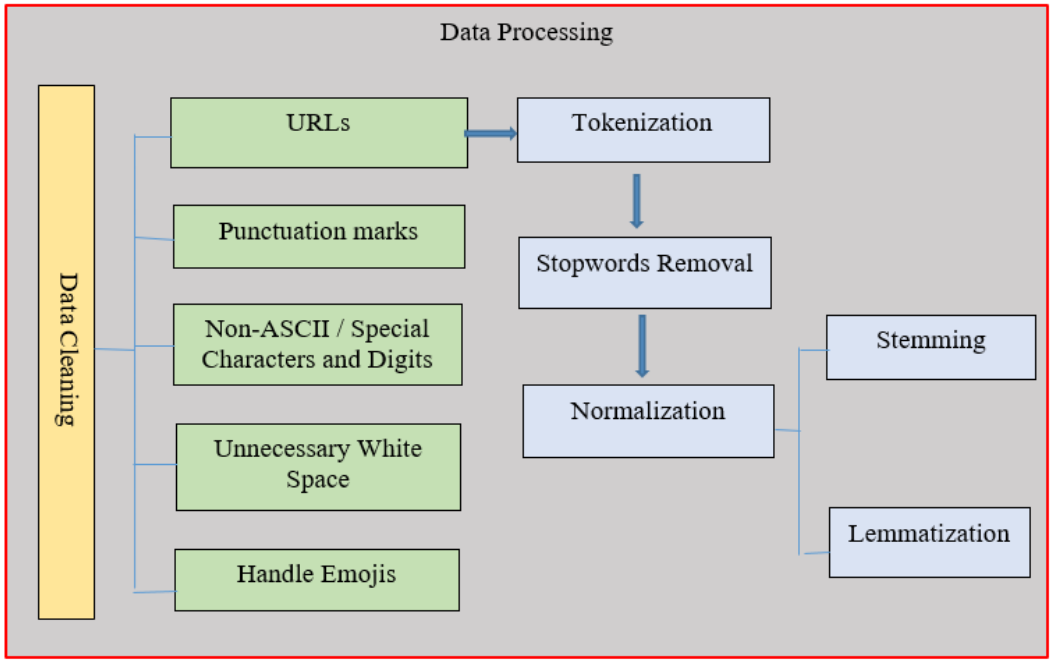


Figure 5: Data Preprocessing Steps

Figure 5 shows the data pre-processing technique to make the data clean, and to make the data suitable for the feature extraction. The details procedure is discussed below: -

- Data Cleaning: In this phase, we deleted the urls, punctuation marks, and special characters, emojis, white spaces and bad ascii digits.
- Tokenization: This stage divides the text into words (the smallest unit).
- Stopwords Removal: Some terms in the text, such as "and", "but" "so" and others, are often used but are useless in the analysis. We don't utilize preset stopwords from any libraries because removing "not" or similar negative words would radically affect the tone of the statement. As a result, we employed our own stopword list, which we created by updating the most comprehensive stopword collection for the English language [3]. We deleted all negative terms from the above-mentioned list so that sentiment analysis would not be affected.

- Data Normalization:
  - Stemming: Stemming: We standardized the words in this phase by truncating them to their stem words. Porter Stemmer from the NLTK library was used.
  - Lemmatization: Then, according to the part of speech, we lemmatized words to retrieve the root words.

3.4.3. Word Cloud

A wordcloud is a visual representation in which the most often used words are displayed in larger font sizes and the less frequently used words are displayed in smaller font sizes. Figure 6 depicts a word cloud for the entire dataset, Neutral Sentiment, Positive Sentiment, and Negative Sentiment.

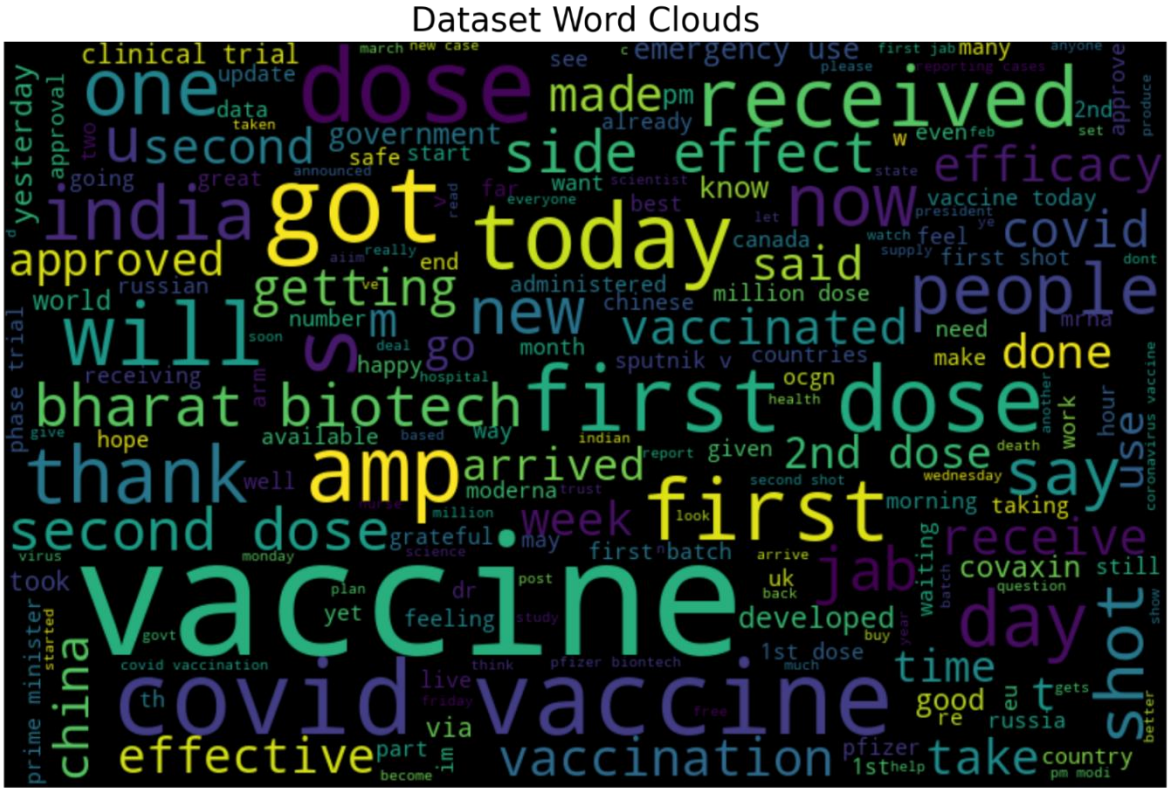
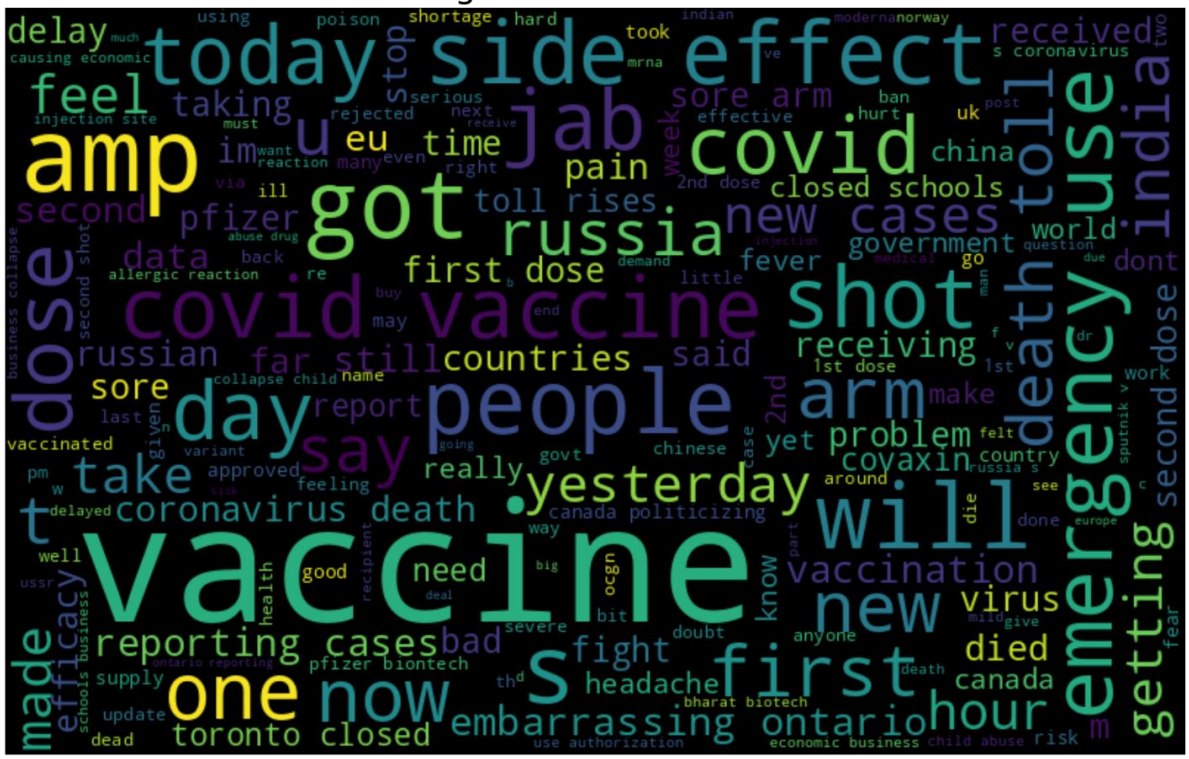


Figure 6: Word Clouds

Negative Word Clouds



Neutral Word Clouds

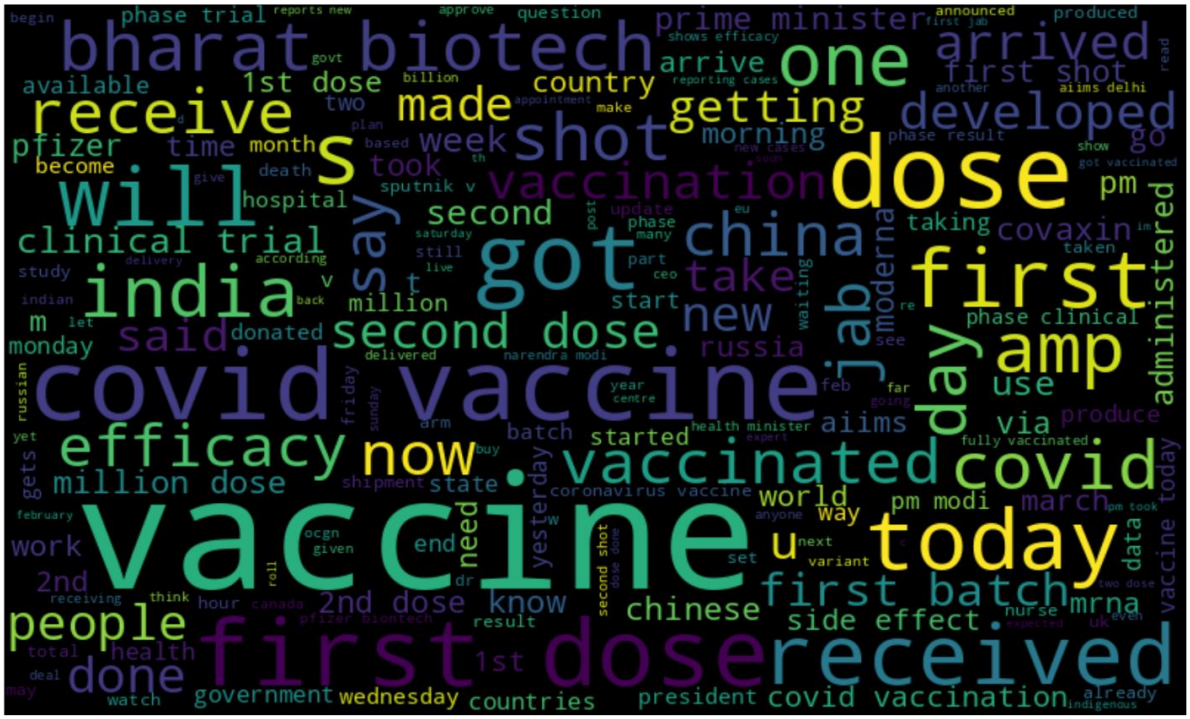


Figure 6: Word Clouds (continued)

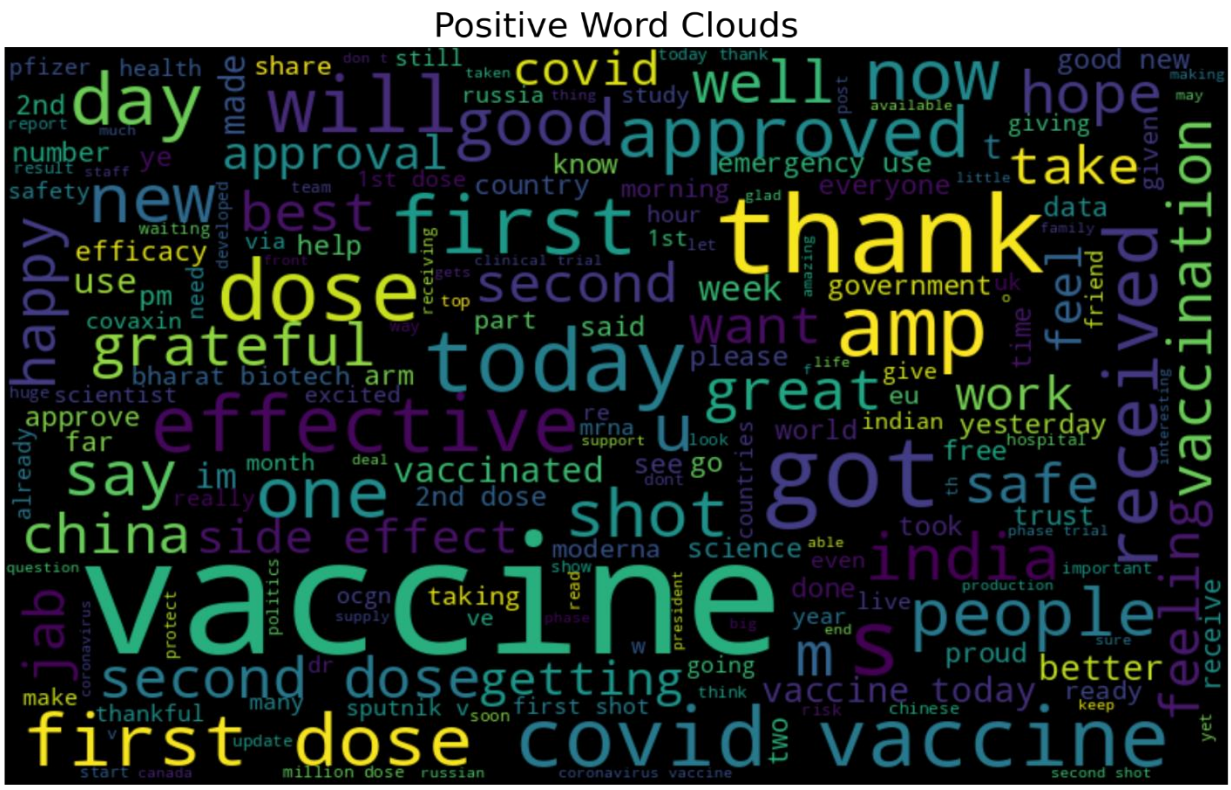


Figure 6: Word Clouds (continued)

3.4.4. Hashtags

Hashtags on Twitter are a way to keep track of what is trending on the platform at any given time. They aid in the classification of tweets into various attitudes. Figure 7 depicts hashtags for the Neutral Sentiment, Positive Sentiment, and Negative Sentiment.

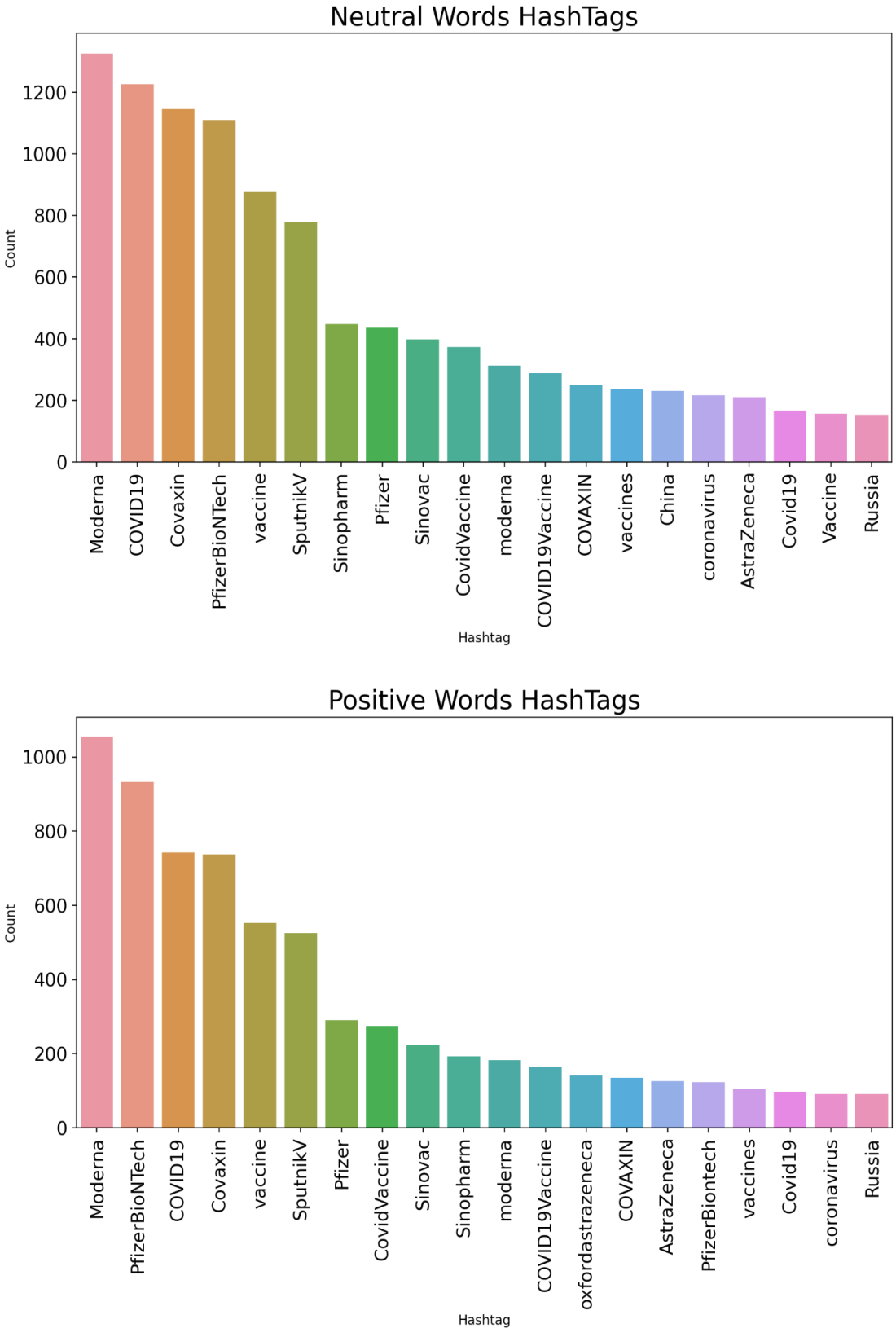


Figure 7: Hashtags of Top Twenty Words

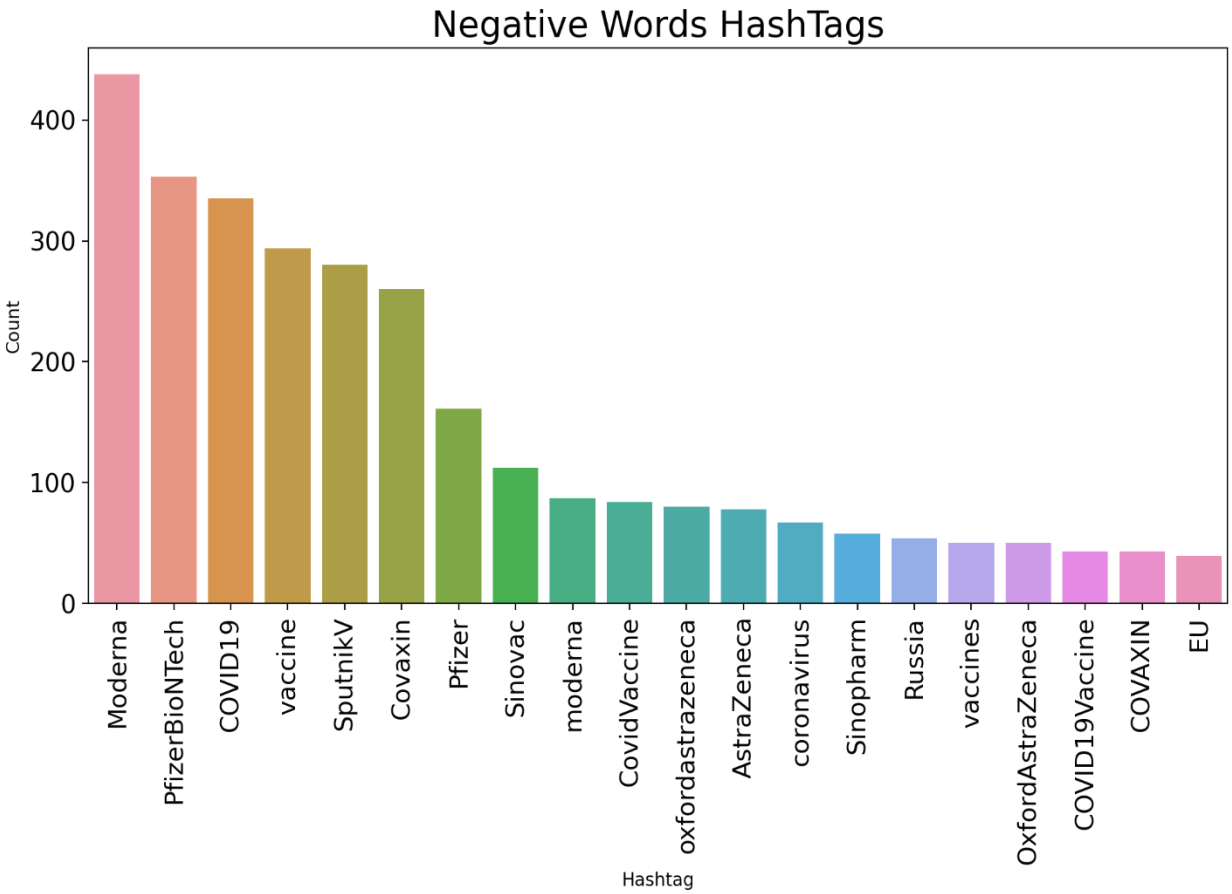


Figure 7: Hashtags of Top Twenty Words (continued)

3.4.5. Feature Extraction

To provide output for the test data, Machine Learning algorithms learn from a pre-defined collection of features from the training data. However, the primary issue with language processing is that machine learning algorithms cannot work directly on raw text. To transform text into a matrix (or vector) of features, we'll need some feature extraction algorithms. Text features may be built using a variety of approaches, including Bag of Words, TF-IDF, and Word Embeddings, depending on the application.

3.4.6. Bag of Words

Bag-of-words (BoW) is one of the most basic approaches for transforming tokens into a set of characteristics is to use words. Each word is utilized as a feature for training the classifier in the BoW model, which is employed in document classification [49]. The first step is text-preprocessing which involves: converting the entire text into lower case characters and removing all punctuations and unnecessary symbols. The second stage is to establish a vocabulary that includes all the corpus's unique terms. In the third stage, we generate a matrix of features by dividing each word into its own column and assigning each row to a review. Text vectorization is the term for this procedure. The existence (or absence) of a term in the review is indicated by each item in the matrix. If the term appears in the review, we add a 1 and if it does not, we put a 0.

3.4.7. IF-IDF Vectorizer

The phrase term frequency-inverse document frequency (TF-IDF) stands for term frequency-inverse document frequency. It draws attention to a specific issue that, while not common in our corpus, is extremely important. The TF–IFD value rises in proportion to the number of times a word appears in the document and falls in proportion to the number of documents in the corpus containing the term. It is divided into two sub-sections: Term Frequency (TF) and Inverse Document Frequency (IDF) [50].

Term Frequency (TF): The word frequency defines how often a phrase appears throughout the document. It might be compared to the likelihood of discovering a word inside a document. It calculates the number of times a word w_i occurs in a review r_j with respect to the total number of words in the review r_j. It is formulated as:

$$tf(w_j, r_j) = \frac{No.of\ times\ w_j\ occurs\ in\ r_j}{Total\ no.of\ words\ in\ r_j}$$

A different scheme for calculating tf is log normalization. And it is formulated as:

$$tf(t,d) = 1 + logf_{t,d}$$

where, $f_{t,d}$ is the frequency of the term t in document *d*.

Inverse Document Frequency (IDF): The inverse document frequency is a metric that determines whether a phrase is rare or common across all documents in a corpus. It emphasizes terms that appear in a small number of papers across the corpus, or in plain English, words with a high IDF score. The logarithm of the overall term is determined by dividing the total number of documents D in the corpus by the number of documents containing the word t.

$$idf(d,D) = \log\frac{|D|}{\{d_{€}\, D: t_{€}D\}}$$

where,

$f_{t,d}$ is the frequency of the term t in document D.

|D| is the total number of documents in the corpus.

$\{d_{€}\, D: t_{€}D\}$ is the count of documents in the corpus, which contains the term t.

The value of IDF (and consequently TF–IDF) is larger than or equal to 0 since the ratio inside the IDF's log function must always be bigger than or equal to 1. The ratio within the logarithm approaches 1 when a phrase appears in a high number of documents, and the IDF approaches 0.

3.4.8. Word2Vec

In most NLP models, Word2Vec is commonly employed. It converts the text into vectors. Word2vec is a two-layer net that uses words to analyze text. The text corpus is the input, and the output is a set of vectors, with feature vectors representing the words in the corpus. While Word2vec is not a deep neural network, it does turn text into a type of computation that deep neural networks can understand. Word2vec's objective and usefulness is to gather vectors of the

same words in vector space. That is, it looks for mathematical parallels. Word2vec generates vectors based on numerical representations of word components, as well as attributes like individual word context. It accomplishes this without the need for human interaction [51].

Word2vec can create the most accurate predictions about a word's meaning based on prior appearances if given enough data, use, and circumstances. That guess may be used to create word-and-word combinations (for example, "big", "huge", to state "little" is "tiny"), or to group and divide texts by topic. These collections may be used in a variety of sectors, including scientific study, legal discovery, e-commerce, and customer relationship management, to help with search, emotional analysis, and suggestions. The Word2vec net produces a lexicon with each item having its own vector, which may be used in an in-depth reading net or simply to identify the association between words.

Word2Vec excels at capturing the meaning of words in a context. There are two types of flavors available. We are given the nearby words in one technique, called the continuous bag of words (CBoW), and the middle word in another approach, called skip-gram, and we forecast the neighboring words. Once we have a pre-trained set of weights, we can keep it and utilize it for word vectorization later without having to convert the data again. They are kept on a lookup in figure 8.

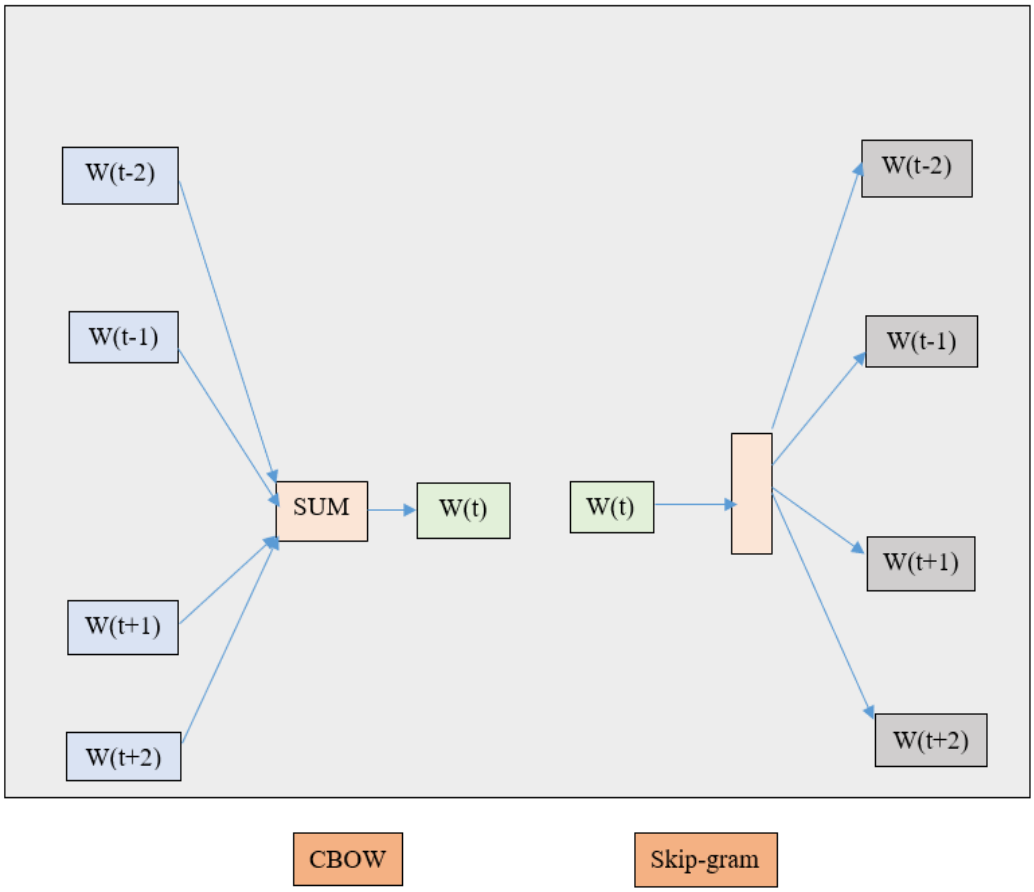


Figure 8: Word2vec

3.5 Machine Learning Algorithms

Classification is a supervised learning technique that classifies unknown data into a finite set of classes by learning an objective function that maps each feature into one of the target classes [52, 53]. The objective function is referred to as the classification model. Classification is applied to many fields to develop the best-performing model by experimenting with different classification algorithms [53] [54]. We use the following well-known machine learning regression algorithms to build our forecasting model classifier. I will add few lines here. Classification and regression

3.5.1. Support Vector Machine

A support vector machine produces a hyper-plane or set of hyper-planes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyper-plane with the most significant distance to the nearest training data point of any class [53]. In two-dimensional space, the support vectors halved a plane into two fragments through a line where each cluster denotes classes.

3.5.2. Logistic Regression

Logistic Regression is a Machine Learning method that is used to solve classification issues. It is a predictive analytic approach that is based on the probability notion. A Logistic Regression model is like a Linear Regression model, except that the Logistic Regression utilizes a more sophisticated cost function [55], which is known as the 'Sigmoid function' or the 'logistic function' instead of a linear function. The logistic regression hypothesis suggests that the cost function be limited to a value between 0 and 1. As a result, linear functions fail to describe it since it might have a value larger than 1 or less than 0, which is impossible according to the logistic regression hypothesis.

3.5.3. Recurrent Neural Network

Data travels from the input layer to the output layer [38], and the linkages between the layers are only one way, forward, and never touch a node again. A recurrent neural network (RNN) is a type of artificial neural network in which nodes form a directed or undirected graph along a temporal axis. Many buried layers of neurons with tanh, rectifier, and max-out activation functions may be found in the network. High prediction accuracy may be achieved using advanced features such as adaptive learning rate, rate annealing, momentum training, dropout, L1 or L2 regularization, checkpointing, and grid search [17]. Each compute node uses multi-

threading (asynchronously) to train a copy of the global model parameters on its local data and adds to the global model on a periodic basis via model averaging across the network.

3.5.4 Stochastic Gradient Descent (SGD)

In machine learning, the Stochastic Gradient Descent (SGD) technique is crucial. It is a stochastic process with constant learning rates that, following an initial period of convergence, produces samples from a stationary distribution. In scalable Bayesian Markov Chain MonteCarlo (MCMC) approaches, where the objective is to produce samples from a conditional distribution of latent variables given a data set, stochastic gradients (SG) have also been employed. In Bayesian inference, our objective is to approximation the posterior of a probabilistic model $p(\theta,x)$ given data x and hidden variables [56].

$$p(\theta|x) = \exp\{\log p(\theta,x) - \log p(x)\}$$

3.5.5 Multi-layer Perceptron (MLP)

A neural network is a set of connected input/output units in which each connection has a weight. During the training phase, the network learns by adapting the weights to forecast the accurate class label of the input samples. Neural networks involve prolonged training times and are, therefore, more appropriate for applications where this is feasible. The most popular neural network algorithm is back-propagation – Multilayer feed-forward networks. A multilayer feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer.

3.5.6 K-Nearest Neighbor (KNN) classifier

The KNN classifier is one of the most used classifiers as it is a simple and effective non-parametric approach for classification. It has one parameter named K that identifies the number

of selected nearest neighbors to predict the class labels of the unknown samples [57]. The value of K has a substantial impact on classification performance.

3.5.7 Random Forest

Random Forest (RF) is a robust classification and regression technique. When given a data set, Random Forest (RF) generates a forest of classification trees rather than a single classification tree [100]. Each of these trees is a weak learner built on a subset of rows and columns. More trees will reduce the variance. It takes the average prediction over all their trees to make a final prediction, whether predicting a class. Sometimes, it uses the highest voting of all trees to make a final prediction.

3.5.8 Ada Boost

AdaBoost is one of the most popular algorithms. It constructs a robust classifier with a linear combination of member classifiers. The member classifiers are selected to minimize the errors in each iteration step during the training process. AdaBoost provides a straightforward and helpful method to generate ensemble classifiers. The performance of the ensemble depends on the diversity among the constituent classifiers as well as the performance of each member classifier [58]. It feeds each classifier separately and modifies the distribution of training data directly. The training dataset's weights are first spread equally among the training samples. The weights relating to each classifier's contributions are adjusted during the boosting operation, though, based on how well each classifier performed individually on the partitioned training dataset [59].

3.5.9. Bagging

Bootstrap Aggregation is a step in the ensemble machine learning meta-algorithm known as bagging. It classifies a new instance using additional voting procedures and combines the

predictions of many equal-weighted models. To develop the classifier model as its parameters, the bagging approach needs a collection of cases with a fixed size and numerous iterations [59].

3.5.10. Extra Trees

According to the classical top-down procedure, the Extra-Trees algorithm builds an ensemble of the unpruned decision or regression trees. Its two main differences from other tree-based ensemble methods are that it splits nodes by choosing cut-points at random and uses the whole learning sample to grow the trees. Researchers utilized extra tree to select the important features

3.5.11. Decision Trees

Decision tree induction is the process of learning decision trees using training samples with class labels. A decision tree is a tree structure that resembles a flowchart, where each internal node represents a test on an attribute, each branch a test result, and each leaf node a class label. The root node is the topmost node in a tree [53].

4. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the experimental results and performance evaluation of our model. We favored python programming language for the implementation of traditional machine learning and deep learning algorithms. It is worth mentioning that we used 70% as training and 30% were applied for the testing.

The performance for the Bag of words features performance is shown in Table 1. From the table 1, we can see considering all the metrics that we considered in this study, RNN outperforms all other classifiers that we investigated in this study. Next, Stochastic Gradient Descent (SGD) perform well. We have got 80% precision by using the Stochastic Gradient Descent (SGD), AdaBoost, and bagging classifiers. Support vector classifier also provide better performance. Among all the classifier that we utilized KNN performs worst with 60% accuracy, recall 60% and 65% precision. In terms of F-1 score it performs very poor with only 56%.

Table 1: Performance for the Bag of Words Features

Algorithms	Precision %	Recall %	F1-Score	Accuracy %
Support Vector Classifier (SVC)	78	78	78	78
Stochastic Gradient Descent (SGD)	80	80	79	80
Multi-layer Perceptron (MLP)	76	76	76	76
KNeighbors	65	60	56	60
Random Forest	77	78	77	78
Ada Boost	80	77	75	77
Bagging	80	78	77	78
Extra Trees	76	77	77	77
Decision Tree	73	73	73	73
Logistic Regression	77	76	76	76
Recurrent Neural Network (RNN)	87	87	87	**87**

The performance of the TF-IDF features is shown in Table 2. From table 2, when all measures used in this study are considered, RNN performs better than other classifiers. Then Extra Tress perform admirably. Using the Extra Tress and bagging classifiers, we achieved an accuracy of 80%. Better performance is also provided using stochastic gradient descent (SGD). KNN performs the lowest out of all the classifiers we used, with accuracy, recall, and precision of 61%, 61%, and 63% respectively. With only a 61% F-1 score, it performs quite poorly.

Table 2: Performance for the TF-IDF Features

Algorithms	Precision %	Recall %	F1-Score	Accuracy %
Support Vector Classifier (SVC)	78	78	77	78
Stochastic Gradient Descent (SGD)	79	79	78	79
Multi-layer Perceptron (MLP)	77	77	77	77
KNeighbors	63	61	57	61
Random Forest	79	79	78	79
Ada Boost	79	76	75	76
Bagging	80	79	78	79
Extra Trees	80	80	79	80
Decision Tree	73	73	73	73
Logistic Regression	76	74	75	74
Recurrent Neural Network (RNN)	86	86	86	86

The performance of word2vec features is shown in Table 3. From table 3, when all measures used in this study are considered, RNN performs better than other classifiers. The performance of multi-layer perceptron's is good (73%). Using the bagging classifiers, we were able to achieve 72% precision. Additionally, the Extra Tress classifier provides good

performance. With 54% accuracy, 54% recall, and 54% precision, Decision Tree performs the least well out of all the classifiers we used. With only a 54% F-1 score, it performs quite poorly.

Table 3: Performance for the Word2vec Features

Algorithms	Precision %	Recall %	F1-Score	Accuracy %
Support Vector Classifier (SVC)	68	67	67	67
Stochastic Gradient Descent (SGD)	66	66	65	66
Multi-layer Perceptron (MLP)	73	73	73	73
KNeighbors	63	63	63	63
Random Forest	69	67	65	67
Ada Boost	61	61	61	61
Bagging	72	72	71	72
Extra Trees	69	68	66	68
Decision Tree	54	54	54	54
Logistic Regression	66	63	64	63
Recurrent Neural Network (RNN)	82	82	82	82

The performance of doc2vec features is shown in Table 4. From the table 4, we can see considering all the metrics that we considered in this study, RNN outperforms all other classifiers that we investigated in this study. Next, Bagging performs well. We have got 62% precision by using the Extra Tress classifiers. Support vector classifier also provide better performance. Among all the classifier that we utilized Decision Tree performs worst with 47%

accuracy, recall 47% and 47% precision. In terms of F-1 score it performs very poor with only 47%.

Table 4: Performance for the Doc2vec Features

Algorithms	Precision %	Recall %	F1-Score	Accuracy %
Support Vector Classifier (SVC)	61	61	61	61
Stochastic Gradient Descent (SGD)	56	55	55	55
Multi-layer Perceptron (MLP)	57	57	57	57
KNeighbors	58	56	51	56
Random Forest	58	58	53	58
Ada Boost	54	56	54	56
Bagging	61	62	59	62
Extra Trees	62	58	53	58
Decision Tree	47	47	47	47
Logistic Regression	58	54	55	54
Recurrent Neural Network (RNN)	86	86	86	86

Here it is to be mentioned that, for the RNN classifier hidden layers we used RELU activation function, Dropout is set to 0.3, final layer activation function is Softmax and optimizer is Adam.

Traditional machine learning performs well in bag of words and TF-IDF features, but poorly in doc2vec features, according to the tables. RNNs, on the other hand, outperform

classical machine learning algorithms in all four different feature sets. Most notably, it performed exceptionally well in the doc2vec feature set, while classical machine learning performed poorly. The performance of Decision Tress is low in the word2vec and doc2vec feature sets, but reasonable in the Bag of Words and TF-IDF feature sets. While KNN performs significantly better in the doc2vec and word2vec feature sets, it performs the poorest on the bag of words and TF-IDF features.

The RNN performance loss and accuracy for a bag of words are shown in Figure 9. X-axis is the number of epochs and Y-axis is the accuracy measures. Figure 10 depicts the TF-IDF RNN performance loss and accuracy. For Word2Vec, Figure 11 demonstrates the RNN performance loss and accuracy. For Doc2Vec, Figure 12 depicts the RNN performance loss and accuracy.

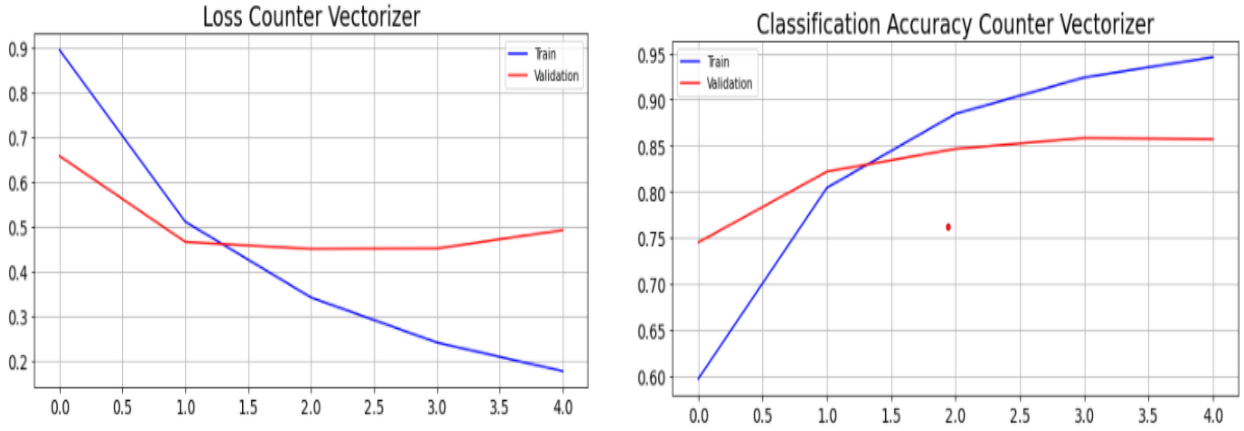


Figure 9: Loss vs Accuracy for Bag of Words Features

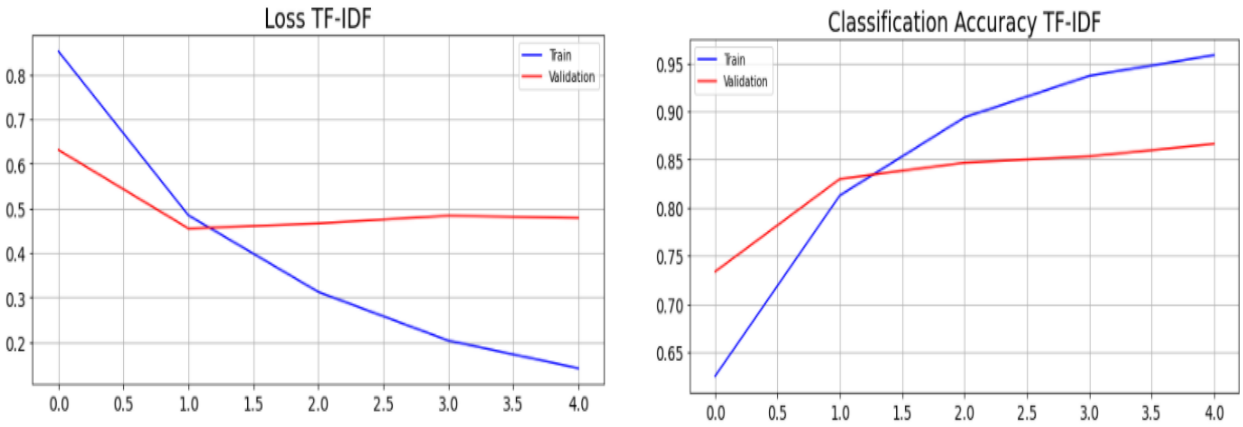


Figure 10: Loss vs Accuracy for TF_IDF Features

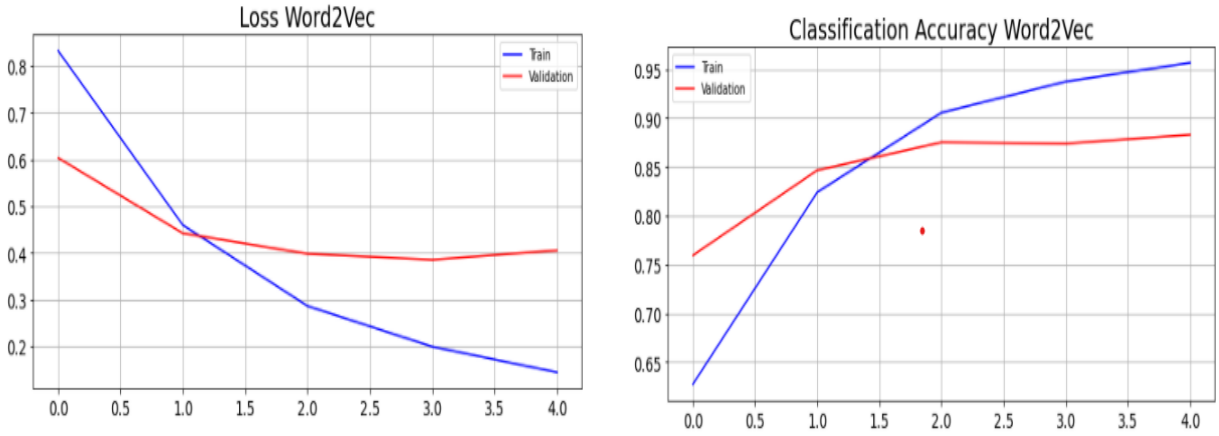


Figure 11: Loss vs Accuracy for Word2Vec Features

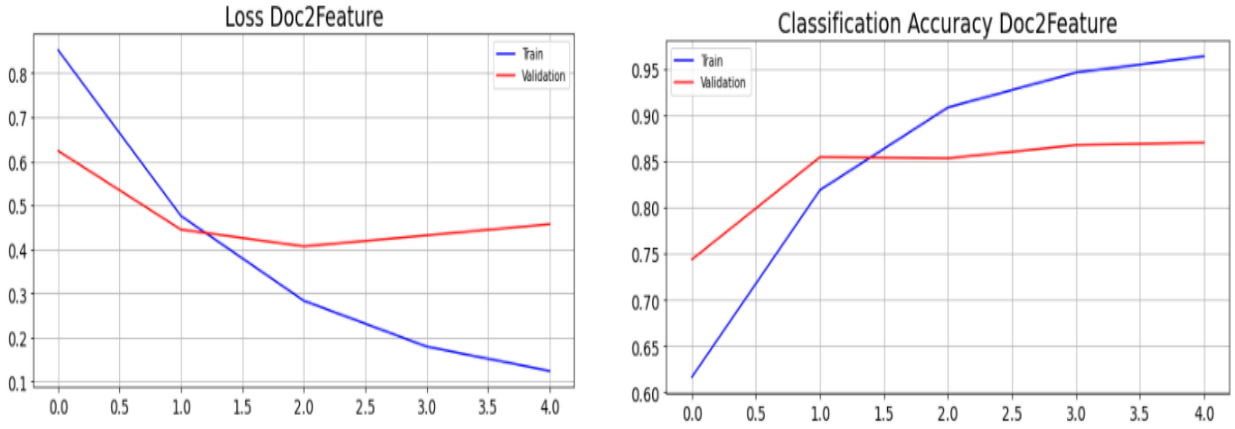


Figure 12: Loss vs Accuracy for Doc2Vec Features

5. CONCLUSION AND FUTURE WORK

Classification, which predicts the target class for each sample in the data, is one of the most important problems in machine learning. Single classifiers are commonly used by researchers to improve performance on available data sets. It is difficult to pick the optimum data mining or machine learning method for a given task. These researchers can produce outstanding results because they employ a range of models to tackle the issue. We looked at classification performance in terms of sensitivity, precision, F1 and accuracy for four different feature extraction sets. We determined that RNN outperforms other machine learning algorithms based on our experimental findings for these four different feature sets. The Bag of Words feature set delivers the slightly better results for all classical machine learning and RNN.

If the dataset grows larger, the sentiment on Twitter improves. In addition, to obtain a sentiment score, we manually level the dataset. If many individuals can conduct the manual leveling for the same tweet, the level will be more accurate in the future. Furthermore, a more complicated RNN model may yield superior results.

REFERENCES

[1] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams engineering journal,* vol. 5, no. 4, pp. 1093-1113, 2014.

[2] N. S. Sattar, S. Arifuzzaman, M. F. Zibran, and M. M. Sakib, "Detecting web spam in webgraphs with predictive model analysis," in *2019 IEEE International Conference on Big Data (Big Data),* 2019: IEEE, pp. 4299-4308.

[3] N. S. Sattar and S. Arifuzzaman, "COVID-19 Vaccination awareness and aftermath: Public sentiment analysis on Twitter data and vaccinated population prediction in the USA," *Applied Sciences,* vol. 11, no. 13, p. 6128, 2021.

[4] M. Roser, H. Ritchie, and E. Ortiz-Ospina, "Coronavirus Disease (COVID-19) https://ourworldindata. org/coronavirus," *Accessed Aug,* vol. 16, 2020.

[5] T. Nabity-Grover, C. M. Cheung, and J. B. Thatcher, "Inside out and outside in: How the COVID-19 pandemic affects self-disclosure on social media," *International Journal of Information Management,* vol. 55, p. 102188, 2020.

[6] R. Ahmed *et al.,* "Deep neural network-based contextual recognition of arabic handwritten scripts," *Entropy,* vol. 23, no. 3, p. 340, 2021.

[7] F. Barbieri and H. Saggion, "Automatic Detection of Irony and Humour in Twitter," in *ICCC*, 2014, pp. 155-162.

[8] W. He, H. Wu, G. Yan, V. Akula, and J. Shen, "A novel social media competitive analytics framework with sentiment benchmarks," *Information & Management,* vol. 52, no. 7, pp. 801-812, 2015.

[9] M. J. Widener and W. Li, "Using geolocated Twitter data to monitor the prevalence of healthy and unhealthy food references across the US," *Applied Geography*, vol. 54, pp. 189-197, 2014.

[10] Z. Saeed *et al.*, "What's happening around the world? a survey and framework on event detection techniques on twitter," *Journal of Grid Computing*, vol. 17, no. 2, pp. 279-312, 2019.

[11] Z. Saeed, R. A. Abbasi, I. Razzak, O. Maqbool, A. Sadaf, and G. Xu, "Enhanced heartbeat graph for emerging event detection on twitter using time series networks," *Expert Systems with Applications*, vol. 136, pp. 115-132, 2019.

[12] U. Naseem, I. Razzak, M. Khushi, P. W. Eklund, and J. Kim, "Covidsenti: A large-scale benchmark Twitter data set for COVID-19 sentiment analysis," *IEEE Transactions on Computational Social Systems*, 2021.

[13] A. B. Nassif, A. Elnagar, I. Shahin, and S. Henno, "Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities," *Applied Soft Computing*, vol. 98, p. 106836, 2021.

[14] P. Mehta and S. Pandya, "A review on sentiment analysis methodologies, practices and applications," *International Journal of Scientific and Technology Research*, vol. 9, no. 2, pp. 601-609, 2020.

[15] D. S. Chauhan, R. Kumar, and A. Ekbal, "Attention based shared representation for multi-task stance detection and sentiment analysis," in *International Conference on Neural Information Processing*, 2019: Springer, pp. 661-669.

[16] A. C. Sanders *et al.*, "Unmasking the conversation on masks: Natural language processing for topical sentiment analysis of COVID-19 Twitter discourse," *medRxiv*, p. 2020.08. 28.20183863, 2021.

[17] T. Pano and R. Kashef, "A Complete VADER-Based Sentiment Analysis of Bitcoin (BTC) Tweets during the Era of COVID-19," *Big Data and Cognitive Computing*, vol. 4, no. 4, p. 33, 2020.

[18] K. K. Bhagat, S. Mishra, A. Dixit, and C.-Y. Chang, "Public Opinions about Online Learning during COVID-19: A Sentiment Analysis Approach," *Sustainability*, vol. 13, no. 6, p. 3346, 2021.

[19] C. Villavicencio, J. J. Macrohon, X. A. Inbaraj, J.-H. Jeng, and J.-G. Hsieh, "Twitter Sentiment Analysis towards COVID-19 Vaccines in the Philippines Using Naïve Bayes," *Information*, vol. 12, no. 5, p. 204, 2021.

[20] A. A. Chaudhri, S. Saranya, and S. Dubey, "Implementation Paper on Analyzing COVID-19 Vaccines on Twitter Dataset Using Tweepy and Text Blob," *Annals of the Romanian Society for Cell Biology*, pp. 8393-8396, 2021.

[21] E. M. Glowacki, G. B. Wilcox, and J. B. Glowacki, "Identifying# addiction concerns on twitter during the COVID-19 pandemic: A text mining analysis," *Substance abuse*, vol. 42, no. 1, pp. 39-46, 2021.

[22] J. Xue *et al.*, "Twitter discussions and emotions about the COVID-19 pandemic: Machine learning approach," *Journal of medical Internet research*, vol. 22, no. 11, p. e20550, 2020.

[23] T. Chen and M. Dredze, "Vaccine images on Twitter: analysis of what images are shared," *Journal of medical Internet research*, vol. 20, no. 4, p. e8221, 2018.

[24] E. L. Ray *et al.*, "Ensemble forecasts of coronavirus disease 2019 (COVID-19) in the US," *MedRXiv,* 2020.

[25] C. M. Liapis, A. Karanikola, and S. Kotsiantis, "An ensemble forecasting method using univariate time series COVID-19 data," in *24th Pan-Hellenic Conference on Informatics*, 2020, pp. 50-52.

[26] A. Hussain *et al.*, "Artificial intelligence–enabled analysis of public attitudes on facebook and twitter toward covid-19 vaccines in the united kingdom and the united states: Observational study," *Journal of medical Internet research,* vol. 23, no. 4, p. e26627, 2021.

[27] M. Rafiuzaman, "Forecasting chaotic stock market data using time series data mining," *International Journal of Computer Applications,* vol. 101, no. 10, 2014.

[28] B. Krollner, B. J. Vanstone, and G. R. Finnie, "Financial time series forecasting with machine learning techniques: a survey," in *ESANN*, 2010.

[29] N. Zhang, A. Lin, and P. Shang, "Multidimensional k-nearest neighbor model based on EEMD for financial time series forecasting," *Physica A: Statistical Mechanics and its Applications,* vol. 477, pp. 161-173, 2017.

[30] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, and B. Seaman, "Sales demand forecast in e-commerce using a long short-term memory neural network methodology," in *International conference on neural information processing*, 2019: Springer, pp. 462-474.

[31] B. M. Pavlyshenko, "Machine-learning models for sales time series forecasting," *Data,* vol. 4, no. 1, p. 15, 2019.

[32] G. Papacharalampous, H. Tyralis, and D. Koutsoyiannis, "Univariate time series forecasting of temperature and precipitation with a focus on machine learning algorithms: A multiple-case study from Greece," *Water resources management,* vol. 32, no. 15, pp. 5207-5239, 2018.

[33] R. Medar, A. B. Angadi, P. Y. Niranjan, and P. Tamase, "Comparative study of different weather forecasting models," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017: IEEE, pp. 1604-1609.

[34] A. Gonzalez-Vidal, F. Jimenez, and A. F. Gomez-Skarmeta, "A methodology for energy multivariate time series forecasting in smart buildings based on feature selection," *Energy and Buildings,* vol. 196, pp. 71-82, 2019.

[35] M. A. P. Mary, "Classifying Future Scope in Energy Resources and Predicting Power Demand using Multilayer Perceptron," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 2332-2339, 2021.

[36] T. Usha and S. A. A. Balamurugan, "Seasonal based electricity demand forecasting using time series analysis," *Circuits and Systems,* vol. 7, no. 10, pp. 3320-3328, 2016.

[37] S. Humeau, T. K. Wijaya, M. Vasirani, and K. Aberer, "Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households," in *2013 Sustainable internet and ICT for sustainability (SustainIT)*, 2013: IEEE, pp. 1-6.

[38] C. X. Ling, J. Huang, and H. Zhang, "AUC: a statistically consistent and more discriminating measure than accuracy," in *Ijcai*, 2003, vol. 3, pp. 519-524.

[39] M. F. Kabir, S. A. Ludwig, and A. S. Abdullah, "Rule discovery from breast cancer risk factors using association rule mining," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018: IEEE, pp. 2433-2441.

[40] J. Roesslein, "tweepy Documentation," *Online] http://tweepy. readthedocs. io/en/v3*, vol. 5, 2009.

[41] E. Loper and S. Bird, "Nltk: The natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[42] A. Kulkarni and A. Shivananda, "Converting text to features," in *Natural language processing recipes*: Springer, 2021, pp. 63-106.

[43] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45-65, 2003.

[44] K. W. Church, "Word2Vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155-162, 2017.

[45] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," *arXiv preprint arXiv:1607.05368*, 2016.

[46] K. Makice, *Twitter API: Up and running: Learn how to build applications with the Twitter API*. " O'Reilly Media, Inc.", 2009.

[47] Y. Wang, J. Callan, and B. Zheng, "Should we use the sample? Analyzing datasets sampled from Twitter's stream API," *ACM Transactions on the Web (TWEB)*, vol. 9, no. 3, pp. 1-23, 2015.

[48] E. Chen, A. Deb, and E. Ferrara, "# Election2020: the first public Twitter dataset on the 2020 US Presidential election," *Journal of Computational Social Science*, pp. 1-18, 2021.

[49] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43-52, 2010.

[50] C.-y. Shi, C.-j. Xu, and X.-J. Yang, "Study of TFIDF algorithm," *Journal of Computer Applications*, vol. 29, no. 6, pp. 167-170, 2009.

[51] B. Li, A. Drozd, Y. Guo, T. Liu, S. Matsuoka, and X. Du, "Scaling word2vec on big corpus," *Data Science and Engineering*, vol. 4, no. 2, pp. 157-175, 2019.

[52] M. F. Kabir and S. A. Ludwig, "Enhancing the performance of classification using super learning," *Data-Enabled Discovery and Applications*, vol. 3, no. 1, pp. 1-13, 2019.

[53] S. M. Rahman, M. F. Kabir, and M. M. Rahman, "Integrated data mining and business intelligence," in *Encyclopedia of Business Analytics and Optimization*: IGI Global, 2014, pp. 1234-1253.

[54] M. F. Kabir and S. A. Ludwig, "Association Rule Mining Based on Ethnic Groups and Classification using Super Learning," *Applied Smart Health Care Informatics: A Computational Intelligence Perspective*, pp. 111-129, 2022.

[55] A. Poornima and K. S. Priya, "A comparative sentiment analysis of sentence embedding using machine learning techniques," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020: IEEE, pp. 493-496.

[56] S. Mandt, M. Hoffman, and D. Blei, "A variational analysis of stochastic gradient algorithms," in *International conference on machine learning*, 2016: PMLR, pp. 354-363.

[57] B. I. Al-Ahmad, A.-Z. Ala'A, M. F. Kabir, M. Al-Tawil, and I. Aljarah, "Swarm intelligence-based model for improving prediction performance of low-expectation teams

in educational software engineering projects," *PeerJ Computer Science,* vol. 8, p. e857, 2022.

[58] T.-K. An and M.-H. Kim, "A new diverse AdaBoost classifier," in *2010 International conference on artificial intelligence and computational intelligence*, 2010, vol. 1: IEEE, pp. 359-363.

[59] H. Raza, D. Rathee, S.-M. Zhou, H. Cecotti, and G. Prasad, "Covariate shift estimation based adaptive ensemble learning for handling non-stationarity in motor imagery related EEG-based brain-computer interface," *Neurocomputing*, vol. 343, pp. 154-166, 2019.