Electronics and Electrical Engineering

AIMS Electronics and Electrical Engineering, 6(4): 345–369.
DOI: 10.3934/electreng.2022021
Received: 03 June 2022
Revised: 07 September 2022
Accepted: 09 September 2022
Published: 28 September 2022

http://www.aimspress.com/journal/ElectrEng

*Research article*

# A novel algorithm for sarcasm detection using supervised machine learning approach

**Abdullah Yahya Abdullah Amer[1,*] and Tamanna Siddiqu[2]**

[1] Research Scholar in department Computer science, Aligarh Muslim University, Aligarh, 200201, India
[2] Department of Computer science, Aligarh Muslim University, Aligarh, 200201, India

* **Correspondence:** Email: Ayaamer@myamu.ac.in; Tel: +918792794298.

**Abstract:** Sarcasm means the opposite of what you desire to express, particularly to insult a person. Sarcasm detection in social networks SNs such as Twitter is a significant task as it has assisted in studying tweets using NLP. Many existing study-related methods have always focused only on the content-based on features in sarcastic words, leaving out the lexical-based features and context-based features knowledge in isolation. This shows a loss of the semantics of terms in a sarcastic expression. This study proposes an improved model to detect sarcasm from SNs. We used three feature set engineering: context-based on features set, Sarcastic based on features, and lexical based on features. Two Novel Algorithms for an effective model to detect sarcasm are divided into two stages. The first used two algorithms one with preprocessing, and the second algorithm with feature sets. To deal with data from SNs. We applied various supervised machine learning (ML) such as k-nearest neighbor classifier (KNN), naïve Bayes (NB), support vector machine (SVM), and Random Forest (RF) classifiers with TF-IDF feature extraction representation data. To model evaluation metrics, evaluate sarcasm detection model performance in precision, accuracy, recall, and F1 score by 100%. We achieved higher results in Lexical features with KNN 89.19% accuracy campers to other classifiers. Combining two feature sets (Sarcastic and Lexical) has shown slight improvement with the same classifier KNN; we achieved 90.00% accuracy. When combining three feature sets (Sarcastic, Lexical, and context), the accuracy is shown slight improvement. Also, the same classifier we achieved is a 90.51% KNN classifier. We perform the model differently to see the effect of three feature sets through the experiment individual, combining two feature sets and gradually combining three feature sets. When combining all features set together, achieve the best accuracy with the KNN classifier.

**Keywords:** sarcasm detection; feature engineering; machine learning; text classification; feature sets

## 1. Introduction

The advent of the web and advances in devices and technology have increased social network tools such as Twitter, Instagram, and Facebook to communicate with families and friends [1]. Therefore, an extensive size of social network text constructed day by day demands study and analysis. People on social networks post messages and share, making the information available to the public. Recognizing somebody's subjective data, such as people's sentiments, emotions, and opinions, is made viable by this information. Recognizing somebody's emotions regarding services, events, politics, products, or individuals gets a lot of advantages for any organization. Hence, it has evolved into the essential step of studying somebody's opinions. The opportunity of recognizing personal data is critical; it aids in developing structured information, which acts as a necessary dataset of information for decision-making individuals. Some of the social text content seen on the platforms consists of extended terms of mockery and sarcasm.

Sarcasm is "the use of comments that mean the opposite of what a person says, created to harm a person's feelings or to blame something in a funny way" [2]. The ironic word means a match between a person's reason to make the actual composition and the utterance. For example, the sarcastic utterance "I love to study on holidays!" here illustrates a conflict between the expression "love" and the exact statement "on holidays." The unique form of opinion mining is the rejection and the opinion polarities shift to confirm with a view to sarcasm. The irony is deeply contextual, and as an outcome, some context indications and shifts into contradictory opinions can help in sarcasm classification. In the text data, by defining the ambiguity of the purpose and enhancing the overall feeling classification of extensive size of user's text, dataset got from the social network [3].

The inadequate understanding of the problem "Context" and the clear case will result in detection problems. Contextual knowledge is one of the significant difficult stages of restraint of text content. Sarcasm detection from social networking utilizes NLP approaches to classify datasets for the effects and features of sarcasm. Automated sarcasm classification is the main problem encountered by NLP because it is difficult for humans to accurately classify statements as non-sarcastic or sarcastic. Furthermore, there is a deficiency of accurately labeled sarcastic text data used for training classifiers [4]. Existing research has tried recognizing sarcasm in tweets by utilizing various feature extraction techniques. A literature review on irony detection shows that those current approaches have two major issues [5]. Firstly, the sarcastic-based features, lexical-based on features, and context-based features of the words in representation are neglected. Therefore, various words can possess equal vector representations. Secondly, dataset sparsity within vector representation since each phrase has a word limitation. These problems could make an issue through the models training because some words can be shown in the testing set only while not seen in the training set, creating most of the training features set vectors sparse. Hence, it is essential to detect additional techniques to overcome this weakness and improve predictive implementation in classifying sarcasm. The existing method may have favorable outcomes in some cases to text Twitter dataset as Facebook and Twitter has unstructured raw data; this is effective for detecting sarcasm dataset. Therefore, there is a requirement to take out this study [6].

Using machine-learning algorithms may offer effective sarcasm detection outcomes because building an effective classified model depends on different factors [7]. The key factors are feature sets used during the learning algorithms that associate efficiently with the class representation. The effective features from both sarcastic and non-sarcastic classes need significant measures in creating the classification models. Therefore, the research objective is to develop an effective model for sarcasm detection depending on the feature set training dataset. We propose an efficient feature set

and machine learning classifier model to best perform sarcasm detection in text mining analysis to achieve much better precision. We described the proposed, evaluated, and dissection results during this research paper. The main contributions of this study are provided below:

- First: We used a novel algorithm with feature extraction to extract discriminatory features in two-phase classification algorithms, considering the feature set in the first phase dealing with text data mining fused preprocess in the second phase to sarcasm identification.
- Second, we apply three features: sarcastic-based on features, lexical-based on features, and context-based on features. And four Supervised Machine Learning classifiers to recognize sarcasm from the Twitter dataset to achieve better performances in text study.
- Third: We proposed the correct feature sets, which lead to the best accuracy, shown in our study's achieved effect result analysis dataset.
- Finally: we Analyze outcomes present that KNN (91.84%) outperforms the accuracy of SVM and Na ïve Bayes to the various features set up.

The rest of this study is organized as follows: Section two presents a summarized literature review of the existing work in sarcasm detection. Section three presents materials methods, and includes the proposed system, dataset, data preprocessing, feature set engineering, and classification Models. Section four includes, experimental settings confusion matrix and evaluation measures. Section five contains the results and analysis. In section sex result discussion evaluating effective sarcasm classifier, evaluating effective Feature Set, Accuracy combined Two Feature Sets, Integrate All Features Sets Incrementally, and Conclusion and feature work.

### *1.1. Sarcasm categories*

Sarcasm detection in this platform is an important task as it has been supported in analyzing tweets. With the help of sarcasm detection, organizations could analyze users' attitudes and feelings about their products. Sarcasm (concerned with the usage of irony in order to convey contempt or mock) detection in tweets or other social network tools is one of the issues encountering moderation and statute of social network text contents. Sarcasm is hard to detect, and humans also face difficulty detecting sarcasm because planned ambiguity by utilizing terms is unclear. Present techniques for automated sarcasm classification essentially depend on linguistic and lexical cues. But, certain techniques have given small or no significant enhancement in accuracy prediction [8], [9]. We present efficient and robust systems to detect sarcasm better and enhance text study precision. Sarcasm is utilized for different purposes, which are illustrated within three categories in this study as follows.

#### 1.1.1. Sarcasm as the escape

It associates with the situation when one user wants to provide a correct answer. Therefore, it creates a presence of irony. In this state, the user utilizes unusual words, some basic expressions, perplexing sentences, and slang.

#### 1.1.2. Sarcasm as whimper

The irony is used to show how bothered or hurt the user is. Therefore, it enables one to describe how bad a situation is, utilizing overstatement and very positive emotions to show the negative event.

### 1.1.3. Sarcasm as humor

When utilized as a joke, the irony is funny; the user employs some appropriate kinds of speeches, adopts a distinct tone, or favors exaggerating this when he/she speaks habitually. While social networks platforms, sound tones are trans-made in particular writing kinds: use of quote repetition, uppercase letter terms, ellipsis, exclamation, interjections, and question marks, in addition to some irony-related emoticons [7].

These three categories of sarcasm extracted three feature sets based on the hypothesis discussed above: these feature sets are features sets based on Lexical, features sets based on Sarcastic, and feature sets based on context [10].

## 2. Literature survey

This section shows a literature review of existing research on sarcasm detection concentrating on feature set engineering. The detailed study of feature set engineering techniques used to detect sarcasm classification responsibilities can be seen in a systematic literature survey [11].

Saha S, Yadav J and Ranjan P (2017) [12] have focused on sarcasm detection from Twitter data; in the method, they used Text Blob to preprocess data through the Natural Language Toolkit and Rapid Miner to discover the polarity also subjectivity of data. With SVM classifiers and Na ïve Bayes classifier. Results showed they achieved accuracy in Na ïve Bayes classifier 65.2% and SVM classifiers at 60.1%.

Sharma S and Chakraverty S (2018) [13] have proposed a framework for sarcasm detection through cross-domain allows addition, processing, and storage of tweets data to detect sarcastic content reviews. The method used the Amazon product dataset around 2000 reviews did experiments using Neural Networks and Support Vector Machines to see sarcasm utilizing lexical and context incongruity features. Evaluating Neural Networks and Support Vector Machines classifiers to sarcasm detection single domain.

Wen Z, et al. (2022) [14] An enhanced attention neural model, known as SAAG, was proposed. By introducing sememe knowledge to Chinese words, they enhanced representation learning. A sememe is a fine-grained representation of a word at the minimum level of meaning. To learn the context and background of sarcasm expression at the sentence level, we leverage auxiliary information, such as the news title. In the next step, we construct a dynamic and progressive representation of text expressions. As demonstrated by our proposed approach, our proposed approach outperforms state-of-the-art models on a sarcasm dataset.

Pawar N and Bhingarkar S (2020) [15] have conducted research experimentation as a basic model for detecting sarcasm, the pattern based on the approach suggested utilizing the Twitter dataset. Four feature sets that hold many distinct ironies are proposed, categorizing tweets as sarcastic and non-sarcastic. The proposed feature sets are analyzed and estimated for their additional cost categories. The result in the random forest classifier gets 81% accuracy, the support vector machine classifier achieved 74%, and the KNN classifier had 58% accuracy.

Halim Z, Waqar M and Tahir M (2020) [16] have The suggested framework is evaluated using four different measures that are considered to be standard. Experiments are carried out on the datasets that are under consideration by vertically partitioning them into all features, top features, and bottom features, taking into account the outcomes of the process of feature selection. In addition to this, a qualitative and quantitative comparison of the planned work is done with two different methodologies that are considered to be state-of-the-art. The findings that were collected point to an

improved performance of the ongoing work, which has an accuracy rate of 83% on mean. By taking a small amount of text as an input and using the framework that has been proposed, it will be possible to recognize human emotions in a variety of contexts.

Sarcasm detection achieved immediate attention as different organizations used sentiment analysis opinion mining to their outcomes and detected activity in the social networks. According to the literature, Twitter mockery detection methods could be classified into four categories: Sarcastic-based on features, lexical-based on features, context-based on Features, and TF-IDF feature extraction with machine learning-based approaches in Figure 1.

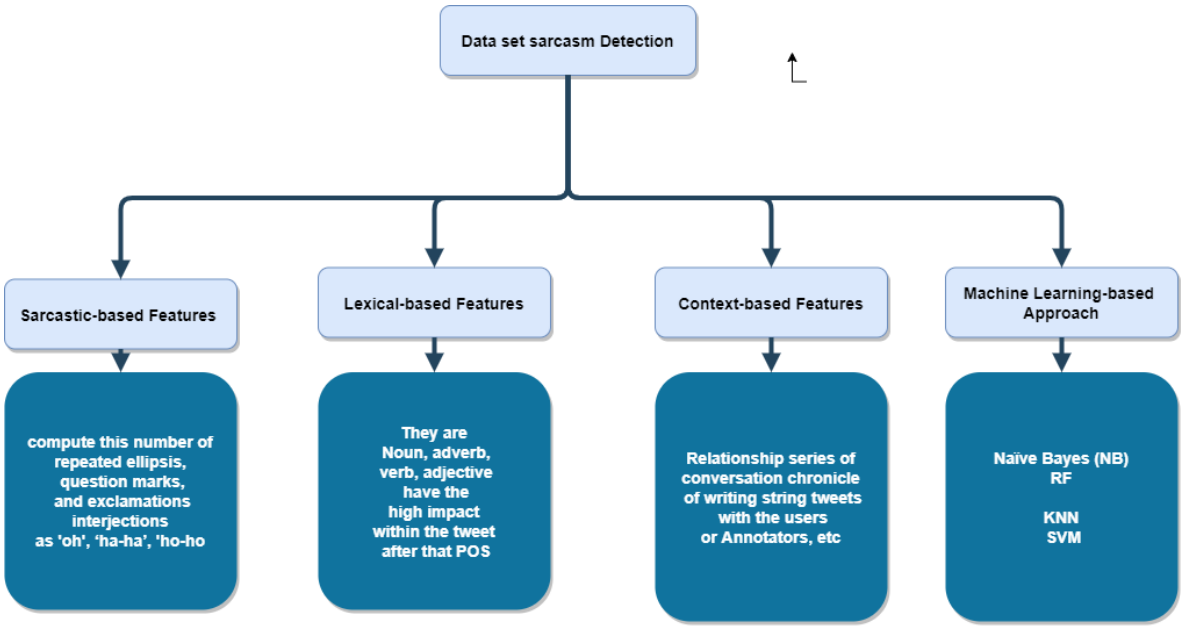


**Figure 1.** Classify social network sarcasm detection method.

## 3. Materials and methods

This section discusses dataset, preprocess, feature engineering, proposed models, algorithms, and classifiers.

### *3.1. Tweets dataset*

In this subsection, we describe dataset sarcasm tweets used in our work. This publicly available corpus online sarcasm dataset includes three columns: first column tweet index, second column text body, and last column labels. This tweets dataset is available online on Kaggle. It consists of 3834 balanced between sarcastic labels by (1) and not-sarcastic labels by (0), as proved in the link below. https://github.com/ronanmmurphy/Irony-and-Sarcasm-Detection/blob/main/data.txt[1].

### *3.2. Data preprocessing*

One of the disadvantages of getting a dataset from social networks is the noise that gets on with the dataset. Social network data, for example (tweet), maybe within a style of user's mention (@ &

[1] https://github.com/ronanmmurphy/Irony-and-Sarcasm-Detection/blob/main/data.txt

user), simple text, reference, content tag as hashtags (#), and URLs, in this step, preprocessed for the preparation dataset before the feature extraction and classification task as shown in Figure 2 below [46].


**Figure 2.** Preprocess steps for the preparation dataset.

### 3.2.1. Tokenization

Tokenization is the method of cutting sentences or words into shorter pieces named tokens like symbols, words, and phrases that are helpful on their own. This tokenization reduces (empty white area) character observed via the text document. The token indicates a series of characters obtained in the document attached to make the proper semantic unit beneficial after studying. That task was performed by applying the NLP toolkit [17].

### 3.2.2. Stop words removal

One of the essential manners of preprocessing is to clean out useless datasets related to stopping a word. Stop words involve text (the, an, a) essentially, preposition as (to, of, in, and), alongside another generally utilized word. All these, indeed, non-have any effect on detecting sarcasm in text data sentences [18]. Hence, we remove it before starting the process of the dataset. For Instance, after removed stop words from this sentence, "I don't understand how this topic," it will be "understand topic" terms remaining.

### 3.2.3. Noise removal

We have to remove symbols, non-ASCII, number newlines that are unnecessary for the data set, Twitter-saved terms, single-word tweets, and some basic string literals to rate the features mining manner.

### 3.2.4. Stemming

Stemming is the mending of the derived word concerning the source root word named a stem through removing the suffixes and prefixes of the word. In another meaning stemming is a process that decreases the keyword term numbers and improves classification display performance during the individual keywords are received from the other forms of that keyword. E.g., the word ‘shoplift’ can stem from ‘shoplifting.’ Many studies declared that a stem procedure contributes to classification

performance [19].

#### 3.2.5. Punctuation removal

Punctuations make noise for text data sentences represented by "?, %,#,$,&'+,-./:()*+,-./;~<=>:@[\]^_`{|}" these scrapped before extraction feature from the dataset, darning some particular punctuation, such as, ? !… are held that occasionally symbolize a sign of irony (sarcasm).

#### 3.2.6. Removal URLs

URLs mean (Uniform Resource Locators) into tweet dataset indicate the location in that website so not give any further knowledge. So we are removing it during the preprocessing stage of the sarcasm detection method.

#### 3.2.7. Elongated and truncated word

Tweets data sets contain duplicated characters in words like gooood, toooo, speeeed, and several more styles frequently in social network posts and comments. Those words habitually symbolize sarcasm in tweets [20]. To remove multiple characters, we need first remove the stop words' youuuu' then require to determine how many duplicated characters we can recognize in a word string, e.g., If we want to delete two or more two duplicates, characters will also face an issue because some word will change such as 'book' become 'bok' and 'cool,' 'col' first request Analyze the text data then duplicate characters remove by using a function. Input: 'youuuu looking niceeee' Output: 'you looking nice.' After all, filtered and cleaned text is gathered in that chunking process. Finally, saved text data is set as a file in CSV format for further binary text classification [21].

#### 3.2.8. Contraction replace

Contractions weren’t, haven’t, aren’t, couldn’t, shouldn’t, won’t, can’t, shan’t, etc. Then replace that with the complete form that will analyze the single term. Furthermore, apply forms to popular ellipses in tweet data like jk, lah, hella, lmao, etc.

#### 3.2.9. Part-of-Speech tagging

POS is the process of matching the words for their morphological form that helps learn their use inside a sentence. The critical post included in part of speech Tagging are Noun, Adverbs, Adjectives, and verbs. POS taggers carry the series of terms as data input and make the tuple listing as output, wherever every word is related to the appropriate tags [22,23].

### *3.3. Proposed model*

The structure of the proposed approach is illustrated in Figure 3 and consists of three sections text dataset preprocessing features extraction and irony classification modules. Proposed the improvement model to detect sarcasm by opinion study.

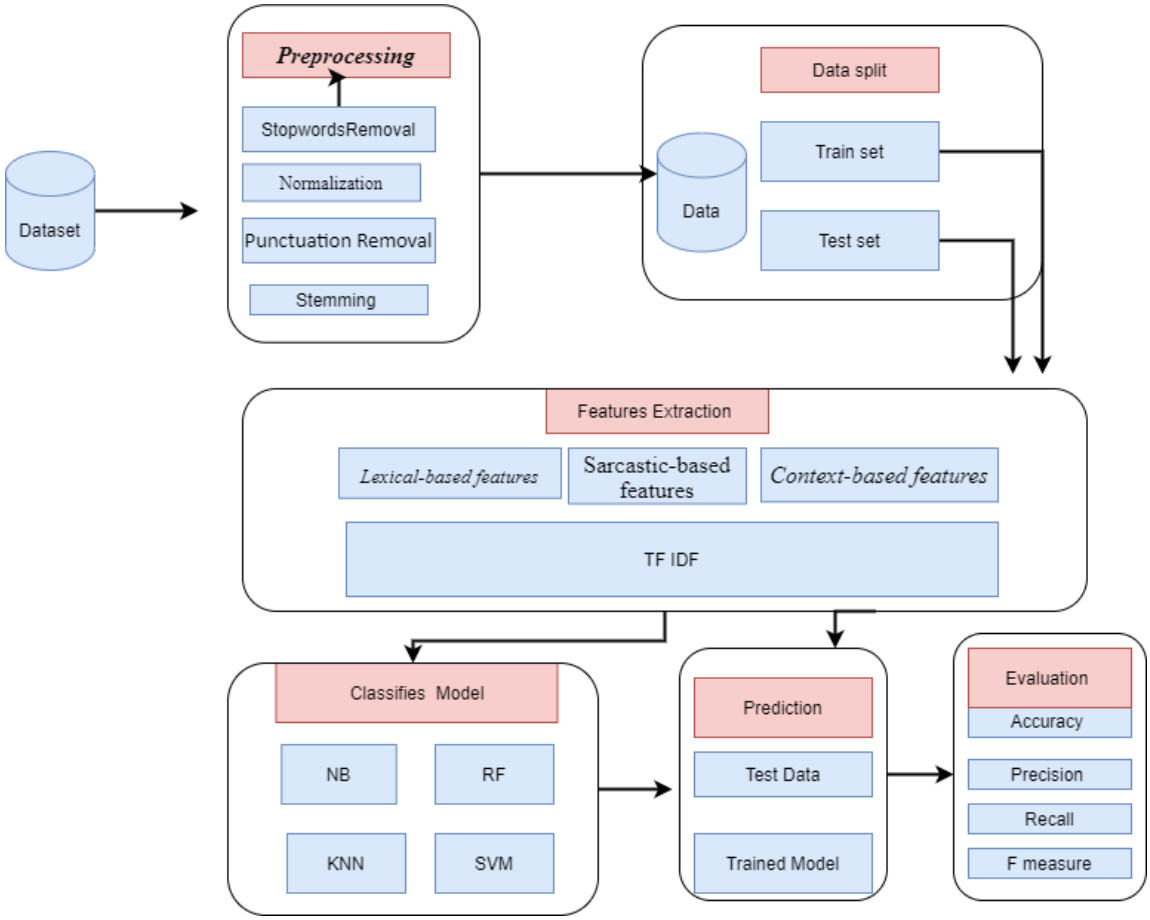


**Figure 3.** Proposed methodology sarcasm detection.

We proposed developed our model to detect sarcasm opinion from social networks. This model contains four main feature sets sarcastic based on features, lexical based on features, Context-based features, and Machine Learning-Based Approach using four Ml classifiers. In the Table below is a sample output of features engineering.

**Table 1.** A sample output of features engineering.

| **Sarcasm tweets** | **Definitive phrases** |
|---|---|
| **Oh good!! You do that. Thank you for making me fool** | " Oh, Good" |
| **Ahaa, full enjoying** | "Ahaa" |
| **Aww!!! I like hard work on holiday** | "Aw," "like hard work." |
| **I love waiting forever for my teachers** | "waiting forever" |

### *3.4. Feature engineering*

Feature engineering reduces the number of resources demanded to represent the dataset [24]. The main issues of interpreting complex text datasets stem from the number of variables required. The study of the huge variables number usually needs a huge piece of computation power and memory. These classification algorithms overfit the training models and propagate inadequate for

other models. Therefore, feature extracts become necessary to deal with classified problems with many variables. We have applied feature sets for this purpose of classification. Utilizing multiple features allows us to compare the accuracy and F-measures obtained to various features. Sarcasm categories illustrated how extracted three feature sets [25].

### 3.4.1. Sarcastic based feature set

Commonly, people create complex or confusing sentences that utilize rare terms to make them unintelligible to the reader or listener to get a definite resolution to events. Admittedly, when people utilize sarcasm to avoid it, people plan to hide their real feelings or opinion by hiding them with joy. People occasionally attempt to move the sarcasm information by punctuation, like a repeated ellipsis, question mark, also exclamation. Therefore, we compute this number of repeated ellipsis, question marks, and exclamations. Addition to, interjections following 'oh', '-ha-ha-ha, 'ho-ho', 'ho-ho-ho-ho', 'shh', 'oww', 'ah', 'Nah', 'aha', 'awesome', 'aww', 'kid', 'uh', 'yay', 'bingo', 'bah', 'brilliant', 'boo', and numerous more. Somebody accomplish them to express feelings in many forms that distinguish sarcastic tweets from the dataset. Words such as mooooooove, looove, gooood, duplicated letters of more than two, a probable indication of sarcastic tweets, and duplication of vowels to indicate the same. Thus, compute several duplicated letter parts and vowel repeats into the tweets dataset. Therefore, we extract that feature before removing duplicated letters in words. On the other hand, somebody uses capital letters, for example, GREAT, WONDERFUL, and parts of the term as uppercase, for instance, Instance, gREatd, wONDERful, for showing their mockery activity. We get the number there as well. finally count the number of laugh particularly 'lool', 'wow', "lhh", 'kidding', "jk", 'haa haa', 'hahah', 'wtf' 'ha-ha'. Ironical slangs such as 'ayfkmwts', 'kyso', 'lmao', 'fubar', 'lhh', roflmao', 'stfw', 'rofl' 'stf',' are utilized within the tweet, also we get in consideration them also. We count duplicated number quotes into tweets like letter repeats. Moreover, hashtags too cover feeling content. They were applied to ambiguities tweet data true purpose provided within a post during several events. Consequently, we, too, compute and analyze the mining score. Lastly, we compute the polarity summary (+1 to positive while -1 to negative of polarity) to n-gram unigram w1, bigram w2, and trigram w3 in the text data.

### 3.4.2. Lexical based feature set

We extract textual or Lexical-based features. These nouns, verbs, adjectives, and adverbs significantly influence the tweets after that POS. We count it separately according to individual text tweets. Furthermore, intensifiers like awfully, bloody, strikingly, absolutely, ridiculously, outrageously, excessively, tremendously, amazingly, etc. We are sometimes used to exaggerating a tweet to express negative feelings during positive vice versa and intensifiers. Accordingly, we, too, compute the number of negative and positive intensifiers present in each tweet. Finally, all tweets' opinion is calculated, showing the tweets' absolute polarity.

### 3.4.3. Context-based feature set

Regarding feature set, obtain the numbers of users' hashtags and mentions involved in a tweet. People tend to express their activities by employing different hashtags and posting comments. The relationship between the audience and users followed with a direct communicative context could be important for improving the sarcasm prediction precision. Using a context-based model, post-level

irony detection on a social network was applied to sarcasm detection. Quantitative evidence of actual tweets of the author can give further context to sarcasm detection.

#### 3.4.4. Term Frequency-Inverse Document Frequency (TF-IDF)

Inverse Document Frequency (IDF) is the technique used in conjunction with term frequency to lessen the impact of essentially popular text words within the data set. IDF assigns the highest weighting to a word by either low or high-frequency words within text documents. Here compound of TF then IDF is greatly learned as Term Frequency-Inverse document frequency (TF-IDF) [26,27]. In a way, a mathematical illustration of the weighted word into the document with TF-IDF is provided in Equation (1).

$$W(d, t) = TF(d, t) * \log\left(\frac{N}{df(t)}\right) \quad (1)$$

Then the end equation for TF-IDF (2)

$$W_{t,d} = \left(1 + \log tf_{t,d}\right) * \log_{10}(N/df_t) \quad (2)$$

Here df(t) are the number of documents and N number of text documents covering the term t in the text data set.

The number one word in Equation (1) improved the recall while the number two-term improved the precision of the words embedding [28,29]. So TF-IDF tries to overcome the difficulty of general terms within a text dataset but suffers from different descriptive limits. Specifically, TF-IDF cannot account for the similarity among words within text documents, for each word is independently given as the index.

### *3.5. Algorithm classification of the proposed improved model to detect sarcasm*

Description of the term.
Td: Tweets dataset
N: tweet numbers
P: preprocessing dataset
T: Tokenization
SW: Stop word removal function
NR: Noise Removal
S: Stemming
P: Punctuation Removal
ET: Elongated and Truncated Word
Pos: Part-of-Speech Tagging
L: Lower Case conversion function
LF: Lexical features sets function extraction
SF: Sarcastic features sets function extraction
CF: Context features sets function extraction

**Algorithm 1. Tweets text preprocessing processing.**

1) **Input:** Tweets data (TD)
2) **Output:** Features sets as input for ML Classifiers.
3) **Procedure:** Features Extraction (TD)
4) i ←1
5) While i < = N
6) Td ← Td (i) from tweets dataset
7) Td_sw← sw (Td _sw) // Stop word removal from tweets dataset
8) Td_ T ← T (Td _ T) // tokenize the tweets
9) Td_ NR ← NR (Td _ NR) // Noise Removal from tweets dataset
10) Td_ S ← S (Td _ S) // Stemming dataset
11) Td_ P ← P (Td _ P) // Punctuation Removal from tweets dataset
12) Td_ ET ←ET (Td _ ET) // Elongated and Truncated Word dataset
13) Td_ Pos ←Pos (Td _ Pos) // Part-of-Speech Tagging dataset
14) Td_ L ←L (Td _ L) // Lower Case conversion for dataset
15) P (i) ←Td_T // preprocessing tweet
16) i ← i + 1
17) **END**

18) i ←1
19) While i < = N
20) FFL←FL (P (i)) // extraction lexical features from the preprocessed dataset
21) F SF ←SF (P (i)) // extraction Sarcastic features from the preprocessed dataset
22) F CF ← CF (P (i)) // extraction Context features from the preprocessed dataset
23) IF ← [FL, SF, CF] // features sets extracted

24) WRITE IF
25) i ← i + 1
26) **END**

The processes of building features vectors to the presented features set can be described as follows. The stages are divided into two parts: dataset preprocess and features engineering. In part one of the data preprocessing, tweets dataset is first loaded to memory. After that, we used eight preprocessed processes executed before extracting determine features utilizing the preprocessing (P) function. It begins with stop words removed from a tweets dataset by using (SW). Then Td to tokenize sarcasm expression to the unique token by using T. Noise Removal from the tweets dataset using NR on Td and stemming dataset using (S) on Td. Punctuation Removal from tweets dataset (P) on Td. Elongated and Truncated Word dataset (ET) on Td. Part-of-Speech Tagging dataset (Pos) on Td, Lower Case conversion for dataset (L) on Td. Finally, the processed tweets dataset Td is stored in a file named (P). The second part features engineering; on the other way, the processing dataset is loaded for memory to sarcasm training and testing model for classification outcomes. For every processed text tweet, the features in the numerical form are extracted utilizing the features extraction

function represented by (I). Those features have Lexical feature (FL), extract sarcastic feature from the preprocessed dataset (SF), and extract Context feature from the preprocessed dataset (CF). Also, the feature sets operation is executed employing [FL, SF, and CF] feature sets extracted. Finally, the combined feature sets are converted for the file format and fed as the input for a classifier to classification steps.

### *3.6. Classifiers models*

This subsection presents ML algorithms related to the classification model applied in this research. ML analyzes algorithms that can learn from it and gives the prediction on text datasets [30]. It is regularly applied to a model training set that occurs and then features an engineering step. The selected and extracted features from a text data set were utilized to build an ML algorithm for sarcasm detection classification. ML algorithms have been used to categorize tweets as sarcastic or not-sarcastic. Different algorithms such as RF [31], NB [21], SVM [32], and KNN [33] classifiers have been tested in various experimentations to determine the best classifier with different performance methods for sarcasm prediction. Hence, we work to determine a better classifier for the unique data set. Therefore, four various classifiers, including the NB, RF, SVM, and KNN, have been used to define the model's implementation of features set to irony classification [34]. As the focus in choosing the ML algorithm to be used in this research, three-point applied for scaling down the selection. First we presented literature on classification algorithms for sarcasm detection is required during the choice of a particular classifier. Second, the text data mining research was utilized to guide model choice. Third, they compared outcomes on extensive data sets and trained in selecting classification algorithms. The classifiers are briefly illustrated below.

#### 3.6.1. Na ïve Bayes (NB)

Na ïve Bayes (NB) is beneficial for classifying separated features like text classification or text documents. Na ïve Bayes develops multinomial order including Bayes theorem variables V1, V2... In class, C is independent. Equations (3) and (4) handled NB to date text classification in the test dataset:

$$P\ (C|V) = \frac{p(V|C)}{p(V)} \tag{3}$$

$$P\ (C|\ (v1, v2... vn) = \frac{p(V1|C)P(V2|C)\ldots\ldots\ldots\ldots P(Vn|C)P(C)}{p(V1)P(V2)\ldots\ldots\ldots\ldots P(Vn)} \tag{4}$$

Hence, C means class, then V = (v1, v2..., vn) describes the features of the vector. Here, find these features continue conditionally autonomous. Measure residue consistent with each provided input.

$$NC = GMA\ C\ p(C)\ \prod_{i=1}^{n} p(vi|C) \tag{5}$$

Equation (5) is related to calculating a probability of a distributed group of inputs to every probable value of class C, also taking an output by the highest probability. La Place semi thing applied and earlier probably of a class that fitted according to the text data [35].

#### 3.6.2. Random Forest (RF)

Random Forest (RF) includes a set of trees (decision trees) that run independently in this

algorithm. Each branch has a 'Gini index' utilized for the acquisition decision branch. Here is the index estimated by Equation (6):

$$\text{Gin} = 1\text{-}\sum_{i=1}^{c}(\boldsymbol{pi}) \quad (6)$$

Hereabouts, *pi* indicated the probability of I class, and c indicated all number of classes. Then we utilized 100 trees in the forest wherever the kind of split with 'Gin'—internal nodes [36].

### 3.6.3. K-nearest Neighbor Classifier (KNN)

KNN is the simple classifier approach for adapting and achieving each feature type. Here, the model too easily handles binary class and multi-class. KNN is restricted through data set constraints to big search difficulties to obtain the nearest neighbor. In addition, the execution of KNN performances depends on getting the significant space function, so working that method is a powerful dataset-dependent process [31].

### 3.6.4. Support Vector Machine (SVM)

SVM is concerned with the classification of binary data associated difficulties; support vector machines SVM are common models [37,38] (SVM) is used the nonlinear map in order to convert the real training dataset to the highest dimension. In the new dimension, it was searched during the linear dividing hyperplane. A hyperplane is a "decision limit" separating the tuples from one class from different. By a suitable nonlinear mapping into enough high dimension data set. The cases are classified with a hyperplane into a binary classification problem in this way $w^t$ X + b = (0), wherever w is dimensional weight, the vector standard to the hyperplane. The prejudice term b, an offset of the values data points, is described by x. Defining a value of w including b holds basic tasks during SVM. While the liner's position, w may be resolved.

## *3.7. Algorithm classification of the proposed improved model to detect sarcasm*

Description of the term.

L: Lexical feature content.

S: Sarcastic feature content.

C: Context feature content.

B: Classification label.

Ls: Lexical-based sarcasm tendency feature

Ss: Sarcastic –based on sarcasm tendency features

Cs: Context-based on sarcasm tendency features

**Algorithm 2. Feature extraction process.**

1) **Input:** Training set T = {(L1, S1, C1), (L2, S2, C2)..…….. (Ln, Sn, Cn)};

2) Three classifiers, G1, G2, and G3; a testing object F = (L, S, C);

3) **Output:** The label of F;
4) Training:
5) Build lexical features training sets T1 = {(L1, B1), (L2, B2)……... (Ln, Bn)};
6) Training G1 on T1;
7) For i = 1 to n do
8) Applying G1 on Li to get Ls;
9) **End** For
10) Build Sarcastic features training sets T2 = {(S1, B1), (S2, B2)… (Sn, Bn)};
11) Training G2 on T2;
12) For i = 1 to n do
13) Applying G2 on Si to get Ss;
14) **End** For
15) Build Context features training sets T3 = {(C1, B1), (C2, B2), . . ……., (Cn, Bn)};
16) Training G3 on T3;
17) For i = 1 to n do
18) Applying G3 on Ci to get Cs;
19) Test:
20) Apply G1 over F = (L) to get ls L its label Ls B;
21) Apply G2 over F = (S) to get ls its label Ss B;
22) Apply G3 over F = (C) to get ls L its label Cs B;
23) Return B;

Extract three feature sets in our work to detect sarcasm. These features have two novel algorithms: one in preprocess and the second in the feature extraction phase. Therefore, stand to s classify sarcasm dataset identifications through the lexical features Sarcastic feature content and Context feature content in the second steps—the effectiveness of ‘Algorithms one and Algorithms two as followings.

- The preprocess parts of the features engineering techniques used in algorithms assist in a dataset preparations phase, such as Stop word removal, tokenization, Noise Removal, Stemming, Punctuation Removal, Part-of-Speech Tagging, and Lower Case conversion before doing the feature extraction method.
- Algorithm feature extraction feeds stepwise representation to extract the proposed form of the feature vectors used as an input for the ML algorithm.
- Two algorithms with three feature sets to classification algorithm aids define the stepwise procedures to classify tweets within sarcastic and non-sarcastic for achieving the predictive proposed performance, making it easy to understand.

## 4. Experimental settings

This section presents the classification experiments to detect the social network's sarcasm (sarcastic and not-sarcastic) (tweet). Performance experiments were implemented on the system run on Windows ten by Intel(R) Core(TM) i7-9250U CPU @ 1.60 GHz, 1.80 GHz operating systems by 16 GB Random Accessed Memory (RAM). The dealing with data, preprocess and features engineering tasks were executed within the Jupyter notebook environment, and the python program language development environment was performed. Feature sets were applied to the sarcasm investigation experiment as input to different ML-based classifier. Next, we experimented and tested with four ML classifiers models, including NB, RF, SVM, and KNN, to determine sarcastic opinions in a provided dataset. The goal of applying various models is to achieve a better performance outcome. We used the ten folds cross-validation method in performing total experimentations in our work. The default setting was utilized throughout the experiment. Four regular evaluation metrics, accuracy, f-measure, precision, and recall, were weighted and tested on both classes through the experiments. Furthermore, the parameter settings of classifiers were applied throughout the experiment. The classifiers parameter turning is depicted in Table 2.

**Table 2.** Parameters optimizer and classifiers are tuning values.

| Classifier | Parameters |
|---|---|
| NB | Alpha = 1.0, fitting priors = true, class priors = none |
| RF | N−Estimators = 100 |
| SVM | C = 3 Batch size = 100 |
| KNN | K = 5 (Neighbors = 5) Batch size = 100 |

### *4.1. Confusion matrix*

The confusion matrix or (error matrix) is the unique representation that provides an idea of classifier performance, particularly supervised ML classification. Here confusion matrix contains two instances (“actual” and “predicted”) in the same class [39]. The positive is identified, whereas the negative is discarded. Hence, following category, true positive is the case that is correctly classified, whereas false positive is not accurately categorized. The false-positive Instance expresses the type 1 error, symbolizing that the number of instances is not correctly symbolized as positive. But, the true-negative in these cases are accurately discarded, then false negatives are indicated as incorrectly classified cases [40]. False-negative expresses type 2 error, symbolizing that many cases are inaccurately classified as negative. A confusion matrix is represented in Table 3.

**Table 3.** Confusion matrix.

| | True condition | |
|---|---|---|
| **Predicted condition** | TP | FP |
| | FN | TN |

### *4.2. Evaluation measures*

In this part, we applied Precision, Accuracy, Recall, and F1 Measure by 100% for model

evaluation metrics outcome that has been used to compare and analyze the achievements of the supervised classification algorithms to sarcasm detection. Frequently evaluated metrics are applied during supervised classification algorithms, enabling us to efficacy test the algorithm. We used a confusion matrix to estimate the performance detection of sarcasm such as TP, FP, FN, and TN [41,42].

#### 4.2.1. Precision

Precision indicates a proportion of correctly detected sarcasm news messages between classified posts/messages/news records. It defines the efficiency of the proposed methods. The formulation for estimating the precision is specified in Equation (7) [45,46].

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FP}} \tag{7}$$

#### 4.2.2. Recall

Recall shows the proportion of accurate activity detection of sarcasm posts/news/messages. The recall is further recognized as Sensitivity. The recall values could be estimated utilizing the formulation represented in Equation (8) [47,48].

$$\text{Recall} = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FN}} \tag{8}$$

#### 4.2.3. F-measure

F-measure indicates evaluating all performance measured by the measured outcomes of recall and precision. The values of the f-measure could be estimated utilizing this formulation shown in Equation (9).

$$\text{F} - \text{Measure} = \frac{\mathbf{2}\times\mathbf{pre}\times\mathbf{Rcall}}{\mathbf{pre}+\mathbf{Rcall}} \tag{9}$$

## 5. Results and analysis

This section discusses the experiment results and our effective model with a dataset. First, all four classifiers were applied to our proposed employing 10-folds cross-validation. We described all classifiers' accuracy, f-measure, recall, and precision performance. Here, performance results were usually satisfying and helped predict better classifiers for sarcasm classification. Second: the evaluation of effective feature set performance outcomes has been listed and discussed in Table 4, where it shows.

### *5.1. Evaluate effective sarcasm detection*

We have evaluated Four ML classifiers based on accuracy, F1-score, recall, and precision for the study this model performs. As illustrated in Table 4, Naïve Bayes and KNN display the higher precision in both, around 90%, but SVM shows the lowest value during precision. Regarding F1 metrics and recall, the KNN classifier performs better than other classifiers, with a 91.65% in f score and 90.89% in recall. The KNN outperforms different classifiers, as illustrated in Table 4.

**Table 4.** Evaluation of effective sarcastic detection classification.

| Model Classifiers | F1-score | Precision | Recall |
|---|---|---|---|
| NB | 89.71 | 90.34 | 89.21 |
| SVM | 74.00 | 77.65 | 77.45 |
| RF | 82.39 | 81.34 | 81.90 |
| KNN | 91.65 | 90.76 | 90.89 |

We notice that KNN Evaluation of Effective Sarcastic Detection Classification has the best performance of other model classifiers.

*5.2. Evaluating of effective feature set*

We have evaluated three sets of feature sarcastic feature sets, context feature sets, and lexical features set by four common ML classifiers, such as SVM, RF, KNN, and NB. Therefore, we measure our outcome by four metrics evaluation: F1-score, precision, and recall. Table 5 and Figure 4 show that sarcastic feature sets achieve much better values with KNN and RF in evaluating metrics F1 score, precision, and recall than other features. Conversely, the context on-based feature sets obtained the lowest values with F1 score, precision, and recall.

**Table 5.** Different classification along with valuation of efficient all feature set.

| Model Classifiers | Features set | F1-score | Precision | Recall |
|---|---|---|---|---|
| NB | Sarcastic | 76.90 | 74.82 | 74.12 |
| | Context | 59.43 | 57.25 | 57.98 |
| | Lexical | 60.65 | 60.01 | 60.22 |
| SVM | Sarcastic | 71.98 | 74.56 | 71.18 |
| | Context | 58.40 | 59.26 | 58.91 |
| | Lexical | 61.65 | 59.22 | 61.31 |
| RF | Sarcastic | 80.00 | 80.01 | 79.11 |
| | Context | 59.47 | 59.21 | 59.01 |
| | Lexical | 62.61 | 60.24 | 61.65 |
| KNN | Sarcastic | 81.91 | 79.86 | 80.19 |
| | Context | 49.88 | 43.26 | 44.92 |
| | Lexical | 42.67 | 42.11 | 45.20 |

Through evaluation of the efficiency, all feature sets Table 5 above and Figure 4 below illustrate the Sarcastic feature set achieves much better values with KNN, RF, NB, and SVM, respectively, with the Sarcastic. After that Lexical Features set comes in the second phase. Finally, the context-based feature set obtained the lowest values.

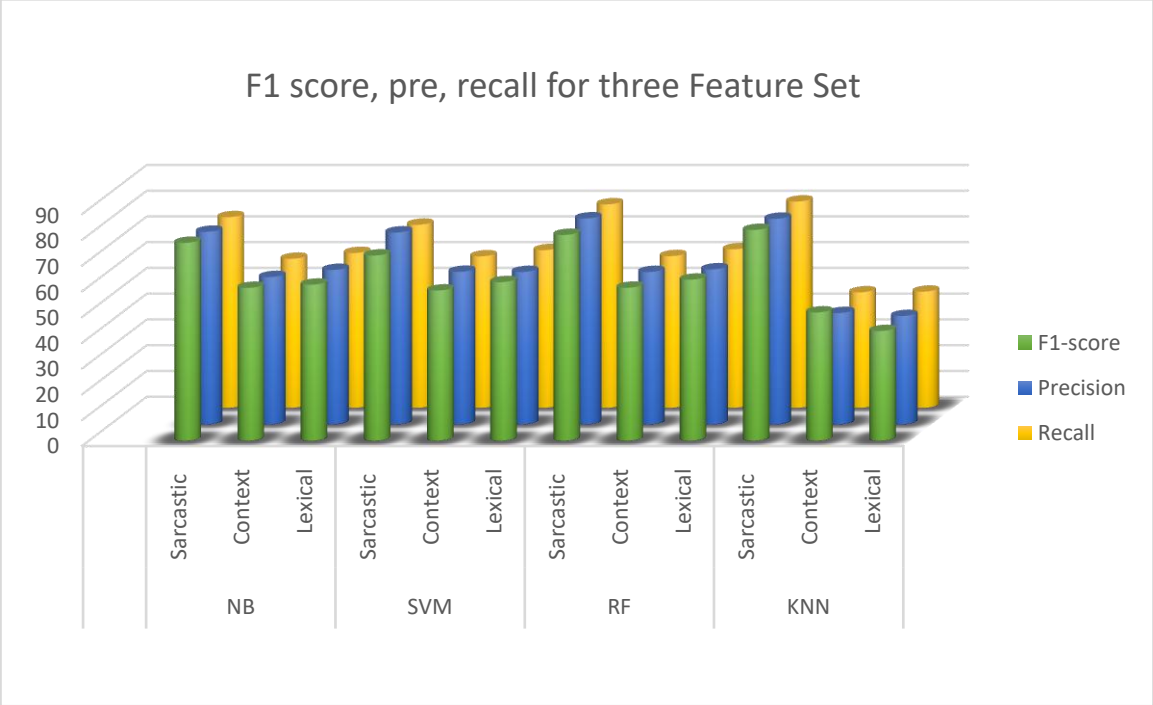


**Figure 4.** F1 score, pre, recall for three feature sets.

*5.3. Performances of individual accuracy for features set*

Accuracy with all Feature Set shown in Table 6 and Figure 5 below, Accuracy analysis results with the sarcastic feature in our research have a better contribution during irony detection and interpretation of a text in the dataset than another set of feature context and lexical features. Regarding the remainder of the feature set, lexical features have been archived with much better accuracy than context based on the feature. We noticed context based on features had less influence on irony detection. Naïve Bayes consistently achieves much better representations of the classification approach. Also, the sarcastic feature achieves about 80 % accuracy. SVM gives the lower accuracy with lexical features and context features, respectively, while sarcastic sets are less valuable in SVM classifier. For sarcastic features to provide notably more precision, it is deserving of remarking that the appropriate selection of the feature set of sarcastic could improve accuracy alternatively selecting more feature sets. Viewing the values achieved during our sarcasm identification experiments showed that sarcastic and lexical feature sets presented more influence for sarcasm detection in the text dataset among the three feature sets specifically for RF and KNN, much better than other classifiers models. In particular, those feature sets' precision is continually more refined than different feature sets by RF and KNN classifiers. The KNN achieves maximum accuracy of 89.19% for the Lexical Features set of 82.70% for the sarcastic feature set, as shown in Table 6.

**Table 6.** Performance three feature set accuracy.

| Classifier / Feature Set | Sarcastic Features | Lexical Features | Context Features |
|---|---|---|---|
| | Accuracy | | |
| NB | 77.87 | 63.32 | 60.14 |
| SVM | 74.89 | 73.12 | 75.60 |
| RF | 80.43 | 83.52 | 59.32 |
| KNN | 82.70 | 89.19 | 55.87 |

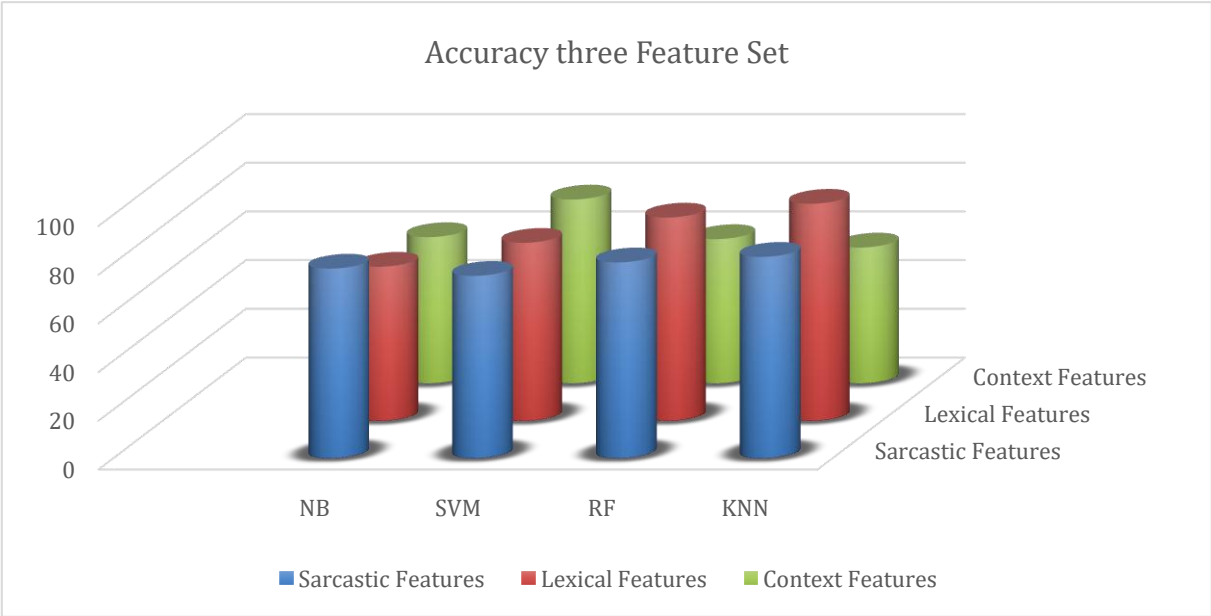


**Figure 5.** Performance three feature set accuracy.

*5.4. Combined two feature sets accuracy*

In Figure 6 and Table 7 discussed below, accuracy combined two feature sets Lexical with Sarcastic-based on features achieved higher accuracy for KNN than other combinational features. The investigation indicates that sarcastic and lexical feature combinations are obtained more accurately than feature sarcastic with context and Lexical with Context features. Chiefly, K-Nearest Neighbor produces a higher accuracy of lexical and sarcastic features, around 90%. After that, lexical and context for the same classifier were 89.97%. Here we can say that combining two feature sets increases and improves accuracy remarkably. To complete techniques, lexical and context-based features combined presented the lowermost accuracy, around 62.87%. As clarified, excluding lexical and sarcastic sets, it is seen that the sarcastic feature integrated is feature set which leads to obtaining the highest accuracy of the other combination sets. Through accuracy combined two feature sets into three feature sets together as following first combined lexical and sarcastic with four models classifies KNN, RF, NB, and SVM the best accuracy with KNN and lowest accuracy with SVM. Secondly, combining sarcastic with context with four models classify shown KNN, RF, SVM, and NB as the best accuracy with KNN and lowest accuracy with NB, and third, combining lexical with context with four models classified shown KNN, RF, NB, and SVM the best accuracy with KNN and lowest accuracy with SVM.

**Table 7.** Combined two feature sets accuracy.

| Classifier | Lexical with Sarcastic | Sarcastic with Context | Lexical with Context |
|---|---|---|---|
| NB | 79.06 | 78.98 | 62.87 |
| SVM | 76.87 | 75.32 | 75.13 |
| RF | 86.54 | 80.73 | 84.21 |
| KNN | 90.00 | 81.21 | 89.97 |

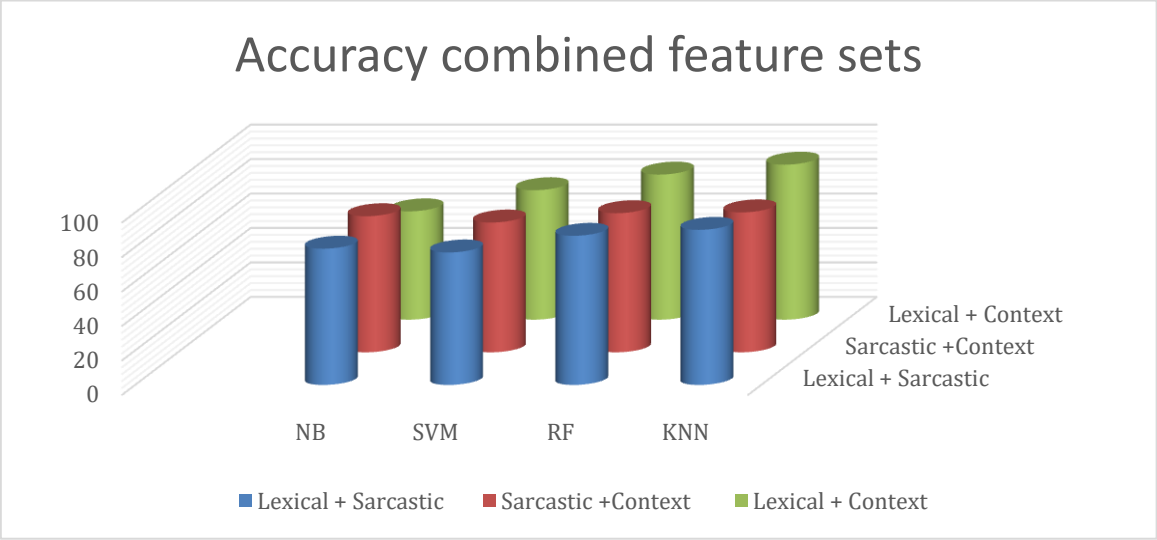


**Figure 6.** Combined two feature sets accuracy.

*5.5. Integrate all features sets incrementally*

We integrate all three features sets incrementally here in this experiment. We analyzed three separate combinations in order to know which combination performs better than the others. In Table 8. And Figure 7 below. We combined every two features, as shown in Table 6. Here we do the (Sarcastic Lexical and context) feature set, which significantly improves by integrating all feature sets incrementally in this experiment. We obtained the highest accuracy, approximately 90%, with KNN for the Lexical and Sarcastic-based feature set & with RF accuracy of 87.34%. Hence, we notice slight improvement through combining features. In fact, it seems that combining features and then adding more features does not help improve these models' performances; we notice feature selections are the primary participant of performance effectiveness classification.

**Table 8.** Accuracy combined feature sets integrate all feature sets incrementally.

| Classifiers /Accuracy | Sarcastic and Lexical and Context |
|---|---|
| NB | 79.89 |
| SVM | 77.65 |
| RF | 87.34 |
| KNN | 90.51 |

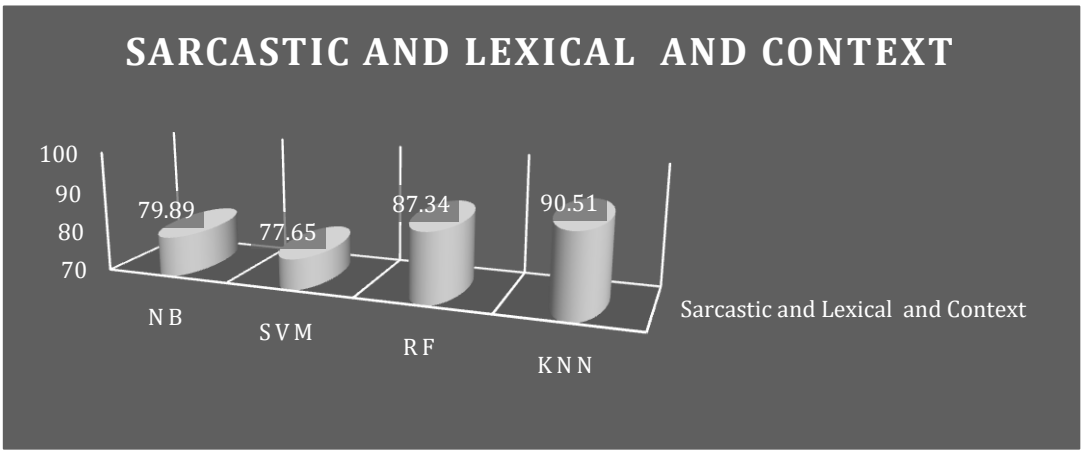


**Figure 7.** Accuracy of combined feature sets Integrate all feature sets incrementally.

## 6. Result discussion

Depending on values that are achieved in our model sarcasm distinction experiments, Sarcasm is detected in tweets more effectively by the sarcastic feature set than in the other three. It has consistently exhibited higher accuracy in comparison with other feature sets with all classifiers, with the KNN achieving a higher accuracy (around 91.51%). In this section, we'll examine the results of different feature set combinations. The results of our study were based on the analysis of three distinct combinations to determine which one outperforms the others.

Accuracy to incrementally combined category: based on our achieved as of Tables 5–8, including Figure 7, we note that the sarcastic-based and Lexical features controlled the accuracy. In Table 7, accuracy combined with two feature sets, Lexical with Sarcastic-based on features, achieved higher accuracy for KNN than other feature combinations. The investigation outcome indicates that lexical and sarcastic feature combinations obtain more accuracy than Sarcastic with Context and Lexical with Context features. Chiefly, K-Nearest Neighbor produces a higher accuracy to sarcastic and lexical feature sets, around 90%. We analyzed three combinations to know that combinations perform better than others but with a different style. We combined three features, and two features were set once and Sarcastic-based Features as shown in Table 6. Here we do the (Sarcastic-based Features) features set, then we combine (Sarcastic and context) features set, and we combine all once (Sarcastic, Lexical, and context) in this experiment table. We achieved the highest accuracy (approximately 90.51%) with KNN for three features together with the Sarcastic, Lexical, and Context-based feature set & we noticed improvement with RF accuracy of 87.34% exceeding the accuracy-related to NB, and SVM, to different features selection. Hence, we see slight improvement even through combining features. But, combining features or additional features does not help increase these models' performances; as we notice in the previous Table 6, feature selection is the primary part of the performance-effective classification.

Finally, we performed tests to see that classifiers are better if we increased the feature set to model, considering the feature set of sarcastic as the primary feature set. According to analysis, every feature set accuracy we notice is not more important than others-for features combinational, and feature selection is the primary part of performance efficient classification. Therefore, adding more and more feature sets is not constantly useful to achieve the best accuracy. But, KNN and RF achieved the highest accuracy among classifiers models in this case.

### *6.1. Compared of proposed approach with baselines*

Proposed performance studies compared to the technique have been used for similar tasks. Our proposed model sarcasm detection from tweets dataset uses three feature set datasets to investigate and study a performance difference of the proposed approach with existing models. Our proposed model is shown in Table 9. Ghosh et al. (2015) [43] achieved 71.00% accuracy, and Khodak et al. (2017) [44] achieved 83.00 %, while our proposed model achieved higher accuracy of 90.51% in sarcasm detection from social networks.

**Table 9.** Compared proposed approach with baselines.

| Our Proposed / Baselines | Accuracy (%) | Reference |
|---|---|---|
| Ghosh et al. (2015) | 71.00% | [43] |
| Khodak et al. (2017) | 83.00% | [44] |
| Our Proposed Approach | 90.51% | |

## 7. Conclusion and future work

Sarcasm detection in the social network provides invaluable vision in the present public opinion on events and trends in real-time, so detecting sarcasm is highly significant in this area. This research paper proposed an improved and effective model to detect sarcasm in text analysis. We applied three feature set engineering: context-based features set, Sarcastic-based features, and lexical-based features. These features by two Novel Algorithms for an effective lead model to detect sarcasm first algorithm deals with preprocessing, and the second algorithm for features sets raw unstructured text data from the social network. And we used various supervised machine learning (ML) such as K-nearest neighbor classifier (KNN), Naïve Bayes (NB), support vector machine (SVM), and Random Forest (RF) classifiers with TF-IDF feature extraction representation data. Evaluation model performance for different evaluation Measures used precision, accuracy, recall, and F1 score by 100% to model evaluation metrics results to compare and analyze the achievements from training effective model to detect sarcasm. We perform the model differently to see the effect of three feature sets through the experiment individual, combining two feature sets and gradually combining three feature sets. The best Lexical features with KNN are higher with 89.19 % accuracy campers to other classifiers. Combining two feature sets (Sarcastic and Lexical) has slightly improved with the same model KNN with 90.00% accuracy. When combining three feature sets (Sarcastic, Lexical, and context), the accuracy improved by 90.51% with the same KNN classifier. Curacy with the KNN classifier when combining all features set together. The proposal can be extended to study the detection of different languages such as Arabic in the future. Also, we are working on various deep learning-based feature set approaches to sarcasm detection; we hope to publish our research findings soon.

## Conflicts of interest

The authors declare they have no conflicts of interest to report regarding the present study.

## References


1. Pawar N, Bhingarkar S (2020) Machine Learning based Sarcasm Detection on Twitter Data. *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 957–961. https://doi.org/10.1109/ICCES48766.2020.9137924
2. Suhaimin MSM, Hijazi MHA, Alfred R, et al. (2017) Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts. *2017 8th International Conference on Information Technology (ICIT)*, 703–709. https://doi.org/10.1109/ICITECH.2017.8079931
3. Bharti SK, Babu KS, and Raman R (2017) Context-based Sarcasm Detection in Hindi Tweets. *2017 9th Int. Conf. Adv. Pattern Recognition, ICAPR*, 1–6. https://doi.org/10.1109/ICAPR.2017.8593198
4. Zhang M, Zhang Y, Fu G (2016) Tweet sarcasm detection using deep neural network. *COLING 2016 - 26th Int. Conf. Comput. Linguist. Tech. Pap.*, 2449–2460.
5. Athira MR, Chithra C, Anil G, et al. (2020) Sentiment Analysis - Sarcasm Detection in Twitter. *Journal of Computer Engineering (IOSR-JCE)* 22: 42–46. https://doi.org/10.9790/0661-2204024246
6. Prasad AG, Sanjana S, Bhat SM, and B. S. Harish (2017) Sentiment analysis for sarcasm detection on streaming short text data. *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, 1–5. https://doi.org/10.1109/ICKEA.2017.8169892
7. Bindra KK, Prof A, Gupta A (2016) Tweet Sarcasm : Mechanism of Sarcasm Detection in Twitter. *International Journal of Computer Science and Information Technologies* 7: 215–217.
8. Bharti SK, Vachha B, Pradhan RK, et al. (2016) Sarcastic sentiment detection in tweets streamed in real-time: a big data approach. *Digit Commun Netw* 2: 108–121. https://doi.org/10.1016/j.dcan.2016.06.002
9. Sarsam SM, Al-Samarraie H, Alzahrani AI, et al. (2020) Sarcasm detection using machine learning algorithms in Twitter: A systematic review. *Int J Market Res* 62: 578–598. https://doi.org/10.1177/1470785320921779
10. Bouazizi M, Ohtsuki T (2015) Sarcasm Detection in Twitter: "All Your Products Are Incredibly Amazing!!!" - Are They Really? *2015 IEEE Global Communications Conference (GLOBECOM)*, 1–6. https://doi.org/10.1109/GLOCOM.2015.7417640
11. Eke CI, Norman AA, Shuib L, et al. (2020) Sarcasm identification in textual data: systematic review, research challenges, and open directions. *Artif Intell Rev* 53: 4215–4258. https://doi.org/10.1007/s10462-019-09791-8
12. Saha S, Yadav J, Ranjan P (2017) Proposed Approach for Sarcasm Detection in Twitter. *Indian J Sci Technol* 10: 1–8. https://doi.org/10.17485/ijst/2017/v10i25/114443
13. Sharma S, Chakraverty S (2018) SARCASM DETECTION IN ONLINE REVIEW TEXT. 1674–1679.
14. Wen Z, Gui L, Wang Q, et al. (2022) Sememe knowledge and auxiliary information enhanced approach for sarcasm detection. *Inf Process Manag* 59: 102883. https://doi.org/10.1016/j.ipm.2022.102883
15. Pawar N, Bhingarkar S (2020) Machine learning-based sarcasm detection on Twitter data. *Proc 5th Int Conf Commun Electron Syst ICCES 2020*, 957–961. https://doi.org/10.1109/ICCES48766.2020.9137924
16. Halim Z, Waqar M, Tahir M (2020) A machine learning-based investigation utilizing the in-text features for the identification of dominant emotion in an email. *Knowledge-Based Syst* 208: 106443. https://doi.org/10.1016/j.knosys.2020.106443

17. Jain T, Agrawal N, Goyal G, et al. (2017) Sarcasm detection of tweets: A comparative study. *2017 Tenth International Conference on Contemporary Computing (IC3)*, 1–6, https://doi.org/10.1109/IC3.2017.8284317
18. Biere S, Bhulai S, Analytics MB (2018) Hate Speech Detection Using Natural Language Processing Techniques. Vrije Univ. Amsterdam.
19. Konduri V, Padathula S, Pamu A, et al. (2020) Hate Speech Classification of social media posts using Text Analysis and Machine Learning. Oklahoma State University.
20. Panda S, Kusum (2020) Detecting Twitter's Impact on COVID-19 Pandemic in India. *Int J Innov Technol Explor Eng* 9: 815–819. https://doi.org/10.35940/ijitee.H6718.069820
21. Amer AYA, Siddiqui T (2020) Detection of Covid-19 Fake News text data using Random Forest and Decision tree Classifiers. *International Journal of Computer Science and Information Security IJCSIS* 18: 88–100. https://doi.org/10.5281/zenodo.4427204
22. Shaalan K, Hassanien AE, Tolba F (2018) *Intelligent Natural Language Processing: Trends and Applications*. vol. 740, Springer. https://doi.org/10.1007/978-3-319-67056-0
23. Salloum SA, Al-Emran M, Monem AA, et al. (2018) Using text mining techniques for extracting information from research articles. *Studies in Computational Intelligence* 740: 373–397. https://doi.org/10.1007/978-3-319-67056-0_18
24. Kowsari K, Meimandi KJ, Heidarysafa M, et al. (2019) Text classification algorithms: A survey. *Information* 10: 150. https://doi.org/10.3390/info10040150
25. Nhlabano VV, Lutu PEN (2018) Impact of Text Preprocessing on the Performance of Sentiment Analysis Models for Social Media Data. *2018 Int Conf Adv Big Data, Comput Data Commun Syst icABCD*, 1–6. https://doi.org/10.1109/ICABCD.2018.8465135
26. Dawei W, Alfred R, Obit JH, et al. (2021) A Literature Review on Text Classification and Sentiment Analysis Approaches. *Lect Notes Electr Eng* 724: 305–323. https://doi.org/10.1007/978-981-33-4069-5_26
27. Zhou Z, Guan H, Bhat MM, et al. (2019) Fake news detection via NLP is vulnerable to adversarial attacks. *ICAART 2019 - Proc 11th Int Conf Agents Artif Intell* 2: 794–800. https://doi.org/10.5220/0007566307940800
28. Mansour S (2018) Social media analysis of user's responses to terrorism using sentiment analysis and text mining. *Procedia Comput Sci* 140: 95–103. https://doi.org/10.1016/j.procs.2018.10.297
29. Ahmad I, Yousaf M, Yousaf S, et al. (2020) Fake News Detection Using Machine Learning Ensemble Methods. *Complexity* 2020: 680–685. https://doi.org/10.1155/2020/8885861
30. Sharmin S, Zaman Z (2018) Spam detection in social media employing machine learning tool for text mining. *Proc - 13th Int Conf Signal-Image Technol Internet-Based Syst SITIS 2017*, 137–142. https://doi.org/10.1109/SITIS.2017.32
31. Neeraja M, Prakash J (2016) Detecting Malicious Posts in Social Networks Using Text Analysis. *International Journal of Science and Research (IJSR)* 5: 345–347. https://doi.org/10.21275/v5i6.NOV164091
32. Hussain MG, Hasan MR, Rahman M, et al. (2020) Detection of Bangla Fake News using MNB and SVM Classifier. *2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. https://doi.org/10.1109/iCCECE49321.2020.9231167
33. Neha D, Vidyavathi BM (2015) A Survey on Applications of Data Mining using Clustering Techniques. *International Journal of Computer Applications* 126: 7–12. https://doi.org/10.5120/ijca2015905986
34. Kaur R, Singh S (2016) FULL-LENGTH ARTICLE A survey of data mining and social network analysis based anomaly detection techniques. *Egypt Informatics J* 17: 199–216. https://doi.org/10.1016/j.eij.2015.11.004

35. Sharath KA, Singh S (2013) Detection of user cluster with suspicious activity in online social networking sites. *Proc - 2nd Int Conf Adv Comput Netw Secur ADCONS 2013*, 220–225. https://doi.org/10.1109/ADCONS.2013.17
36. Al Mansoori S, Almansoori A, Alshamsi M, et al. (2020) Suspicious Activity Detection of Twitter and Facebook using Sentimental Analysis. *TEM JOURNAL - Technology, Education, Management, Informatics TEM J* 9: 1313–1319. https://doi.org/10.18421/TEM94-01
37. Rajeswari K, Shanthibala P (2018) SARCASM DETECTION USING MACHINE LEARNING TECHNIQUES. *Int J Recent Sci Res* 9: 26368–26372. http://dx.doi.org/10.24327/ijrsr.2018.0904.2046
38. Chen J, Yan S, Wong KC (2018) Verbal aggression detection on Twitter comment : convolutional neural network for short-text sentiment analysis. *Neural Comput Appl* 32: 10809–10818. https://doi.org/10.1007/s00521-018-3442-0
39. Li Y, Li T (2013) Deriving market intelligence from microblogs. *Decis Support Syst* 55: 206–217. https://doi.org/10.1016/j.dss.2013.01.023
40. Kharde VA, Sonawane SS (2016) Sentiment Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications* 139: 5–15. https://doi.org/10.5120/ijca2016908625
41. Joshi S, Deshpande D (2018) Twitter Sentiment Analysis System. *International Journal of Computer Applications* 180: 35–39. https://doi.org/10.5120/ijca2018917319
42. Rui H, Liu Y, Whinston A (2013) Whose and what chatter matters ? The effect of tweets on movie sales. *Decis Support Syst* 55: 863–870. https://doi.org/10.1016/j.dss.2012.12.022
43. Ghosh D, Guo W, Muresan S (2015) Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. *Conf Proc - EMNLP 2015 Conf Empir Methods Nat Lang Process*, 1003–1012. https://doi.org/10.18653/v1/D15-1116
44. Khodak M, Saunshi N, Vodrahalli K (2018) A large self-annotated corpus for sarcasm. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*
45. Rahman AU, Halim Z (2022) Identifying dominant emotional state using handwriting and drawing samples by fusing features. *Appl Intell* 2022: 1–17. https://doi.org/10.1007/s10489-022-03552-x
46. Halim Z, Ali O, Khan MG (2021) On the Efficient Representation of Datasets as Graphs to Mine Maximal Frequent Itemsets. *IEEE T Knowl Data Eng* 33: 1674–1691. https://doi.org/10.1109/TKDE.2019.2945573
47. Savini E, Caragea C (2022) Intermediate-Task Transfer Learning with BERT for Sarcasm Detection. *Mathematics* 10: 844. https://doi.org/10.3390/math10050844.
48. Halim Z, Rehan M (2020) On identification of driving-induced stress using electroencephalogram signals: A framework based on wearable safety-critical scheme and machine learning. *Inf Fusion* 53: 66–79. https://doi.org/10.1016/j.inffus.2019.06.006



AIMS Press