Hindawi
Journal of Sensors
Volume 2022, Article ID 6287559, 10 pages
https://doi.org/10.1155/2022/6287559

Research Article

Evolution and Evaluation: Sarcasm Analysis for Twitter Data Using Sentiment Analysis

Monika Bhakuni,[1] Karan Kumar,[2] Sonia,[1] Celestine Iwendi,[3] and Avtar Singh[4]

[1]Yogananda School of Artificial Intelligence, Computer and Data Science, Shoolini University, Solan, India
[2]Electronics and Communication Engineering Department, Maharishi Markandeshwar Engineering College, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala-133207, India
[3]School of Creative Technologies, University of Bolton, Bolton BL3 5AB, UK
[4]Department of Electronics and Communication Engineering, Adama University of Science and Technology, Adama, Ethiopia

Correspondence should be addressed to Karan Kumar; karan.170987@gmail.com, Sonia; soniacsit@yahoo.com, and Avtar Singh; avtarsingh@astu.edu.et

Received 28 July 2022; Accepted 29 August 2022; Published 11 October 2022

Academic Editor: Sweta Bhattacharya

Copyright © 2022 Monika Bhakuni et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper addresses the evolution and evaluation of sarcasm in textual form. The growing popularity of social networking sites is well known, and every individual generates a whole new set of opinions in form of blogs, microposts, etc. Sentiment analysis is one of the fastest evolving aspects of artificial intelligence categorizing opinions under positive, negative, or neutral sentiments. One such part of sentiment analysis is sarcasm. Sarcasm is becoming a common phenomenon in networking sites where expressing murky feelings wrapped by positive words for conveying contempt is highly used, making it difficult to understand the actual meaning of a statement. When reading customer reviews or complaints, it might be helpful to understand the consumers' genuine intentions in order to enhance the efficiency of customer support or after-sales services. In this paper, different classifiers—decision tree, Naïve Bayes, k-nearest, and support vector machine are used to predict a statement under the category sarcastic or nonsarcastic using tweeter data; the following proposed methodology is used for the experimental evaluation concluding that the given classifiers SVM gains the highest accuracy of 93%, whereas Naïve Bayes and decision tree are performing well with an accuracy of 83% and 86%, respectively, along with the lowest of 51% attained by KNN.

1. Introduction

One of the leading aspects of artificial intelligence is natural language processing (NLP); as easy to say, it is otherwise to interpret currently the most enduring and apprehensive study worldwide in understanding opinions, particularly sentiments of an individual. Specific nouns used to elaborate opinions depending on the case scenarios are one of the major research areas. In recent years, the world has encountered a lot of recurrent changes, wherein the process of being upfront on words is well justify by social networking sites. Unlike the traditional anonymous survey or questionnaire, online posts, interactions, reviews, and media offer, it is more efficient and accurate insight into the minds of people around the world [1]. The growing popularity of Twitter, Facebook, and Instagram has deluded the era with a lot of opinions saying positive, negative, and neutral statements. Just as artificial intelligence is becoming more sophisticated and the internet more accessible, so too is the ability to observe human behavior [1].

Over the recent years, ML and AI are the new hot topics of this era; the formulated approaches are a part of a whole new categorizing of products and appliances [2]. Machine learning algorithms can be classified in four types: supervised, semisupervised, reinforcement, and unsupervised learning. Supervised learning is a kind of machine learning that has the capability to construct a function from a labeled set. Because of the presence of the labeled output value, supervised learning can construct a decent model. This can be happening because the expected results which need to be processed by the model are already provided in the training dataset [3].

Sentiments are defined as an opinion held or expressed by different people at the same time about a particular topic or otherwise. Sentiment analysis can be said as a process of collecting and analyzing data based on human feeling, reviews, or thoughts [4]. The applications for the following can be stated as monitoring the social media, product analytics, customer analysis, business analytics, etc. Sentiment analysis in the real world is becoming increasingly pivotal, specifying that it is domain-centred, i.e., results of one domain cannot be applied to another domain [4]. Another synonym that can be used is opinion mining, where a speaker speaks about a particular entity and discusses its feedback. The growth of information available in social media makes sentiment analysis more crucial [5]. There are a few key challenges faced by sentiment analysis such as entity-named recognition, anaphora recognition, parsing, sarcasm detection, and many others. From the commercial perspective, sentiment analysis can provide online advice and recommendations for both the customers and merchants [5].

Sarcasm is defined as a mode of paradoxical wit depending on its effect on bitter and often ironic language that is usually directed towards an individual. In this era, people are directing their common way of speaking towards sarcasm, and a lot of sarcasm is stated in order to subjectify a particular topic. In essence, sarcasm can be said to as the new way of expressing an opinion. It can be expressed via speech, text, etc. Understanding sarcasm in speech, it becomes interpretable because a lot of gestures are associated with sarcasm, use of a facial expression, tone, and gestures can be used to identify sarcasm. In textual form, it is marginally difficult to interpret sarcasm; dealing with just a set of words becomes slightly difficult to interpret. Detection of sarcasm is one of the leading areas of research, understanding the true opinion of a person under sarcastic statements. The application of the following can be stated as marketing research, opinion mining, and information categorizing, also benefiting areas of interest in NLP.

What makes the task of detecting sarcasm hard is that it is hard to understand human emotion, sometimes without prior knowledge of the topic. Sarcasm resembles lying in some context, making it a more problematic and a hard task [6]. To understand one's intent in the text, we need to classify sarcasm, and it is important to devise a system that could generate a good and reliable training set for the classifier, a labeled bag of words, and an algorithm that could detect sarcasm [7]. There are, however, various other challenges that are posed by streaming data from social media itself. [8]. Sarcasm detection plays a vital role in the company's feedback where they can analyze customers' true intentions about their product.

The following work is proposed in this paper: sentiment analysis utilising natural language processing, followed by sarcasm detection. The paper addresses the concurrent need for sarcasm detection making it useful for knowing the intent of people. Understanding the intents and actual ideas of customers while reading their reviews or complaints also aids in improving the effectiveness of after-sales service or consumer assistance. The existing system is based on four classifiers for prediction; the classifiers are supervised categorical classification, which results in the detection of a statement under the categories sarcastic and nonsarcastic. The data are processed under the specification needed for the classifier to analyze efficiently. The data preprocessing is done using different libraries from the natural language tool kit (nltk), summarizing the results useful for the prediction and further analysis. The classifiers, Naïve Bayes, decision tree, SVC, and KNN are trained and tested for twitter data, which is scraped using twint library, and the detection of sarcasm is hence proposed.

The main points of the initial research are basically as follows:

(i) The current system uses the several supervised classifiers stated above to train and evaluate a model for the prediction of sarcastic and nonsarcastic comments

(ii) Twint, a Twitter scraping tool, is used to start gathering data and collect 10,000 tweets in total for analysis

2. Literature Survey

Dharmavarapu and Bayana [9] proposed a methodology for constructing reliable and effective algorithms for sarcasm detection on Twitter. The output divides the given list of tweets into sarcastic and nonsarcastic tweets and using sentiment analysis to organize the tweets into positive, negative, or neutral tweets by the probabilities. Naïve Bayes algorithm is used for the classification of tweets, wherein AdaBoost is used to determine the polarity of the same. The use of only two classifiers is done to predict the same making it vulnerable for counter classifiers interpretation.

An eminent reseacher in the field [10] define sarcasm's effects on sentiment analysis; the effect of sarcasm scope on the polarity of the tweets is classified in their study and also mentions the rules that can be used for sentiment analysis incorporated with sarcasm performing with higher accuracy. GATE is their developed hashtag for tokenizing, so that sentiment and sarcasm found within hashtags can be detected more easily. The following are their result for their classification; the hashtag tokenization achieves a precision of 98%, while the sarcasm detection achieved 91% precision. The study was published in 2014, and since then, many recurring changes have been seen in terms of sarcasm covenants, necessitating the urgent need for a new, generalized approach.

"Sentiment Analysis for Sarcasm Detection on streaming short text data" by Prasad et al. [8], has presented the counter problem with social media dataset, known as short text data, i.e., use of short forms and slang along with usage of sarcasm. The paper compares different classification algorithms for the detection of sarcastic tweets, use of random forest, gradient boosting, decision tree, adaptive boost, logistic regression, and Gaussian Naïve Bayes for the twitter

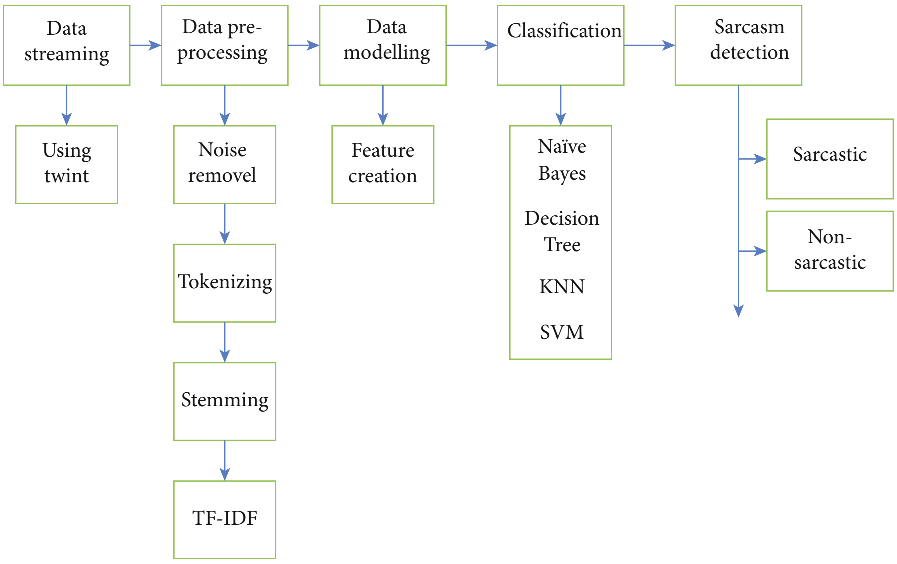


Figure 1: Flowchart representing the proposed methodology.

Table 1: Feature creation.

Fields	Data type	Range
Tweets	Object	
Polarity	Float	-1 to +1
Subjectivity	Float	0 to 1

streaming API with the highest accuracy of 81.82% of gradient boost for results of testing for a split of 60 : 40. Their paper finally concludes with a way of improving the existent sarcasm detection algorithm. Validation on only a 2000 tweet dataset that includes general tweets with sarcastic or nonsarcastic labels serves as the dataset provided for the proposed approach.

"Opinion Mining in Twitter–Sarcasm Detection" [6] by Parveen et al. has made a presentation that the work of impact created by the presence of sarcasm using different components of the tweet. With the use of two datasets, i.e., before adding sarcastic tweets and after adding sarcastic tweets, they have incorporated three different classifiers: Naïve Bayes, maximum entropy, and support vector machine for the impact evaluation of sarcasm-related features on sentiment classification. The results concluded an enhancement after the involvement of sarcasm related features, signifying that the polarity of a tweet was misread due to the presence of sarcasm. The state-of-the-art approaches to sentiment analysis, however, perform less well in Twitter than they do when they are applied to larger texts because of the character limit (140 characters per tweet) and the usage of informal language.

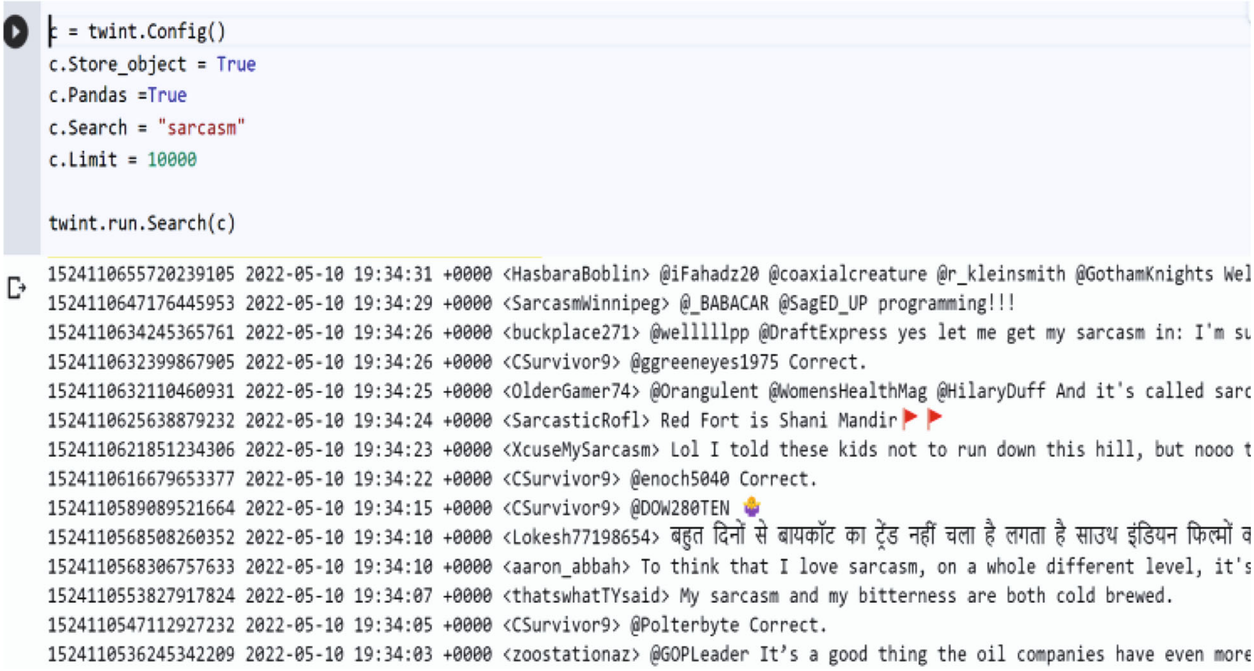
Sindhu and Vadivu's "A Comprehensive Study on Sarcasm Detection Technique in Sentiment Analysis" [7] has covered numerous methodologies and procedures used in sentiment analysis to identify sarcasm in text; data used in the following work are Amazon product reviews. The biggest numbers of model implementations are obtained through Twitter API. Detection of sarcasm is done using classifiers and rule-based methods. Using the SVM, an accuracy of 54.1% achieved.

"A pattern-based Approach for Sarcasm Detection on Twitter" [11] by Bouazizi and Ohtsuki presents the work of a pattern-based approach for the detection of sarcasm and also finds the effectiveness of the model created for sarcasm detection. Data are retrieved from Twitter API. Four sets of feature extraction are used which include as a witticism, a whimper, a form of evasion, and sarcasm. Sentiment-related features, punctuation-related features, lexical and syntactic features, and pattern-related features are employed to classify texts, and their suggested strategy achieves an accuracy of 83.1%.

Several reasearchers [12] use the concept of a supervised machine learning-based approach to defining sarcasm detection on Facebook, concentrating on both post contents (such as text or images) and Facebook users' interactions with those posts. Data collection was done using Facebook graph API. Public 10 pages were selected for the collection of the sarcastic post. Machine learning classifiers were used for the analysis, where random forest and SVM performed better than the rest.

Khare et al. [13] proposed a methodology for analysis of sentiment for government bodies in their paper "Sentiment Analysis And Sarcasm Detection Of Indian General Election Tweets". Analysis is done on Twitter data of duration 2019 where a collection of tweets is done for Lok Sabha election. The textual information contained in tweets is handled using an SVM classifier. The authors have written about a subject where some users tweet in jest. They have achieved an accuracy of 84 percent when comparing their model outcome to the results of the election, which is sufficient for transfer learning. Moreover, the dataset used for the following methodology is from the data science website Kaggle; the full dataset of tweets linked to the election is accessible.

Ashwitha et al. [14] later studied sarcasm detection using natural language processing, covering the striking properties



```
c = twint.Config()
c.Store_object = True
c.Pandas =True
c.Search = "sarcasm"
c.Limit = 10000

twint.run.Search(c)
```

```
1524110655720239105 2022-05-10 19:34:31 +0000 <HasbaraBoblin> @iFahadz20 @coaxialcreature @r_kleinsmith @GothamKnights Wel
1524110647176445953 2022-05-10 19:34:29 +0000 <SarcasmWinnipeg> @_BABACAR @SagED_UP programming!!!
1524110634245365761 2022-05-10 19:34:26 +0000 <buckplace271> @welllllpp @DraftExpress yes let me get my sarcasm in: I'm su
1524110632399867905 2022-05-10 19:34:26 +0000 <CSurvivor9> @ggreeneyes1975 Correct.
1524110632110460931 2022-05-10 19:34:25 +0000 <OlderGamer74> @Orangulent @WomensHealthMag @HilaryDuff And it's called sarc
1524110625638879232 2022-05-10 19:34:24 +0000 <SarcasticRofl> Red Fort is Shani Mandir🚩🚩
1524110621851234306 2022-05-10 19:34:23 +0000 <XcuseMySarcasm> Lol I told these kids not to run down this hill, but nooo t
1524110616679653377 2022-05-10 19:34:22 +0000 <CSurvivor9> @enoch5040 Correct.
1524110589089521664 2022-05-10 19:34:15 +0000 <CSurvivor9> @DOW280TEN 🤷
1524110568508260352 2022-05-10 19:34:10 +0000 <Lokesh77198654> बहुत दिनों से बायकॉट का ट्रेंड नहीं चला है लगता है साउथ इंडियन फिल्मों व
1524110568306757633 2022-05-10 19:34:10 +0000 <aaron_abbah> To think that I love sarcasm, on a whole different level, it's
1524110553827917824 2022-05-10 19:34:07 +0000 <thatswhatTYsaid> My sarcasm and my bitterness are both cold brewed.
1524110547112927232 2022-05-10 19:34:05 +0000 <CSurvivor9> @Polterbyte Correct.
1524110536245342209 2022-05-10 19:34:03 +0000 <zoostationaz> @GOPLeader It's a good thing the oil companies have even more
```

Figure 2: Scrapping Twitter data.

of satire that affects the social and personal relationship; the authors proposed that sarcasm evaluation bridges the gap between mutual communication of machines and humans. The work is based on four approaches lexicon, pattern, machine learning, and context-based. The project's goal is to demonstrate how current technology may be used to tackle social issues and barriers to free speech. The accuracy gained by their work is 96%.The key difference can be stated as utilising a hyperbolic feature set.

3. Materials and Methods

In this proposed work as in Figure 1, there are four categories that implement in accordance with the desired result: (a) collection of data; (b) preprocessing of the data; (c) feature creation; (d) sarcasm detection. Feature Creation are mentioned here with in Table 1.

3.1. Data Collection. Prior to any analysis, the collection of valid data is one of the prominent tasks in the evaluation of any subject. Validity of the data affects the whole process of analysis, and the collection of unbiased data that is wholly transparent and builds a bride in understanding the sentiments in this case.

Twitter intelligence tool or twint is an advanced Twitter scraping tool in python that is used for scraping tweets from Twitter without Twitter API being used. The work in the following uses the following tool in order to collect data from Twitter with the keyword "sarcasm". A total of 10,000 tweets are collected using twint, and the processing on the following is done.

Figure 2 shows the full description of how data are gathered using twint in the following research. The code used can be further generalized for any keyword, wherein here, it is specified for "sarcasm". The comparison of present work with other eminent researchers in this field has been summarized in Table 2.

10013 entries; 0 to 10012.

Data columns (total 38 columns).

3.2. Data Preprocessing. Data preprocessing is a technique used in data mining to change data into information that is more suitable for work. In the data preprocessing, the input data is first taken, and hashtags are located. These hashtags are eliminated from the data entry [9, 15]. The following module includes field selection, data cleaning including noise removal, tokenization, and stemming. The processing of the data gathered is done in the following way.

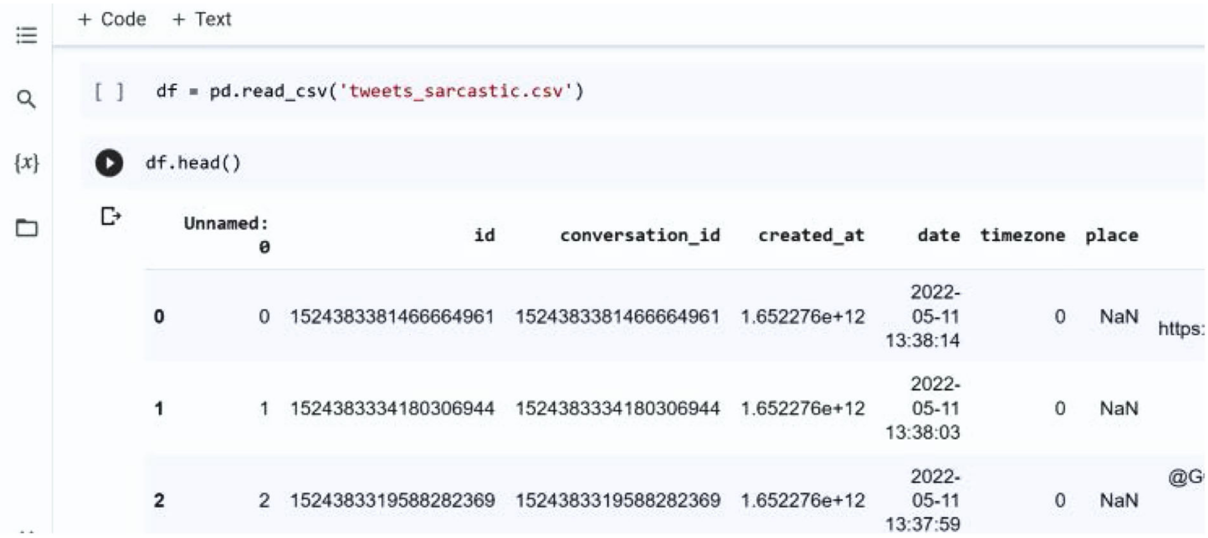
3.2.1. Desired Column Selection. Column selection is one of the major steps in processing the data. A greater impact is made only when the primary column that needs to be processed is used other than subjectify the whole dataset. Since the data collected comes with fields that may not necessarily be used in the processing, and hence it becomes vital for selecting a major column for the study.

Dataset here contains 38 columns mentioning id, tweets, hashtags, cashtags, usr_id, usr_id_name, etc. Before beginning the classification stage, several of the fields in this collection of Twitter data need to be processed. Work completed is specifically bound within the language parameter, i.e., English. Before processing the data, only those tweets are considered with specified language English, making the dataset specific to over 8000 rows. The area of concern is tweets, and hence all the other fields are dropped in

Table 2: Comparison with other work.

Authors	Sarcasm dataset	Classifiers used							Accuracy gained
		LR	SVM	RF	KNN	DT	NB	AB	
[8]	Twitter	•		•		•	•	•	DT: 71.05 NB: 75.18 RF: 77.94 AB: 76.06 LR: 32.03
[13]	Twitter: general election		•						SVM: 80
[19]	Twitter	•		•			•		NB: 57 LR: 80 RF: 80.5
[12]	Facebook		•				•		NB: 73.66 SVM: 88.3
[1]	Twitter	•	•						SVM: 79 LR: 80
[20]	Twitter		•	•	•				SVM: 77.9 KNN: 58 RF: 81

LR: Logistic regression, SVM: Support vector machine, RF: Random forest, KNN: K-nearest neighbor, DT: Decision tree, NB: Naïve Bayes, AB: AdaBoost.

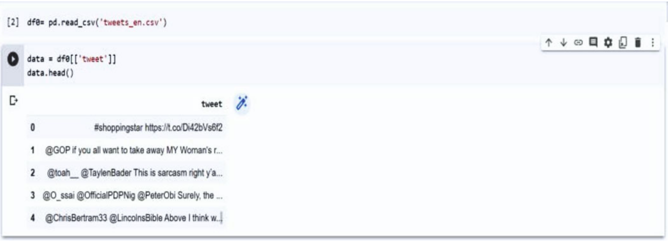


```
[ ]  df = pd.read_csv('tweets_sarcastic.csv')

     df.head()
```

```
   Unnamed:                   id      conversation_id    created_at        date  timezone  place
          0
0         0  1524383381466664961  1524383381466664961  1.652276e+12       2022-         0    NaN  https:
                                                                          05-11
                                                                       13:38:14
1         1  1524383334180306944  1524383334180306944  1.652276e+12       2022-         0    NaN
                                                                          05-11
                                                                       13:38:03
2         2  1524383319588282369  1524383319588282369  1.652276e+12       2022-         0    NaN  @G
                                                                          05-11
                                                                       13:37:59
```

Figure 3: Data frame of Twitter data.



```
[2]  df0= pd.read_csv('tweets_en.csv')

     data = df0[['tweet']]
     data.head()
```

```
                                              tweet
0               #shoppingstar https://t.co/Di42bVs6f2
1   @GOP if you all want to take away MY Woman's r...
2    @toah__ @TaylenBader This is sarcasm right y'a...
3  @O_ssai @OfficialPDPNig @PeterObi Surely, the ...
4   @ChrisBertram33 @LincolnsBible Above I think w...
```

Figure 4: Data frame with language specification.

Table 3: Results obtained by different classifiers.

Classifier	Precision		Recall		F1-score	
	0	1	0	1	0	1
Decision tree	0.83	0.90	0.90	0.83	0.86	0.86
Naïve Bayes	0.78	0.90	0.90	0.77	0.84	0.83
KNN	0.49	0.98	1.0	0.07	0.66	0.14
SVM	0.89	0.98	0.98	0.89	0.93	0.93

Table 4: Accuracy for different classifiers.

Classifier	Accuracy
Decision tree	0.86
Naïve Bayes	0.83
KNN	0.51
SVM	0.93

the table, and a new data frame is made with just the column recognized as "tweets".

Figures 3 and 4 describe the words stated above, where the only concerned data field, i.e., tweets are considered, and a new dataframe is made which will commence the further processing. Further Table 3 and Table 4 are summarizing the results and accuracy level obtained through different classifiers.

8174 entries; 0 to 8173.

Data columns (total 1 column).

3.2.2. Data Cleaning. Data cleaning is a process of removing incompetent data and making the data considerably informative for the desired study. Removing all the unilluminating data from the dataset for the desired output is the major concern of data cleaning. Since the data contains a lot of special symbols, removal of all the same is required. One common library in python that supports the cleaning of data is related to regular expression, named as "re". Following is an example of data cleaning in the study.

Uncleansed Text:Hey-Can I ask some of my fans on #twitter to #subscribe to my #YouTube channel? Where I got none! #podcast #socialmedia #sarcasm #NFL

Clean Text : Hey- Can I ask some of my fans on twitter to subscribe to my YouTube channel? Where I got none! Podcast socialmediasarcasmNFL.

(1) Noise Removal. One of the factors for text analysis is noise removal. In text classification, it is vital to make data apprehensively beneficial in favour of the study. In order to achieve maximum output, processes are applied to data for the utmost results. A process for removing characters' digits, URLs, stop words, punctuation, piece of text, etc., from the text is noise removal. The cleansed data is further used for the next phase.

In the work commenced, following Figure 5 is the example of noise removal, where the unwanted information depending on the goal of the project is done. Table 5 further describes the Accuracy Test for it.

Uncleansed text:@jackpuckering93 @ramsfanryan @Boro Surely suggesting someone 'enjoys a FFP charge' is sarcasm, not irony. Anyway, Jack, enjoy the Champo.

Cleansed text: Surely suggesting someone enjoys a FFP charge is sarcasm not irony Anyway Jack enjoy the Champo

(2) Stop Words Removal. Stop words can be defined as words that are commonly used in the English language. These words are removed as they are classified as nonuseful words and take up space in database; therefore removal of these words is preferred for analysis.

Some common stopwords are ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his'] etc.

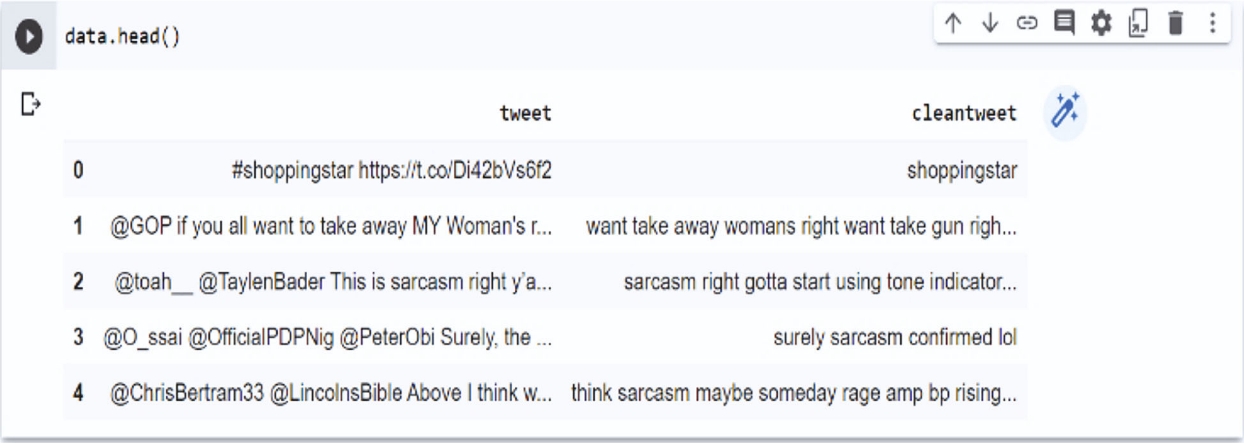
In preprocessing, stop words are removed for the flexibility for the processed analysis; here is the output gained after removing these stop words from the column "tweets" to insure better classification.

Un-cleansed Data : Our brain is at smooth it cant even detect sarcasm

*Cleansed Data:*brain smooth cant even detect sarcasm

(3) Tokenization. One of the aspects of text processing is tokenization, dividing the text into smaller sections known as tokens with the use of delimiters. It is one of the main features of lexically analyzing the text [16]. Tokenization is performed on tweets to break them down into perfect meaningful modules from a sentence [8]. These tokens are further used as vocabulary in traditional NLP using count vectorize and TF-IFD. The division of data is further used in the analysis.

The following example stated below shows how data is tokenized done for in the tweets, considering the tweets, and following Figure 3 shows the code used to tokenize to make



```
data.head()
```

	tweet	cleantweet
0	#shoppingstar https://t.co/Di42bVs6f2	shoppingstar
1	@GOP if you all want to take away MY Woman's r...	want take away womans right want take gun righ...
2	@toah__ @TaylenBader This is sarcasm right y'a...	sarcasm right gotta start using tone indicator...
3	@O_ssai @OfficialPDPNig @PeterObi Surely, the ...	surely sarcasm confirmed lol
4	@ChrisBertram33 @LincolnsBible Above I think w...	think sarcasm maybe someday rage amp bp rising...

Figure 5: Data after noise removal.

Table 5: Accuracy test.

Statement	Prediction using SVM
Trying some holiday at the office!	1
I am not busy but ttyl!	1
Ttyl	0

the sentence more meaningful in accordance with the analysis.

Uncleansed Data: `life peronal pleasure every moment day life suffering love matter uncomfortable fear self indulgence weakness character become hate.`

Cleansed Data: `['life', 'personal', 'pleasure', 'every', 'moment', 'day', 'life', 'suffering', 'love', 'matter', 'uncomfortable', 'fear', 'self',`

(4) Stemming. Stemming is another important aspect of natural language understanding, reducing the word to its stem making it viable in reducing the vocabulary and summarizing different words to their roots for input making it easier for the analysis. The main aim of this is to reduce the repetition of words by dropping the suffix of the word to arrive at the basic form of the word [16].

Utilization of stemming is done in the commenced work by reducing the word to its stem so that the vocabulary is reduced.

Figure 6 shows the words stemmed to their root and the data further ready for analysis.

(5) Term Frequency Inverse Document Frequency. The significance of a word (term) to a document within the corpus is quantified by the TF-IDF statistic [17]. In text summarization and classification software, TF-IDF is frequently used to end filtering words. Additionally, it is employed to enhance a word's frequency in a document proportionally. Inverse term frequency-document frequency (TF-IDF) is a part of information retrieval [16].

A numeric static concluding the importance of a word in the collection, further it is used in the work to detect the word occurrence and its importance. The following image as shown in Figure 7 shows the length of frequency. The code of the following is also mentioned in the same.

3.3. Feature Creation. After the data is fully cleansed, feature creation for further analysis is done. New features promoting the analysis are created. This feature helps in developing a data frame with useful fields that is feasible for further analysis.

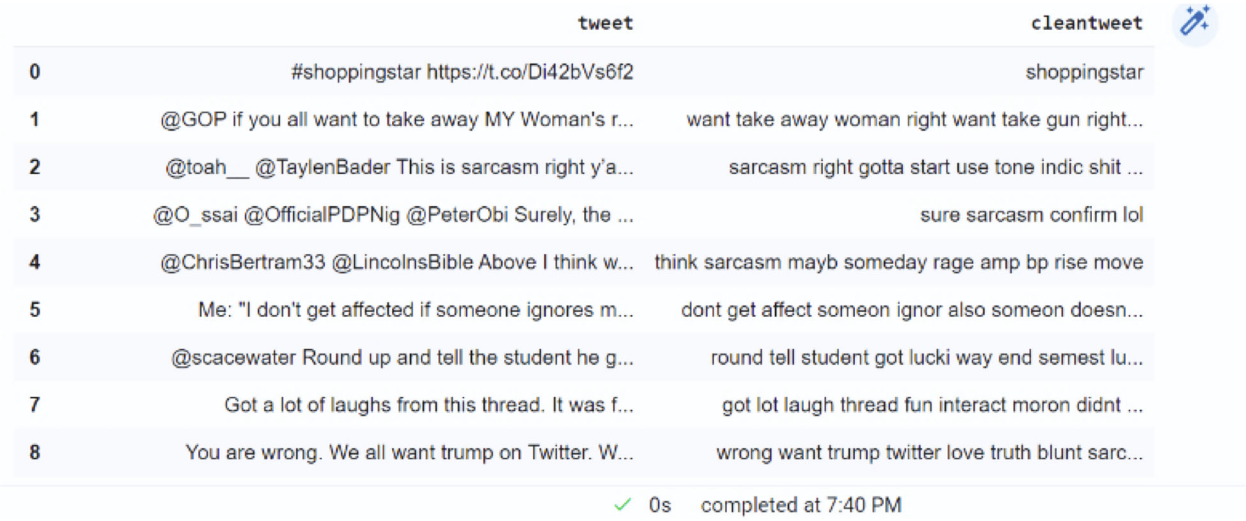
Polarity classification, a fundamental component of sentiment analysis, examines whether an opinion on a certain trait or facet of a target is expressed in a document or a sentence. [5]. The new fields added are polarity and subjectivity in the respective study, whereas another research direction is subjectivity or objectivity identification [5]. Positive, negative, and neutral feelings are the three categories used to categorize sentiment analysis. Now that it is wise to know the polarity of a particular statement, different interpretations can be made in reference to the polarity, and moving forward is subjectivity; it defines whether a statement is under any subject. With the use of the text blob, polarity and subjectivity are classified.

Range Index: 8174 entries; 0 to 8173.

Data columns (total 3 columns):

3.4. Sarcasm Detection. Sarcasm and humor are key human characteristics and one of the largest gaps that artificial intelligence must bridge as they try to become more humanlike in intuition and behavior [1]. Although there are several algorithms in machine learning that are meant to accomplish precisely that, categorizing text based on its sentiment presents many particular difficulties. These can be summed up in the following query: "What kinds of features do we use? [18]."

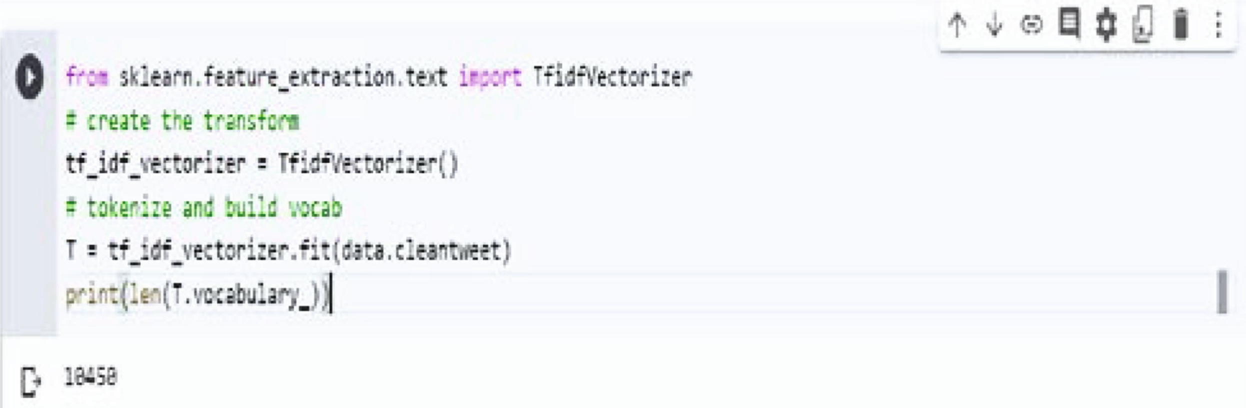
The sarcasm detection in the proposed model is classified using different supervised categorical machine



	tweet	cleantweet
0	#shoppingstar https://t.co/Di42bVs6f2	shoppingstar
1	@GOP if you all want to take away MY Woman's r...	want take away woman right want take gun right...
2	@toah__ @TaylenBader This is sarcasm right y'a...	sarcasm right gotta start use tone indic shit ...
3	@O_ssai @OfficialPDPNig @PeterObi Surely, the ...	sure sarcasm confirm lol
4	@ChrisBertram33 @LincolnsBible Above I think w...	think sarcasm mayb someday rage amp bp rise move
5	Me: "I don't get affected if someone ignores m...	dont get affect someon ignor also someon doesn...
6	@scacewater Round up and tell the student he g...	round tell student got lucki way end semest lu...
7	Got a lot of laughs from this thread. It was f...	got lot laugh thread fun interact moron didnt ...
8	You are wrong. We all want trump on Twitter. W...	wrong want trump twitter love truth blunt sarc...

✓ 0s completed at 7:40 PM

Figure 6: Clean data after stop words, tokenizing, and stemming removal.



```
from sklearn.feature_extraction.text import TfidfVectorizer
# create the transform
tf_idf_vectorizer = TfidfVectorizer()
# tokenize and build vocab
T = tf_idf_vectorizer.fit(data.cleantweet)
print(len(T.vocabulary_))
```

```
10450
```

Figure 7: Length of vocabulary after vectorizing.

classification: decision tree, Naïve Bayes, KNN, and support vector machine. Decision tree classifier poses a series of carefully crafted questions about the features that are supplied to the algorithm [8]. Naïve Bayes is a log-linear model; that is, in both cases the probability of a document belonging to a class is proportional.

The model is trained and tested with a ratio of 80 : 20 split of the 8000 tweets collected. Accuracies are gained from different classifiers, and the prediction of a particular statement is done on the bases of the highest accuracy among the four classifiers.

4. Results

4.1. Dataset. Data generation in this era is immense. Data is gathered from different social networking sites. This data can be both informative and noninformative, purely based on the needs. In this work, data are gathered from one such networking site, Twitter, where millions of tweets are generated on a single day moreover a single topic. Since the work is to check whether a statement is sarcastic or nonsarcastic, the use of sarcastic tweets is done to train the model, using the keyword "sarcasm", and data is scrapped out of Twitter and classified as sarcastic or nonsarcastic on the basis of subjectivity after the cleaning of data. A total of 8000 tweets are gathered, with the preferred language English. Classification of sarcastic and nonsarcastic is done using 0 and 1, respectively.

4.1.1. Comparison Table

(1) Experimental Evaluation. The following Figure 8 shows the word cloud that is processed in the work. The use of the python library WordCloud is done for the formation of the figure. As the figure suggests, these are the following words that are majorly encountered in the tweets, specifying that the tweets related to sarcasm revolve around these words, making it efficient for the machine to learn and interpret for sarcasm detection.

The classifier model is trained using the dataset, and this paper authenticates the use of supervised classifiers for natural language processing; the result obtained is mentioned below in the table giving the precision, recall, and F1-score for each classifier. It can be clearly stated that the highest

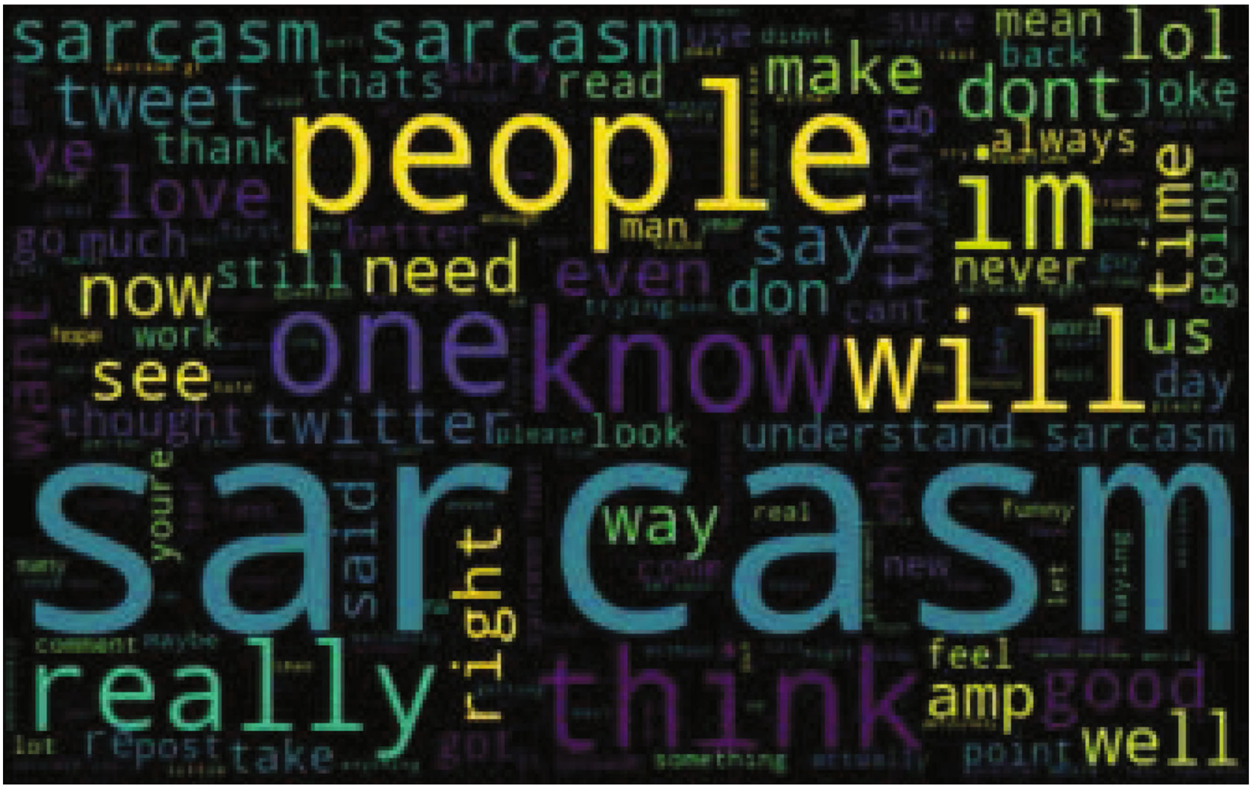


Figure 8: Word cloud.

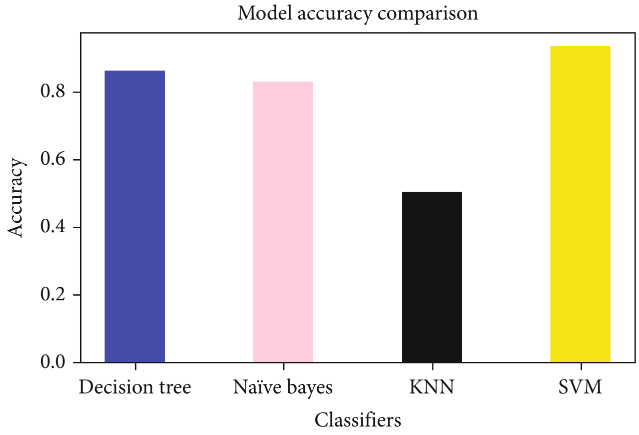


Figure 9: Model accuracy comparison.

accuracy is gained by SVM with 93% wherein the lowest is by KNN. Decision tree is performing greater than that of KNN; similarly, Naïve Bayes with an accuracy of 83% is also a good representative of the data.

4.2. Evaluation Metrics. For this study, the evaluation metrics of recall, precision, F1-score, and accuracy are used. The mathematical definition of these indices are expressed as follows:[21],where TP: true positive, FP: false positive, FN: false negative, and TN: true negative.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{1}$$

$$\text{Precision} = \frac{\text{TN}}{\text{TN} + \text{FN}}, \tag{2}$$

$$\text{F1} - \text{score} = \frac{2\text{TP}}{2\text{TN} + \text{FP} + \text{FN}}, \tag{3}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}. \tag{4}$$

The Figure 9 shows the comparison of model accuracy gained by the different classifiers.

4.3. Analysis. The basic redemption of this work can be classified as the prediction of a statement; all the models are used although known that the highest accuracy is of SVM, testing was done by feeding different statements and evaluating whether the statement was sarcastic or not, using 0 and 1, respectively.

5. Conclusion and Future Work

By now, it is well known to us the diverse nature of data considering sarcasm; well it is not bound to any specification, and thus it becomes a challenge for a machine to interpret the sentiments of an individual, moreover, sarcasm. For a machine to adapt to these recurrent challenges, algorithms need to be processed time and again. Analyzing the sentiment of tweets gives an interesting insight to the opinions of the public about a certain event [6].Therefore; this work is purely based on the aspect of detection of sarcasm using Twitter data, which changes itself repeatedly. By comprehending the intents and true thoughts of customers when reading their feedback or complaints, it also helps to improve the effectiveness of after-sales services or consumer assistance.

This paper aims the classification of sentiments categorized under positive, negative, and neutral sentiments, extending to sarcasm detection. The data collection is done through Twitter, which needs to be preprocessed before any conclusion. Different classifiers are involved in the preceding of our aim. It can be clearly stated that the different supervised algorithms are fit and reliable for the detection of sarcasm.

Future work with traditional machine learning and natural language processing can be used for classifying sarcastic tweets are positive sarcasm and negative sarcasm; this area of research can bring more clarity for a machine to understand sarcasm.

Further, there are an extended versions of sarcasm stated as satire, pun, banter, humor, etc., which can also be classified using the same technique, making machine to understand better and achieve the desired conclusion.

Data Availability

Real-time data has been taken from Twitter and can be available from the author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] L. I. Novic, "A machine learning approach to text-based sarcasm detection," 2022, https://academicworks.cuny.edu/gc_etds/4856Discoveradditionalworksat:https://academicworks.cuny.edu.

[2] M. Razno, "Machine learning text classification model with NLP approach," *Computational Linguistics and Intelligent Systems*, vol. 2, pp. 71–73, 2019.

[3] S. G. Wicana, T. Y. Ibisoglu, and U. Yavanoglu, "A review on sarcasm detection from machine-learning perspective," in *Proceedings - IEEE 11th International Conference on Semantic Computing, ICSC 2017*, pp. 469–476, San Diego, CA, USA, 2017.

[4] P. Garg and V. Bassi, *Sentiment Analysis of Twitter Data Using NLTK in Python*, 2016.

[5] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," *Knowledge and Information Systems*, vol. 60, no. 2, pp. 617–663, 2019.

[6] S. Parveen and S. N. Deshmukh, "Opinion mining in Twitter-sarcasm detection," *Politics*, vol. 1200, p. 125, 2017.

[7] C. Sindhu and G. Vadivu, "Sentiment analysis and opinion summarization of product feedback," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2S4, pp. 59–64, 2019.

[8] A. G. Prasad, S. Sanjana, S. M. Bhat, and B. S. Harish, "Sentiment analysis for sarcasm detection on streaming short text data," in *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, pp. 1–5, London, UK, 2017.

[9] B. D. Dharmavarapu and J. Bayana, "Sarcasm detection in Twitter using sentiment analysis," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 1, 2018.

[10] A. Alsharef, K. Aggarwal, D. Koundal, H. Alyami, and D. Ameyed, "An automated toxicity classification on social media using LSTM and word embedding," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 8467349, p. 8, 2022.

[11] M. Bouazizi and T. O. Ohtsuki, "A pattern-based approach for sarcasm detection on Twitter," *IEEE Access*, vol. 4, pp. 5477–5488, 2016.

[12] A. Alsharef, K. Aggarwal, M. Kumar, and A. Mishra, "Review of ML and AutoML solutions to forecast time-series data," *Archives of Computational Methods in Engineering*, pp. 1–15, 2022.

[13] A. Khare, A. Gangwar, S. Singh, and S. Prakash, "Sentiment analysis and sarcasm detection of Indian general election tweets," 2022, https://arxiv.org/abs/2201.02127.

[14] A. Aswitha, G. Shruthi, H. R. Shruthi, M. Upadhyaya, A. P. Ray, and T. C. Manjunath, "Sarcasm detection in natural language processing," *Materials Today: Proceedings*, vol. 37, Part 2, pp. 3324–3331, 2021.

[15] J. Sharma, M. Arora, Sonia, and A. Alsharef, "An illustrative study on Multi Criteria Decision Making Approach: Analytical Hierarchy Process," *In 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 2000–2005, 2022.

[16] P. Dharwal, T. Choudhury, R. Mittal, and P. Kumar, "Automatic sarcasm detection using feature selection," in *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (ICATccT)*, pp. 29–34, Tumkur, India, 2017.

[17] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright, "Sarcasm detection using machine learning algorithms in Twitter: a systematic review," *International Journal of Market Research*, vol. 62, no. 5, pp. 578–598, 2020.

[18] Y. Mejova, V. Shirsat, and R. S. Jagdale, *Sentiment Analysis: An Overview Comprehensive Exam Paper*, 2009.

[19] T. Jain, N. Agrawal, G. Goyal, and N. Aggrawal, "Sarcasm detection of tweets: a comparative study," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, pp. 1–6, Noida, India, 2017.

[20] N. Pawar and S. Bhingarkar, "Machine learning based sarcasm detection on Twitter data," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 957–961, Coimbatore, India, 2020.

[21] A. Y. Muaad, H. JayappaDavanagere, J. V. B. Benifa et al., "Artificial intelligence-based approach for misogyny and sarcasm detection from Arabic texts," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 7937667, 9 pages, 2022.