Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 2860479, 17 pages
https://doi.org/10.1155/2020/2860479

Research Article

Multi-Rule Based Ensemble Feature Selection Model for Sarcasm Type Detection in Twitter

Karthik Sundararajan and Anandhakumar Palanisamy

Department of Information Technology, Madras Institute of Technology, Anna University, Chennai–603202, Tamilnadu, India

Correspondence should be addressed to Karthik Sundararajan; karthi24mar@gmail.com

Received 31 August 2019; Revised 7 November 2019; Accepted 27 November 2019; Published 9 January 2020

Academic Editor: Mario Versaci

Sentimental analysis aims at inferring how people express their opinion over any piece of text or topic of interest. This article deals with detection of an implicit form of the sentiment, referred to as sarcasm. Sarcasm conveys the opposite of what people try to convey in order to criticize or ridicule in a humorous way. It plays a vital role in social networks since most of the tweets or posts contain sarcastic nuances. Existing approaches towards the study of sarcasm deals only with the detection of sarcasm. In this paper, in addition to detecting sarcasm from text, an approach has been proposed to identify the type of sarcasm. The main motivation behind determining the types of sarcasm is to identify the level of hurt or the true intent behind the sarcastic text. The proposed work aims to improve upon the existing approaches by incorporating a new perspective which classifies the sarcasm based on the level of harshness employed. The major application of the proposed work would be relating the emotional state of a person to the type of sarcasm exhibited by him/her which could provide major insights about the emotional behavior of a person. An ensemble-based feature selection method has been proposed for identifying the optimal set of features needed to detect sarcasm from tweets. This optimal set of features was employed to detect whether the tweet is sarcastic or not. After detecting sarcastic sentences, a multi-rule based approach has been proposed to determine the type of sarcasm. As an initial attempt, sarcasm has been classified into four types, namely, polite sarcasm, rude sarcasm, raging sarcasm, and deadpan sarcasm. The performance and efficiency of the proposed approach has been experimentally analyzed, and change in mood of a person for each sarcastic type has been modelled. The overall accuracy of the proposed ensemble feature selection algorithm for sarcasm detection is around 92.7%, and the proposed multi-rule approach for sarcastic type identification achieves an accuracy of 95.98%, 96.20%, 99.79%, and 86.61% for polite, rude, raging, and deadpan types of sarcasm, respectively.

1. Introduction

Natural language processing (NLP) deals with incorporating the computer with the capacity of understanding the language of human beings just like how it is spoken. NLP falls under the domain of artificial intelligence (AI). The major challenge in developing NLP-based systems is that the human language is not always precise due to its complex linguistic structure and the difficulty in interpretation. Slangs, dialects, and the context play a huge role in making it a little difficult for computers to understand. Recent developments in NLP can be attributed to the fact that vast amount of information has been made available due to the huge impact of social media in the recent years. Sentiment can be considered as a viewpoint that an individual might possess. It is an idea or feeling which is held by a person. It is a combination of feelings and opinions which forms the basis for an action. Human sentiments play a major role in their day-to-day activities. Sentiment analysis is a use case of NLP. Sentiment analysis is generally used by data scientists to assess comments available in social media to determine how a particular business or brand is performing. For example, the reviews of a product can be analyzed to modify the shortcomings and identify the areas of improvements required in the product which in turn can lead to a business growth.

Sarcasm can be considered as an implicit form of sentiment. It usually conveys the opposite of what has been intended. Sarcasm is generally associated with irony and satire or wit that is used to refute, insult, make fun of, or amuse. For example, the teacher exclaimed "Kudos to your hard work. I have never been more impressed in my life. Lol!" A plain look at this sentence may reveal that it is an appreciation. However, the context and the body language of a speaker indicate the sarcastic nature of this expression. In the absence of visible expressions, determining sarcasm in a tweet is a challenge. An interesting perspective of sarcasm was provided by Deliens et al. [1] where the analysis was conducted upon two sarcastic conditions: egocentric and allocentric. The former term indicates that the sarcasm was felt or observed only in the participant's perspective and not from addressee's point of view and the latter indicating sarcasm being observed from both participant's and addressee's point of view. The generic interpretation of results conveys that the prosodic features, the ones involving patterns of stress and sounds, are more helpful in detecting sarcasm than contextual features.

Consider the example: "*Oh how I love being ignored #sarcasm*." While this tweet has a hashtag to indicate sarcasm, it is not necessary for a person to always include hashtags to indicate sarcasm. This poses an additional challenge to extract the right features and characteristics of tweets based on which sarcastic tweets can be identified. It is important in areas like sentiment analysis and affective computing since sarcasm can totally flip the polarity of a sentiment even though it may look different. Basic analysis of sentiments from texts might not be efficient to understand the clear motivation due to the presence of various literary devices such as sarcasm, irony, etc. [2]. Hence detecting sarcasm is very essential in order to avoid any sort of misunderstanding in any type of communication and to ensure that meanings intended in the statements are understood as it is. Manually detecting sarcasm can be a tedious process which can be simplified by automated sarcasm detection and analysis. Detecting sarcastic statements has become a vital task in social media applications as it influences the organizations that mine social media information. Despite the presence of various possible features that can be extracted from text, they can be grouped into broad categories, namely, lexical, pragmatic, hyperbolic, and contextual features [3].

The key focus of this research is to categorize sarcasm into various types which helps to understand the level of hurt or intent to hurt that is present in the sarcastic statement. As sarcasm can invoke a wide range of feelings in a person, it can either create a feeling of fun for the receiver or at the worst case, it can even invoke a deep sense of emotional hurt. The application of type detection can be helpful in understanding the emotions behind sarcasm which in turn can even give an insight into the emotional state of the people involved in a sarcastic conversation, i.e., the person who uses sarcasm and the person upon whom sarcasm was intended. Deeper analysis can even give insight into the relationship between the emotional state of a person and the type of sarcasm that he or she employs during that time. Such levels of understanding will improve the process of sarcasm detection. Initially, an ensemble-based feature selection approach has been proposed to identify the optimal set of features needed to detect sarcasm.

The main contribution of this work is the classification of sarcasm into 4 types, in addition to detecting sarcasm. In order to detect and classify, a multi-rule based approach has been proposed. Along with these, an attempt has been made to model the mood change based on the type of sarcasm exhibited by the user. Experimental results showed that the proposed method obtained encouraging results in detecting and categorizing sarcasm. The major objectives of this research work include the following:

(1) To classify sarcasm into various types based on the emotional aspects, thereby determining whether the emotional state of a person influences the type of sarcasm exhibited by them

(2) To determine the optimal set of features essential for sarcastic type classification with proposed ensemble learning algorithm

(3) To propose a multi-rule based approach for classifying sarcasm into various types in order to tackle the problem of vagueness and uncertainty exhibited by natural languages

2. Related Work

2.1. Literature Survey on Feature Extraction. Ravi and Ravi [4] came up with an ensemble text feature selection method in order to identify sarcasm and irony from reviews and news articles. An AUC value of 91.46% for satiric news and AUC of 88.86% for ironic reviews were recorded. Bouazizi and Ohtsuki [5] developed a system for detecting sarcasm and thereby intended to demonstrate that detection of sarcasm increases the performance of sentiment analysis. Khokhlova et al. [6] suggested the Twitie software for tokenizing and tagging microblog text and the Sketch Engine System for classifying words into eight emotions: Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise, and Trust. Rockwell [7] used the speech data utterances and performed acoustic analysis and perceptual coding and found that the former performed better than the latter in discriminating sarcasm and nonsarcasm. Ragini et al. [8] developed an approach for generating disaster response with sentiment analysis. The authors used POS and lexicon features and observed that the accuracy was 30% more than historical methods. El-Masri et al. [9] applied a sentiment analysis tool in Arabic tweets for polarity detection. The authors used machine learning and lexicon methods in order to analyze the polarity of the tweets. Neppalli et al. [10] developed a system for identifying the polarity of tweets during Hurricane Sandy. The system achieved an accuracy of 75.91%, and the major issue was the lack of connection between the disaster and people who tweeted about it. Kim [11] put up a system which uses the relevance theory method to provide more analytic account by adding cognitive explanations. Yoo et al. [12] proposed a system for detecting real-time events from social media. The authors performed sentiment analysis using convolutional networks. The accuracy was about 84%, and as the number of tweets increased, the model observed a gradual decrease in the accuracy.

Musoff [13] used categorical falsity as metaphor and the contradictory as irony/sarcasm to find out the distinction between irony and sarcasm. The author was able to deduce that there was a clear difference between irony, sarcasm, and metaphor. Ren et al. [14] developed a method by dividing the process into two approaches: one by integrating key contextual information (CANN-KEY) and the other by integrating all contextual information (CANN-ALL). It avoided manual feature engineering and used a real-valued word vectors which resulted in F-measures of 56.37 and 62.05, respectively. Lei et al. [15] developed a system to predict emotions by generating emoticon and part of speech (POS) tagging on news articles. The accuracy was found to be 63.57%. Ghaissi and Lee [16] developed a feature selection module which was domain dependent. Vectorized tweets were taken as input which resulted in highly sparse input matrix. It was found that the sentiment analysis increased sensitivity but domain-specific nature of the process made it difficult to generalize beyond the chosen target. Xiong et al. [17] performed a multi-level sentiment analysis and as a result, enriched the method of word embedding as word level information was integrated with tweet level information. Karoui et al. [18] employed supervised learning approach and used 4 features for identifying irony. Reliability of hashtags and pragmatic features had to be ensured, and the accuracy was about 72.36%. Cai et al. [19] developed a framework based on ensemble text feature selection method for detecting sarcasm, irony, and satire from reviews and news articles. The logistic regression method was found to achieve higher accuracy values of 91.46% and 88.86% in detection of satiric and ironic reviews, respectively. Jiménez-Zafra et al. [20] employed a supervised learning and lexicon based sentiment analysis. Kim et al. [21] developed an ensemble regularization method by combining three regression models. Standard misclassification rate was found to be 0.001.

2.2. Literature Survey on Sarcasm Detection. Bharti et al. [22] processed real-time tweets by using Flume and Hive. The authors detected sarcasm by developing a hidden Markov model-based algorithm and MapReduce algorithm. It can be implemented with or without Hadoop Framework where it was found out that the time taken without Hadoop was 11609 s while time taken with Hadoop was 4147 s. Mukherjee and Bala [23] suggested a system which used Naïve Bayes for classification and fuzzy c-means for clustering. Since context words had been used, it was found out to be more effective than methods which used content words alone. The accuracy was found out to be 65%. A small dataset of only 2000 tweets was used; hence, Naïve Bayes worked well. Voyer and Vu [24] developed a model to experiment how negative literal statements would affect sarcasm detection in auditory perception. The approach achieved 80% accuracy in sarcasm identification. Bhan and D'silva [25] suggested a system to measure sarcasm using different algorithms such as Naive Bayes, logistic regression, and linear regression where scores generated for each algorithm were compared to present the most efficient way. The linear SVC model obtained precision, recall, and F-score values of 0.86, 0.87, and 0.86, respectively. Chaudhari and Chandankhede [26] considered a rule-based method for hashtag tokenization, sarcasm detection, and polarity detection which also included statistical, distributional, and deep learning classification techniques. The SVM classifier of statistical approach achieved a precision of 64% and a recall of 39%, while the distributional semantic models achieved a 7%–10% F-score where it was found that there were not enough feature sets to explore datasets. Dharwal et al. [27] made use of various sarcasm analyzing techniques for filtering sarcastic statements from the text.

Gupta et al. [28] came up with a mood swing analyzer using k-means clustering to perform sentiment analysis. The approach attained an accuracy of 85.03%. Lagerwerf [29] developed a system to detect irony and sarcasm from advertisements and public announcements. AL-Sharuee et al. [30] developed a system for sentiment analysis using contextual analysis and unsupervised ensemble learning in order to handle domain-dependency problem. The average accuracy for many datasets was found to be 82.45%. Persicke et al. [31] developed a system for teaching autistic children a method for detecting sarcasm and responding to the same. The author included rules along with video clips for this purpose. Dynel [32] made use of various competing academic approaches to detect sarcasm and irony. Ager [33] developed a method to detect sarcasm based on assumptions about the speaker and the addressed topic. Deliens et al. [1] developed a noncontextual strategy for sarcasm detection. It was found out to be an efficient strategy but was time consuming. Gent [34] used a chip to design the system to cope with huge volume of data which can monitor thousands of twitter accounts for tweets that mention a specific phrase. Energy efficiency was posing a problem. Fernández-Caballero et al. [35] suggested a system to detect the emotional state of patients by analyzing their behavior, physiological signals, and expressions. Porshnev et al. [36] developed a system to analyze the twitter user's mood and thereby predict the stock price movement. Zhang et al., [37] used network regulators in microblog text to analyze the sentiment. The network regulators analyzed public opinions and made decision regarding the sentiment. Xiaomei et al. [38] used sentiment consistency and emotional contagion for sarcasm detection. The developed model was able to outperform baseline methods consistently and significantly but time consumption was high.

Schuch et al. [39] used congruency sequence effect and a hanker interface paradigm and stroop-like interface for identifying conflict adaptation. Song et al. [40] used convolutional networks for sentiment classification framework. It achieved good results when compared to the state-of-the-art techniques. The method heavily relied on the performance of the pretrained saliency detection networks. Fardoun et al. [41] collected information from student moods using RFID tags. It turned out to be time consuming, and effectiveness was also under question. Swami et al. [42] developed an English-Hindi coded dataset for the purpose of sarcasm detection. A total of 5250 tweets were used, and *n*-grams and sarcastic hashtags were used as features. Random forest was used for classification and achieved an accuracy of 78.4%. Rajeswari and ShanthiBala [43] discussed about recognizing sarcastic emotion from individuals. Though information about sarcastic types was listed in the article, implementation and experimentation details were not provided regarding sarcasm detection as well as categorization.

This highlights the importance of the proposed work where the problem of categorizing sarcasm has been attempted, and the results have been provided with experimental evidence.

2.3. Literature Survey on Fuzzy and Rough Set. Tran et al. [44] developed a rough set system from perspective information, and it was observed to perform better than the traditional greedy methods. Tiwari et al. [45] suggested knowledge extraction in framing expert and intelligent systems which transformed fuzzy decision system (FDS) into intuitionistic FDS (IFDS) with a fixed degree of hesitancy. The presence of significant amount of noise was the major issue in IFDS. Hiai and Shimada [46] came forth with a system that classified sarcastic tweets using three-stage judgement process based on rules, boosting rules, and rejection rules. The approach classified tweets into 8 classes which were more promising than the baseline model as the precision rate and recall rate were found to be 0.028 and 0.543. Fang and Hu [47] developed a system based on fuzzy implicators and co-implicators for fuzzy granules. The uncertainty measures, reductions of the granular variable, and the application of theoretical results are yet to be explored deeply. Qian et al. [48] combined the neighborhood and local rough set approaches. Approximation and attribute reduction algorithm was performed with limited labeled dataset. The authors conducted a rough set-based data analysis with limited labeling and evaluated the performance of the approach using various datasets.

Overall, the major limitations of the existing approaches include the following: most of the existing works focus primarily on lexical and syntactic feature-based approaches, lack of efficient techniques to handle the uncertain and vague aspects of natural language data, presence of noises in the raw twitter data, etc.

3. Proposed Approach

The major objective of the proposed work is to identify if a given sentence is sarcastic or nonsarcastic with an optimal set of features, identified by the proposed ensemble feature selection. These features would be leveraged to further classify the sarcastic statement to a specific type with the proposed multi-rule based type classification approach. The results are validated using various classification algorithms. This research has attempted to find the correlation between the types of sarcasm and the possible mood changes. Figure 1 describes the overall architecture of the proposed work.

Given a set of tweets "t," the objective is to classify a particular tweet as sarcastic or nonsarcastic. The sarcastic tweets are then passed on to the proposed multi-rule based classification framework where it will be classified into its appropriate type.

$$\begin{aligned} \mathrm{SD}(\mathrm{t}) &\longrightarrow \{\mathrm{S}, \mathrm{NS}\}, \\ \forall\, \mathrm{S}, \mathrm{MRC}(\mathrm{S}) &\longrightarrow \{\text{Polite Sarcasm, Rude Sarcasm, Raging Sarcasm, Deadpan Sarcasm}\}, \end{aligned} \tag{1}$$

where SD ⟶ sarcasm detection; S ⟶ sarcastic class; NS ⟶ nonsarcastic class; and MRC ⟶ multi-rule based classification framework.

3.1. Data Acquisition and Preprocessing. Twitter API was used for the purpose of data collection. Twitter is one of the most popular social media platforms. People commonly use twitter for sharing their opinion, views, anger, displeasure, and all sorts of opinions or emotions about any event that makes it so popular. Tweets from Twitter are predominantly utilized in various fields of natural language processing applications. Tweets are obtained through Twitter API (Tweepy and Twython). Tweets are extracted on the basis of the following hash tags: #sarcasm, #sarcastic, #Sarcasm, and #notSarcasm. A total of 76,799 tweets are used for experimentation purpose. The tweets that are non-English are filtered out. The data from the twitter as such might be incomplete, inconsistent, and likely to possess many errors. Hence, the raw data obtained need to be cleaned and then transformed into an understandable format for further processing. Hashtags, URLs, and links are removed in preprocessing Algorithm 1 followed by which POS tagging, stemming, and lemmatization are performed to obtain understandable data.

3.2. Feature Extraction. Feature extraction has a huge role in determining the outcome of any machine learning task. The quality of classification, both qualitatively and quantitatively, depends on the features selected. This section, at a high level, focuses on extracting the features from tweets that can be categorized into various types, namely, lexical, hyperbolic, pragmatic, sentiment, and contradiction. Lexical features include *n*-gram, bigram, and unigram which are combination of words that are extracted from the tweets to aid in tokenization. Intensifiers are also identified as they might help in the sarcasm detection process. Pragmatic features like emoticon and smileys are extracted. The proposed system extracts a total of 20 features: noun and verb count,

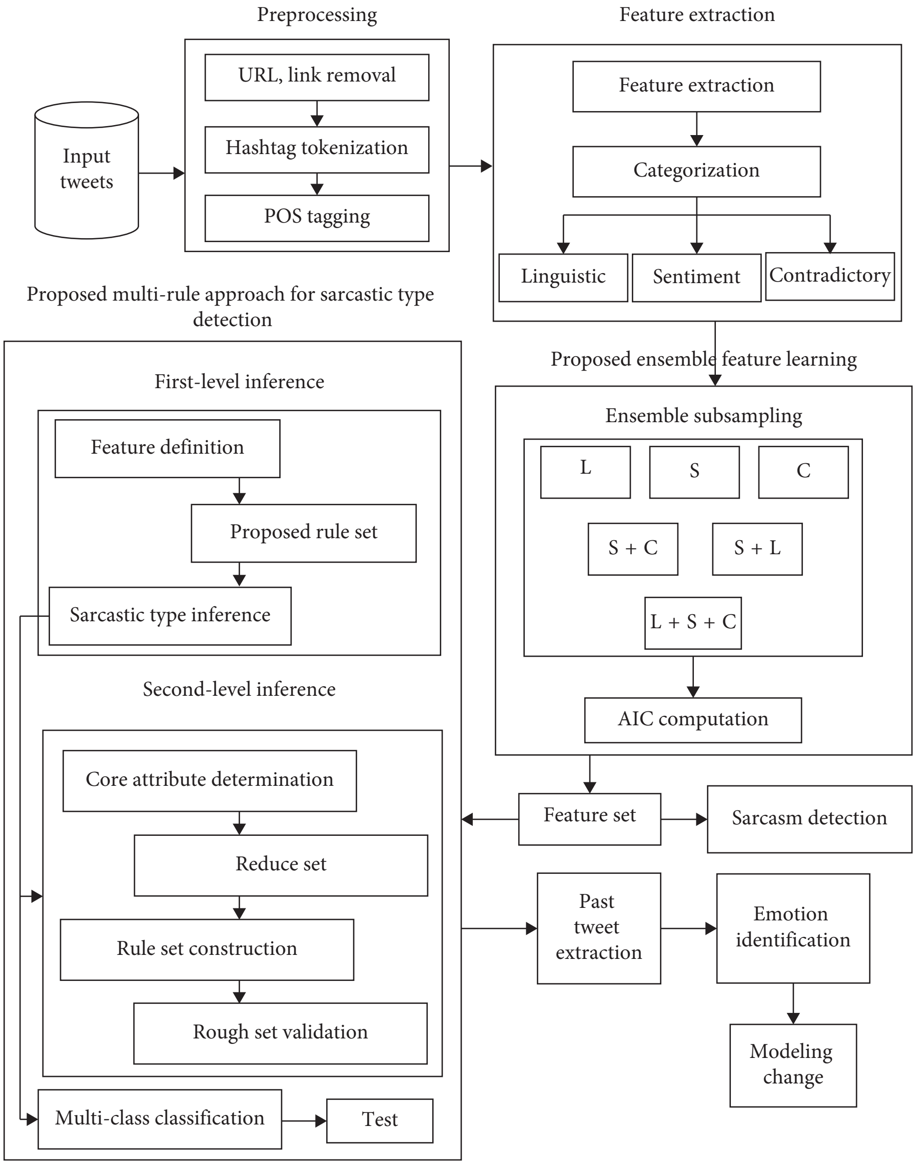


Figure 1: Proposed architecture diagram for sarcastic type detection.

positive intensifier, negative intensifier, bigram, trigram, skip gram, unigram, emoji sentiment, sentiment score, interjections, punctuators, exclamations, question mark, uppercase, repeat words count, positive word frequency, negative word frequency, polarity flip, and parts of speech tagging.

Various sentiment-based features are extracted from the tweets like positive words frequency—total number of positive words; negative words frequency—total number of negative words; positive intensifiers—intensifiers exhibiting positive emotion; negative intensifiers—intensifier exhibiting negative emotion; *n* grams—set of consecutively occurring “*n*” words($n = 1$(unigram); $n = 2$(bigram), etc.); and skip gram—*n* grams with an additional factor called skip distance; passive aggressive count gives the indirect expression of hostile intention; sentiment score gives the sentiment value in which a “−1” indicates negative sentiment and a “+1” indicates a positive sentiment; emoticon sentiment gives the polarity of the emoticon; polarity flip gives the reverse polarity of sentiment; the co-occurring terms need not be consecutive as the tokens can be skipped based on the skip distance value; noun and verb counts can be obtained from POS tagging of a tweet; POS Tagging is a way to tag each word present in the tweet with its appropriate parts of speech; exclamations and question marks are most meaningful among the various punctuators for detecting sarcasm. Uppercase words are extracted as features because sometimes people use capital lettered words to stress on the things that they want to convey strongly. These are the prominent set of features which will be useful for sarcasm detection. Once these features are extracted, a numerical value for the features is obtained. These extracted features are categorized into different groups such as linguistic, sentiment based, and contradiction based feature sets. Table 1 represents a sample of trigram feature extracted from the input file. Once the features are extracted, they are then

```
Input: Comma separated raw tweet file
Output: Preprocessed and clean tweet file
Create a list preprocessed_docs
for each tweet S in T:
    create a list result
    for each word W in S:
        if W not in STOPWORDS:
            Add stem (W) to result
        end if
        if W[0] = #
            Tokenize W
        end if
        if W has @, /, RT, http
            Remove the word and then add to result
        end if
    end for
    preprocessed docs.Append (result)
end for
end
```

Algorithm 1: Preprocessing of raw twitter data.

Table 1: Sample *n*-gram and skip gram feature extraction.

Unigram	{'it'}	{'was'}	{'supposed'}	{'to'}	{'be'}	{'a'}	{'joke'}	
Bigram	{'it', 'was'}	{'was', 'supposed'}	{'supposed', 'to'}	{'to', 'be'}	{'be', 'a'}	{'a', 'joke'}		
Trigram	{'it', 'was', 'supposed'}	{'was', 'supposed', 'to'}	{'supposed', 'to', 'be'}	{'to', 'be', 'a'}	{'be', 'a', 'joke'}			
1-skip 3-grams	{'it', 'was', 'supposed'}	{'it', 'was', 'to'}	{'was', 'supposed', 'to'}	{'was', 'supposed', 'be'}	{'supposed', 'to', 'be'}	{'supposed', 'to', 'a'}	{'to', 'be', 'a'}	{'to', 'be', 'joke'}

passed on to the proposed ensemble feature selection module, where the optimal set of features for detecting sarcasm will be identified. Algorithm 2 explains about the extraction of aforementioned features in a brief manner.

3.3. Proposed Ensemble Feature Selection. This section aims at identifying the optimal set of features from the extracted feature set that can help in identifying a sarcastic tweet. Ensemble learning aims at finding and selecting the best set of features that will aid in accurate sarcasm detection. Various classification algorithms are also leveraged in identifying the optimal set of features that is sufficient enough to identify a sarcastic tweet. Once the features are extracted, sarcasm is detected by training and testing for individual features. The results of classification for individual features are compared. The features are then grouped into various categories such as linguistic features, contradictory features, and sentiment-based features. Ensemble is done for each category of features. Further ensemble is carried out with combination of categories of features. Models are obtained by training using the various classifiers like Random Forest, Naive Bayes, Support Vector Machine, K-Nearest Neighbor, Gradient Boosting, AdaBoost, Logistic Regression, and Decision Tree.

The Classification algorithms listed here are chosen based on the detailed analysis of literatures involving classification tasks. These algorithms have exhibited better performances, and after various experiments, the results obtained by these algorithms were found to be better. Multiple classification algorithms help in validating the performance of the sarcasm classification. Initially, the dataset is annotated with a “1” for all the sarcastic tweets and a “0” for all the nonsarcastic tweets. The annotation has been carried out manually and with the help of hashtags too. The annotated dataset after preprocessing is passed on to feature extraction module. Once the features are extracted, it is passed over to the classification module where different classification algorithms were applied to detect sarcasm. The aforementioned features were fed as input to classification algorithms, and based on those features, a tweet has been classified into sarcastic and nonsarcastic. The performance of different algorithms for classifying sarcasm has been discussed in the result section. Once the tweets are classified into sarcastic and nonsarcastic, feature ensembling has been proposed to identify the optimal set of features for classifying sarcasm into various types. The best set of features that provide a better accuracy for the aforementioned classifiers is selected for determination of type of sarcasm (Algorithm 3).

The results obtained in the previous step for feature set model selection are validated by Akaike information criterion (AIC). AIC is a measure to compute the quality of a particular model with respect to every other model under

```
Input: Preprocessed tweets
Output: Set of features
  (1) Read the Data from the file
  (2) Create empty lists for each feature that had to be extracted.
  (3) For each tweet, $t_i$, do the following:
        //Extracting various features
    3.1 word = nltk.pos_tag(indiv_tokens)
    3.2 nouns = ['NN', 'NNS', 'NNP', 'NNPS']
    3.3 verbs = ['VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ']
    3.4 if word in nouns then
            increment noun_count
    3.5 else if word in verbs then
            increment verb_count
    3.6 return the normalized sum values of verbs and nouns.
    3.7 Initialize pos_int and neg_int to 0
    3.8 sent_id = SentimentIntensityAnalyzer()
    3.9 for index in range of tokens
    3.10 if tokens in intensifier_list:
            compute sent_id.polarity_scores(tokens)
    3.11 if score is negative then
            increment neg_int count
    3.12 else
            increment pos_int count
    3.13 return pos_int, neg_int
    3.14 Initialize sk_value = 0
    3.15 var = [$x$ for $x$ in nltk.skipgrams(token, $n$, $j$)
          //$n$ is the degree of the $n$ grams & $j$ is the skip distance
    3.16 for $i$ in range(len(var)):
    3.17 for $j$ in range($n$):
    3.18 word = sid.polarity_scores(var[i][j])
    3.19 if word corresponds to positive
            increment the sk_value
    3.20 else
            decrement the sk_value
    3.21 return sk_value
    3.22 Read a tweet from the input data set
    3.23 Load the dictionary containing popular emojis
    3.24 for 'i' in emoji_list:
    3.25 if $i$ in tweet:
    3.26 update the emoji list and increment sentiment value based
            on the total occurrence of that particular emoji
    3.27 return the normalized emoji_sentiment value
    3.28 Initialize the interjection counter to 0
    3.29 Load the file containing list of interjections for interjections in the list
    3.30 if tweet contains the corresponding interjection
    3.31 update the interjection count
    3.32 for every word in tokens
    3.33 if word.isupper()
            increment uppercase count
    //Apply regular expression to find out repeating letters
    3.34 result = re.compile(r'(.)\1*')
    3.35 for text segment in result (repeating letters)
    3.36 if length of text segments exceeds 3 //minimum 3 consecutive
            occurrence of same letter
               Increment the repeated words count
    3.37 Initialize pos_count, neg_count, flip_count to 0
    3.38 for words in tokens:
    3.39 sent_score = sent_id.polarity_scores(words)
    3.40 if the score obtained is negative then
            Increment neg_count
    3.41 if the previous word encountered is positive then
```

Algorithm 2: Continued.

```
            Increment flip_count value
      3.42 if the score obtained is positive then
            Increment pos_count
      3.43 if the previous word encountered is negative then
            Increment flip_count value
      3.44 return pos_count. neg_count, flip_count
      3.45 punct = punctuations_counter(tweet, [‘!’, ‘?’, ‘. . .’])
      3.46 return exclamation.append(punct[‘!’])
      3.47 return questionmark.append(punct[‘?’])
    (4) Extract the features and append the features to the lists that was created initially.
```

Algorithm 2: Feature selection.

```
Input: Feature set
Output: Optimal set of features
  (1) For feature $x_i$ in $x_1, x_2, \ldots, x_n$
        a. Read feature $x_i$ into the array named X
           $X = \{x_1, x_2, x_{3,} \ldots x_i\}$
        b. Read the target variable into array named Y
        c. Set the train–test split ratio
           Train_$r$ = 0.8
           Test_$r$ = 0.2
        d. Fix the initial seed for random generator in train and test
           random_state = $n$
        e. Split the data set into $x$_train, y_train, y_train and y_test using train–test split ratio and random_state
        f. Train and the classifier for feature $x_i$ and target
        g. Compute accuracy using
           Accuracy = (TP + TN)/(TP + FP + TN + FN)
        h. Precision is computing by
           Precision = TP/(TP + FP)
        i. Calculate Recall rate using
           Recall = TP/(TP + FN)
        j. Find F-score using
           F-score = 2 ∗ (Recall ∗ Precision)/(Recall + Precision)
        k. Repeat steps c through j by setting different values of train test split ratio
  (2) Combine features into categories $C_1$ (Linguistic features (L)), $C_2$ (Sentiment features (S)) and $C_3$ (Contradictory features (C))
        a. For feature category $C_i$ in $C_1$, $C_2$ and $C_3$
        b. Repeat steps a through k
  (3) Combine categories of features (L + S), (S + C), (L + C) and (L + S + C)
  (4) For each category combination $C_i$ in (L + S), (S + C), (L + C) and (L + S + C)
        a. Repeat steps a through k
  (5) Repeat steps 1 through 4 for different types of classifier
  (6) Select the feature, category, or category combination that gives high accuracy
```

Algorithm 3: Proposed ensemble feature selection.

consideration. AIC is based upon the theory that a model would be of higher quality if there is a minimal information loss. If the AIC value for a particular feature is less, then it is more prominent. The AIC value is computed by

$$\mathrm{AIC} = 2k - \ln(\mathrm{L}), \tag{2}$$

where k represents the number of estimated parameters in the model and likelihood function is denoted by L. Assume that there are “R” models and the Akaike information criterion values of each models are given by AIC_1, AIC_2, AIC_3, ..., AIC_R. Then, the expression

$$\exp\left(\frac{(\mathrm{AIC}_{\min} - \mathrm{AIC}_i)}{2}\right), \tag{3}$$

denotes the probability that the i^{th} model might minimize the information loss.

This concept is applied in the selection of the features and the classifier for detection of sarcasm (Algorithm 4).

3.4. Proposed Multi-Rule Based Sarcastic Type Detection. This section involves developing a fuzzy system based on the part of the features identified in the previous module in order

```
Input: Accuracy of various classifiers
Output: Akaike information criterion
  (1) For each feature and for every classifier, read the accuracy obtained by each classification algorithm
  (2) Compute AIC using the equation
      a. 2k-2ln (accuracy) where $k = 1$ in this case
  (3) Select the classifier with the lowest AIC computed for every feature category; let the lowest be $\mathrm{AIC}_{\min}$
  (4) Compute information loss for the various classifiers using the formula
      Exp ($\mathrm{AIC}_{\min}$−AIC value)/2
  (5) Select the classifier and feature with information loss close to 1 (i.e., the model that can reproduce compared to the given model
with very minimal information loss)
  (6) Repeat Steps 1 through 5 for combinations of feature categories
```

Algorithm 4: AIC computation.

```
Input: Feature Set
Output: Type of Sarcasm
  (1) For every fuzzy input variable $f_i$ in selected feature set, F
        a. Define fuzzy set $F(s)$ corresponding to the crisp values
        b. Define the membership functions $m(f)$
           $\mathrm{Triangle}(X{:}\ a,b,c) = \begin{cases} 0 & \text{if } x \le a \\ (x-a)/(b-a) & \text{if } a \le x \le b \\ (c-x)/(c-b) & \text{if } b \le x \le c \\ 0 & \text{if } c \le x \end{cases}$
  (2) Define fuzzy set $F(s)$ for the consequent i.e. Type of Sarcasm
  (3) Define membership function $m(x)$ for the output variable
  (4) For $f(s) \in F(s)$ in type of sarcasm
        a. Define fuzzy rules $r_1, r_2, \ldots, r_n$
  (5) Perform a control system simulation using the defined rules $r_1, r_2, \ldots, r_n$
  (6) For tweet $\mathrm{t}_i \in$ list of tweets T
        a. Compute the fuzzy value of type of sarcasm
        b. Compute the defuzzified value using centroid method
             $Z^* = \int \mu_B(Z) \cdot Z\mathrm{d}Z / \int \mu_B(Z) \cdot \mathrm{d}Z$
  (7) Defuzzified value will determine the type of sarcasm to which each text belongs to.
```

Algorithm 5: Fuzzy algorithm for sarcastic type detection.

```
Input: Feature Set
Output: Type of Sarcasm
  Step 1: Load data in isf format
  Step 2: Replace missing values if any
  Step 3: Discretize the data
  Step 4: Generate Cut set/Reducts
  Step 5: Generate Decision Rules using Minimal Extended Covering
  Step 6: Classify data using the decision rules
```

Algorithm 6: Rough set algorithm.

```
Input: History of tweets.
Output: Polarity and Mood
  (1) For every sarcastic tweet, $\mathrm{s}_{i,}$ in Tweet set, T
        1.1. Extract the past tweets (p) of the user, $u_i$
        1.2. Select the preceding ((T − 1), (T − 2), (T − 3), . . . , (T − 10)) and succeeding tweets (T1, T2, T3, . . . , T10)
        1.3. Set $\mathrm{s}_i$ as T0
  (2) For every identified tweet in step 1.2
        2.1. Find the sentiment of every tweet (t)
```

Algorithm 7: Continued.

```
        2.1.1. If sentiment score = 0, set polarity as neutral
        2.1.2. Else if sentiment score >0, set polarity as Positive
        2.1.3. Else set polarity to negative
(3) Determine the level of mood change of the various users for varying types of sarcasm
```

Algorithm 7: Modelling the user's Mood changes.

to classify the type of sarcasm for every sarcastic tweet. Fuzzy module estimates how much a particular sarcastic tweet belongs to a particular type of sarcasm based on the various membership functions defined as part of fuzzy rule-based system (FRBS). Rough set theory deals with vague, imprecise, and inconsistent data. After developing a fuzzy system, the rough set model is developed to determine the sufficiency and effectiveness of classification of type of sarcasm.

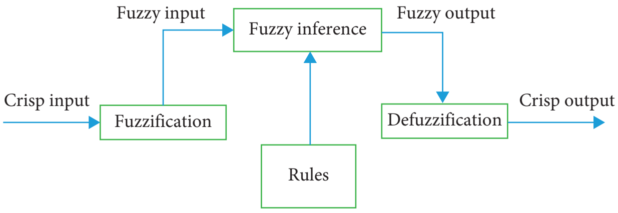


Figure 2: Fuzzy system architecture.

3.4.1. Fuzzy Rule System. Fuzzy logic deals with systems with ambiguity or vagueness. Real-world applications may not be suited for binary solutions always. There may be instances where approximate logic would be preferred to crisp logic. It is in such cases that fuzzy logic exhibits its usefulness. The specialty of fuzzy logic is that it handles uncertainties to a good extent. It employs a "degree of truth" to all factors rather than plain "true" or "false" cases. Fuzzy associates a membership value to every element in the set. Consider a statement which says "The probability of a bottle containing wine is 0.7" which usually employs that there is a 70% chance that there is a bottle that contains wine. Now consider the statement which goes like "The membership value of a bottle containing wine is 0.7" which means that the bottle will contain wine up to 70% of its entire volume. This is basically the difference between fuzzy logic and probability.

A fuzzy set, A, can be represented as

$$A = \{(x, \mu_A(x)) \mid x \in \mathrm{U}\}, \tag{4}$$

where "x" is any element, "μ" represents the membership function, and "U" denotes the universal set.

Fuzzy logic accomplishes task with fuzzy sets which represents a linguistic variable. Initially, a set of input and output variables are identified. Once the variables are identified, then linguistic variables are set for all the input variables. Linguistic variables contain linguistic hedges which signify various approximate states of the corresponding input and output. Once the variables are set, then the fuzzification process is performed with the help of membership functions which represents the fuzzy set in a graphical form. Then rules are formulated. Fuzzy logic is governed by IF-THEN rules. Fuzzy Inference System then evaluates the rules and finally performs the defuzzification task using one or more defuzzifying techniques. The aforementioned processes are described in Figure 2.

The various steps involved in defining a fuzzy logic system is described below (Algorithm 5):

Fuzzificiation: the process of converting crisp inputs into fuzzy inputs

Rule engine: this involves the development of IF-THEN rules that operate on the fuzzy variables

Inference engine (also known as the controller): this step involves the execution of various fuzzy rules and determining the associated membership based on the input values

Defuzzification: the process in which the resulting fuzzy output is converted into crisp outputs.

3.4.2. Fuzzy Rule-Based Sarcastic Type Detection. In this work, a fuzzy rule-based system is developed with the features identified in the previous module to determine the type of sarcasm. In this work, we have attempted to classify sarcasm into four basic types: Polite, Rude, Raging, and Deadpan.

Polite is a form of sarcasm where the sarcastic sentence is more positive in nature. For example, "Wow, I love being ignored!!"

Rude is a form of sarcasm in which the sarcastic sentence contains moderately negative implications. For example, "I can explain it to you but I cannot understand it for you!"

Raging is an extension of rude sarcasm but is highly negative in the way of expression. For example, "Person1: You made me feel like the most horrible person in the world. Person 2: Yeah I made you feel like yourself."

Deadpan is a type of sarcasm which is implicit in nature and is difficult to say if it is positively or negatively implied. For example, "I am confused in a good way!"

Based on the proposed ensemble learning method, the following set of features is extracted as the best set of features for sarcasm detection and are utilized for determining the type of Sarcasm. The features identified are as follows:

(i) Positive intensifier
(ii) Negative intensifier
(iii) Positive word count
(iv) Negative word count
(v) Sentiment score

(vi) Emoji sentiment

These features are set as the antecedents in fuzzy rules. The fuzzy set for these terms is defined. Figure 3 represents the fuzzy definition (membership function) for one of the input variables.

Type of sarcasm is set as the consequent and can take one of the following values: Rude, Raging, Polite, and Deadpan. Figure 4 describes the membership function for one of the output variables.

Membership functions are defined for the various fuzzy variables. After defining the membership functions with features as inputs and type of sarcasm as output, fuzzy rules are constructed. Examples of rules that are constructed for evaluation are shown below:

rule7 = ctrl.Rule(PositiveIntensifier['High'] &NegativeIntensifier['Low'] & (PositiveWordCount['VeryHigh']| PositiveWordCount['High']|PositiveWordCount['Medium']| PositiveWordCount['Low']|PositiveWordCount['VeryLow']) &(SentimentScore['VeryHigh']) &NegativeWordCount['Low'] &EmojiSentiment['Medium'], TypeOfSarcasm['Polite'])

rule24 = ctrl.Rule((NegativeIntensifier['Medium']|NegativeIntensifier['Low']) & (PositiveWordCount['Low']| PositiveWordCount['VeryLow']) & (NegativeWordCount['Low']|NegativeWordCount['VeryLow']|NegativeWordCount['Medium']) &SentimentScore['Low'] &EmojiSentiment['High'], TypeOfSarcasm['Rude'])

The rules described above are sample rules for detecting polite sarcasm and rude sarcasm. For every sarcastic tweet, the controller evaluates the input values with various linguistic hedges associated with membership function to decide the output. The output obtained is defuzzified to in order to get the crisp output, which is the type of sarcasm in this case.

Figure 5 shows an example of an output determined using the fuzzy logic system for a particular input combination.

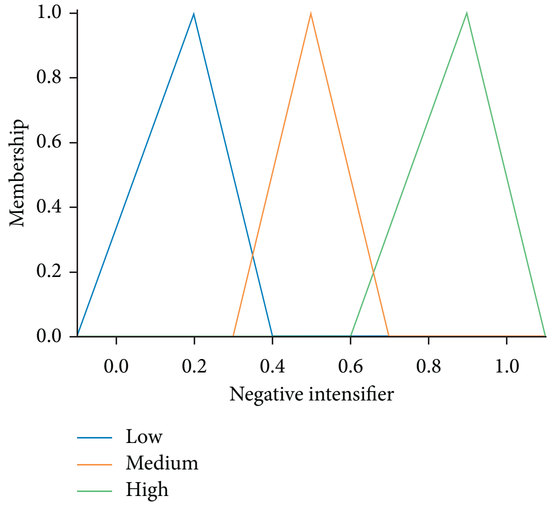


Figure 3: Fuzzy input variable.

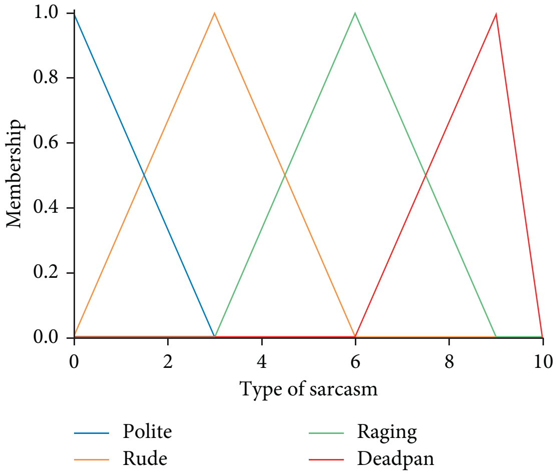


Figure 4: Fuzzy output variable.

3.4.3. Rough Set-Based Sarcastic Type Detection.

In addition to the fuzzy rule-based type detection, the proposed approach employs rough set rule set-based validation in order to strengthen and validate the classification process. Rough set approach is generally applied for the problems which has inadequate or approximate information and to perform decision making over it. It can be applied in order to obtain knowledge abstraction followed by which decision can be arrived upon. It involves developing an association rule generator that searches through the entire dataset and finds the rules that are sufficient to reveal the nature and frequency of relationship between the various data elements. Rough set-based knowledge discovery has certain important merits as in discovery of hidden patterns in the dataset, finding the relationships which cannot be found using statistical methods, finding out a set of data that is minimal and adequate enough for classification (data reduction), obtaining a set of decision rules from data, etc.

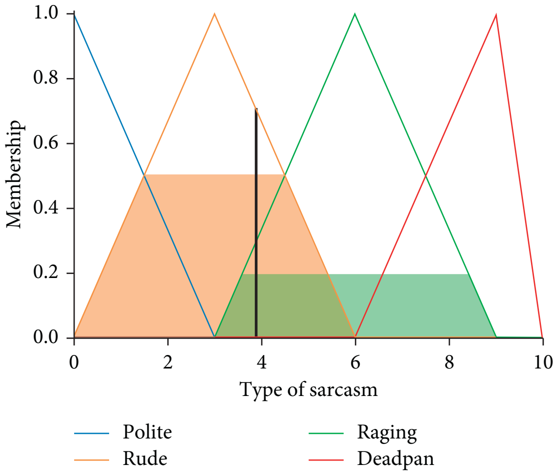


Figure 5: Sample output based on fuzzy rules.

	A1	A2	A3	A4	A5	A6	D1[D]
1	0	0	0.15	0.38	0.71	0	Raging
2	0	0	0	0.12	0	0	Raging
3	0	0	0.08	0.12	0.44	0	Raging
4	0	0	0.08	0.12	0	0	Raging
5	0	0	0.15	0.12	0.46	0	Raging
6	0	0	0.23	0.25	0.9	0	Rude
7	0	0	0.08	0.38	−0.34	0	Raging
8	0	0	0.08	0.12	0.64	0	Raging
9	0	0	0	0.12	0	0	Raging
10	0	0	0.08	0.12	0.53	0	Raging
11	0	0	0	0.12	0	0	Raging
12	0	0	0.31	0.25	0.71	0	Rude
13	0	0	0.08	0.12	0.64	0	Raging
14	0	0	0	0.12	0	0	Raging
15	0	0	0.08	0.12	0.49	0	Raging
16	0	0	0	0.25	−0.32	0	Raging
17	0	0	0.08	0.12	0.3	0	Raging
18	0	0	0	0.12	0	0	Raging
19	0	0	0.08	0.12	0	0	Raging
20	0	0	0	0.12	0	0	Raging

Figure 6: Sample snapshot of rough set attributes and decision making.

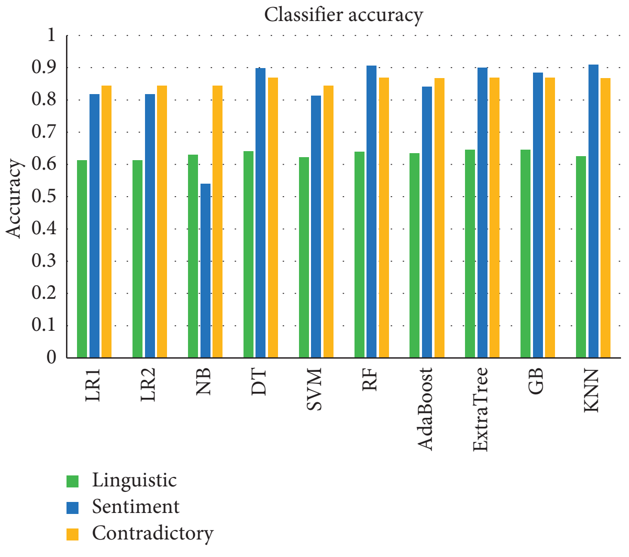


Figure 7: Classification accuracy of various models for feature set categories.

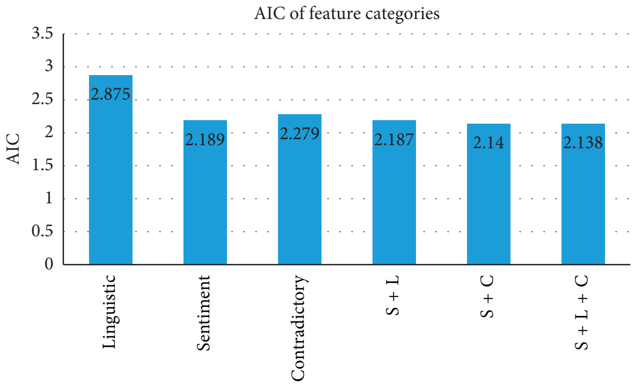


Figure 8: AIC value for feature set categories.

Assume a class of data, C; the definition of rough set for this class C is given by two approximations: lower approximation of C and upper approximation of C. Rough sets involve determining the minimal set of features or attributes which are sufficient enough and more prominent for representing knowledge. This reduced attribute set is referred to as a reduct set. The attribute set common to all reducts is referred to as the core. Rough set decision making involves identification of minimal set of rules that are sufficient enough for modelling the system.

The fuzzy rule set-based classification has been validated based on the rough set approach. The type of sarcasm as predicted by the fuzzy rules and the feature set from the rules act as inputs for rough set analysis. The implementation is performed in ROSE2. As an initial step, the input data file is converted into.isf format in ROSE2. There were no missing values in the dataset. Figure 6 shows the feature attributes and the decision attributes applied for analysis.

In Figure 6, columns A1 to A6 represent the features selected out of the proposed ensemble learning and last column denotes the result of the classification. Reducts are identified on the dataset and the rules are generated based on

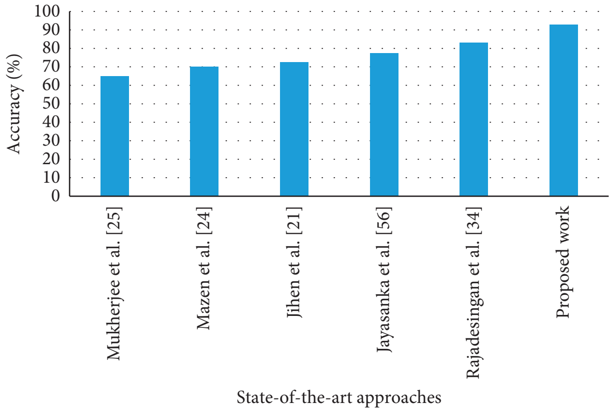


Figure 9: Comparison of the proposed system with existing approaches.

Table 2: Comparison of the proposed approach with the state-of-the-art approaches.

State-of-the-art approaches	Accuracy (%)
[23] Mukherjee and Bala	65
[9] El-Masri et al.	70
[18] Karoui et al.	72.36
[49] Jayasanka et al.	77.28
[50] Rajadesingan et al.	83.46
Proposed work	92.7

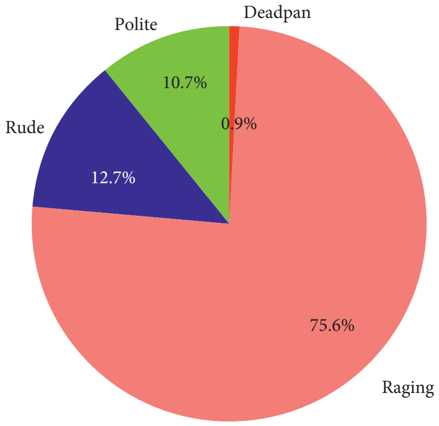


Figure 10: Sarcastic type classification based on proposed multi-rule approach.

extended minimal covering. Classification of data is carried out using the generated decision rules. Clustering analysis is carried out on the type of sarcasm dataset obtained from the fuzzy logic rule set. This is used in validating the correctness of classification and the assumption made on the number of clusters in the data set (Algorithm 6)

3.4.4. Modelling User's Mood Change. This section deals with the study of emotion or sentiment associated with the past tweets of user before and after a sarcastic tweet. The mood change is predicted to show how a sarcastic statement influences or reflects the mood of the person and vice versa. An attempt has been made in this work to model the change in mood based on the type of sarcasm. To predict mood change based on sarcastic type, a series of history of tweets for various users are extracted. The mood of the user is identified based on the past tweets of a user, and the mood changes for various types of sarcasm are studied. The changes in mood based on sarcastic type are modelled as a graph for individual people. The rest of this section addresses how the aforementioned steps for mood analysis of tweets are implemented (Algorithm 7).

Table 3: Confusion matrix of the proposed approach.

Type	Polite	Rude	Raging	Deadpan
Polite	3290	46	85	7
Rude	66	4018	87	6
Raging	24	23	24725	5
Deadpan	2	6	31	263

Table 4: Classification accuracy obtained by the proposed rough set approach.

	Average accuracy (%)	
	Correct	Incorrect
Total	98.81	1.19
Polite	95.98	4.02
Rude	96.2	3.8
Raging	99.79	0.21
Deadpan	86.61	13.39

The implementation of the mood analysis of tweets is carried out with NLTK and TextBlob. A sample set of sarcastic tweets for different types of sarcasm is selected. The past tweets of the user associated with these tweets are extracted. Feature extraction and classification of type of sarcasm, as described in the previous sections, are performed. For individual users, the history of tweets is extracted from either using the username or the user_id of the user. The polarity and sentiment score for these tweets are analyzed using TextBlob and NLTK. The change in mood

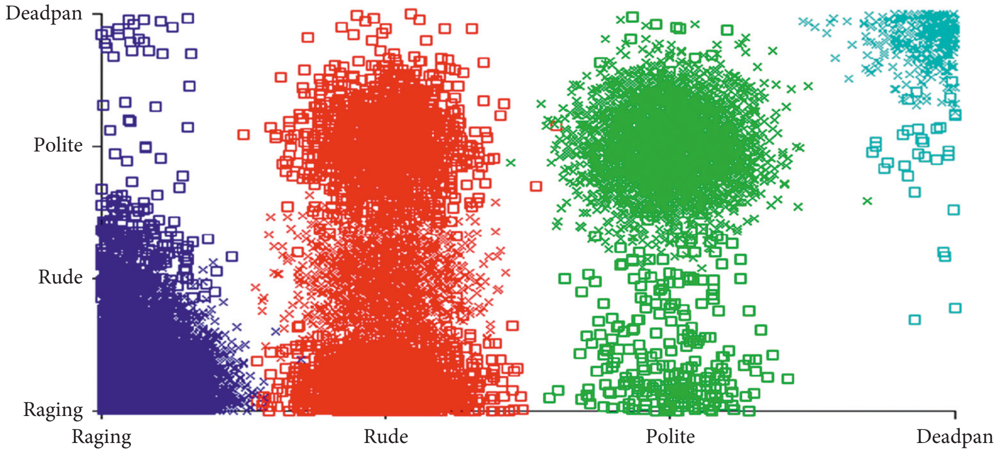
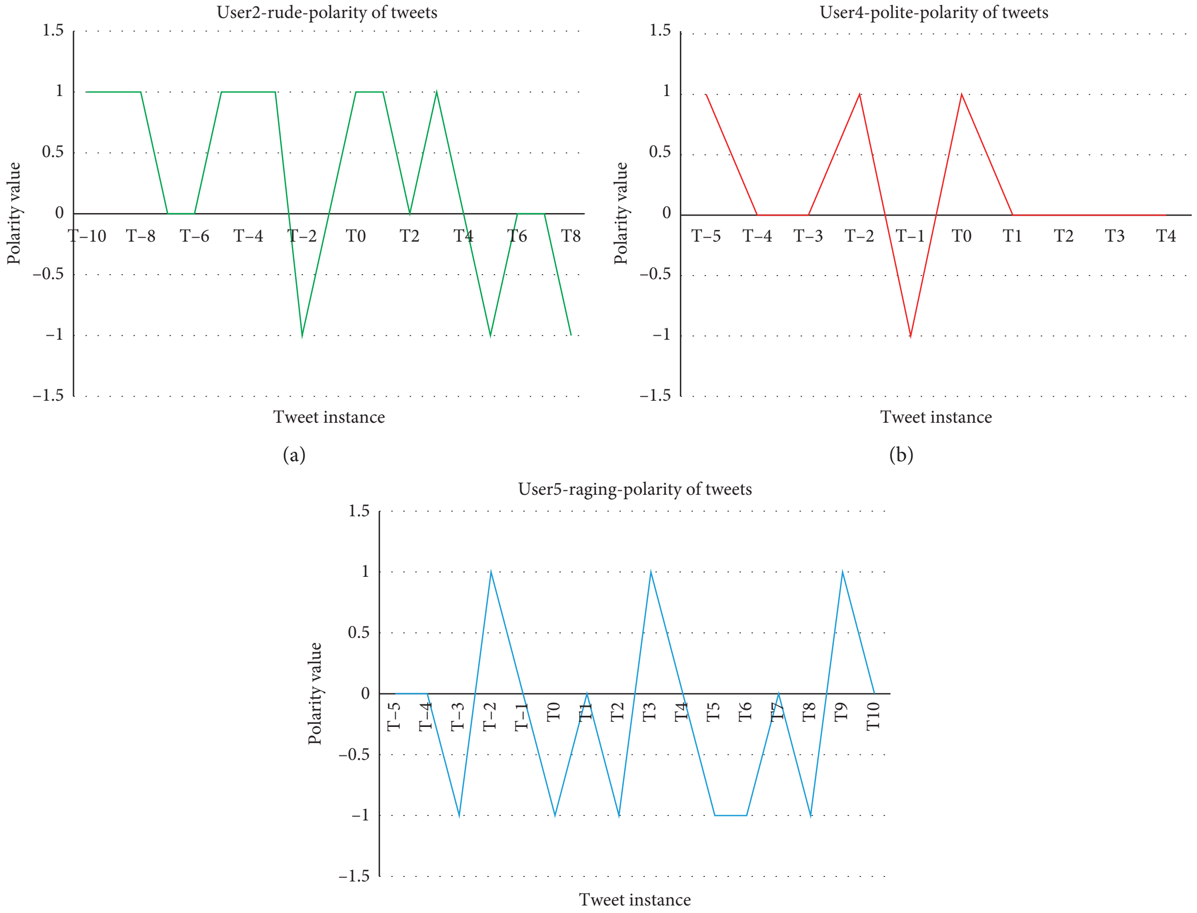


Figure 11: Cluster analysis of type of sarcasm.

Table 5: Performance measure of the proposed system.

Class	TP-rate	FP-rate	Precision	Recall	F-measure
Raging	0.983	0.231	0.93	0.983	0.955
Rude	0.809	0.017	0.729	0.309	0.434
Polite	0.931	0.041	0.730	0.931	0.818
Deadpan	0.913	0.001	0.869	0.913	0.890
Weighted average	0.891	0.181	0.882	0.891	0.874



(a)

(b)

(c)

Figure 12: Representation of mood changes of users. (a) User 2 polarity of tweets: Rude. (b) User 4 polarity of tweets: Polite. (c) User 5 polarity of tweets: Raging.

is modelled as positive or negative change. The fluctuation in mood is predicted using the change and modelled as graph for individual person.

4. Results and Discussion

Around 76,799 tweets were used for experimentation purpose. The data are split into training and testing with 80 : 20 ratio. For the purpose of training, all the sarcastic tweets are annotated with a 1 and nonsarcastic tweets are annotated with a 0. Once preprocessing of tweets is done as mentioned in Section 3.1, feature extraction is carried out. Features that were extracted include bigram, trigram, skip gram, missing values, positive intensifier, negative intensifier, positive word count, negative word count, emoji sentiment, etc. The main goal of the proposed work is to get an optimal feature set for detecting sarcasm from the text input. The features were trained and tested to evaluate its performance. Further, the features were grouped into three categories, namely, textual, emotion based, and contrast based. Each category of feature is ensembled by using different classifiers such as Support Vector Machines, Gradient Boosting, Logistic Regression, AdaBoost, Random Forest, K-Nearest Neighbour, Naive Bayes, Decision Tree, and Bagging. It is found that sentiment-based features provide better predictability of sarcasm followed by the contradictory feature set as shown in Figure 7. It is observed that Random Forest yields better accuracy for various features.

From Figure 7, it is clear that out of all feature categories, sentiment feature category (emotion) and the contradictory feature set achieve highest accuracy value. Hence, the optimal feature set among the extracted feature set comprises of sentiment features and contradictory features. This helps to ascertain the correctness and comprehensiveness of selected features identified by ensemble learning for sarcasm detection. The AIC value calculation, as shown in Figure 8, demonstrates that the combination of sentiment and contradictory features has the least AIC value, thereby leading to inference that these features can better model sarcasm detection. S denotes sentiment feature set; L denotes linguistic feature set; and C denotes contradictory feature set. The performance of the proposed system is compared with few of the baseline approaches.

Figure 9 shows the accuracy of the proposed system compared with existing baseline systems. It can be concluded from the graph that the proposed system for sarcasm detection yields a better accuracy than the baseline models.

In Figure 9, the state-of-the-art techniques are compared with the proposed approach for detecting sarcasm. The classification algorithms that are discussed in the above approaches are listed as follows: Naïve Bayes algorithm with fuzzy clustering [23]; maximum entropy and Naïve Bayes approach [49]; Naïve Bayes algorithm [9]; random forest classification [18]; and scuba framework [50]. The comparison of the existing approaches with the proposed work is tabulated in Table 2.

Figure 10 shows the different types of sarcasm detected by the proposed multi-rule based approach. In the proposed method, sarcasm was classified into four types, namely, Polite, Rude, Deadpan, and Raging.

Rough set-based analysis of type of sarcasm is performed on the results obtained by the fuzzy system, and the confusion matrix obtained by the rough set is tabulated in Table 3.

As seen from the confusion matrix presented in Table 3, the misclassification of types is very minimal across the four types under study. Table 4 shows the accuracy obtained through the rough set approach-based validation. As inferred from the confusion matrix, accuracy of classification is high for all the types of sarcasm (Polite—95.98%, Rude—96.2%, Raging—99.79%, and Deadpan—86.61%). The misclassification rate for each type of sarcasm is minimal too with 4.02%, 3.8%, 0.21%, and 13.39% for polite, rude, raging, and deadpan types of sarcasm, respectively.

Based on the result obtained, a clustering analysis was performed on the sarcastic types and is represented in Figure 11. Performance measure of the proposed system is tabulated in Table 5. Precision, recall, and F-measure were calculated based on equations mentioned in Algorithm 3. It can be inferred that the overall accuracy of 89.1% in rough set analysis affirms the performance of the fuzzy rule base developed for type of sarcasm.

Polarity analysis of preceding and succeeding tweets of a sarcastic tweet for different users is studied, and the results obtained are presented as figures. Figures 12(a)–12(c) depict the change in mood level obtained for users for different types of sarcasm obtained. T0 represents the sarcastic tweet under consideration and the past tweets of T0 are represented as (T−1), (T−2), etc.

The tweets following sarcastic tweet are named as (T1), (T2), (T3), etc. The polarity analysis is carried out for the preceding and succeeding tweets for different users and different types of sarcasm. For polite sarcasm, the polarity of tweets tends to be more positive in nature while for rude and raging sarcasm, the mood of the user tends to be more negative. It matches with the assumption that when exhibiting polite sarcasm, the general mood of the user tends to be positive, and similarly, when exhibiting rude and raging sarcasm, the overall mood of the user tends to be negative as his/her mood tends to be disturbed by anger or rage, which is exhibited by Figure 12.

5. Conclusion

Sarcasm is an implicit form of the sentiment. Detecting sarcasm present in the text helps in understanding the actual emotion with which information was conveyed. In order to detect sarcasm, various features are extracted which include positive and negative intensifiers, skip grams, sentiment value, polarity flip, etc. An ensemble-based feature selection method has then been proposed to identify the optimal feature set among all features and to classify whether a tweet is sarcastic or not, and it achieves an accuracy of 92.7%. Apart from classifying a text as sarcastic or nonsarcastic, a novel attempt has been carried out in this work for categorization of sarcasm into various types. Sarcasm has been classified into four types after analyzing the properties of sarcasm in detailed manner, namely, polite, rude, raging, and deadpan. In order to classify sarcasm into various types, a multi-rule based approach has been proposed. The proposed multi-rule based approach consists of two levels: fuzzy rule-based type detection and rough set-based type detection and validation. Fuzzy rules have been constructed for modelling the various types of sarcasm and the same has been validated and verified with rough set theory-based rules. The proposed multi-rule based approach attained the following results for different sarcastic types: Polite, 95.98%; Rude, 96.2%; Raging, 99.79%; and Deadpan, 86.61%. The results obtained have supported the notion that mood levels of the person influence the type of sarcasm they exhibit. When a person is in not so good mood, rude and raging sarcasm tends to be predominant in the outcome exhibited, and when the mood is positive, they tend to exhibit polite sarcasm predominantly. The validation of the results of the type of sarcasm obtained through fuzzy system was carried out with the rough set approach. Finally, after determining the type of sarcasm, user's mood changes were modelled in the proposed approach. To predict how a user's mood influences the sarcasm and vice versa, the tweets before and after a particular exhibition of a sarcastic type are obtained. This can help in modelling the emotion change of a user by collecting the past tweet histories of each user. The obtained mood change model demonstrates how the type of sarcasm exhibited by the users affects their mood levels.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to convey their thanks to MIT Campus, Anna University, for the support. This research was supported in part by Anna Centenary Research Fellowship.

References

[1] G. Deliens, K. Antoniou, E. Clin, and M. Kissine, "Perspective-taking and frugal strategies: evidence from sarcasm detection," *Journal of Pragmatics*, vol. 119, pp. 33–45, 2017.

[2] A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset, "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network," *IEEE Access*, vol. 7, pp. 23319–23328, 2019.

[3] S. K. Bharti and S. B. Korra, "Sarcasm detection in twitter data: a supervised approach," in *Semantic Web Science and Real-World applications*, M. D. Lytras, N. Aljohani, E. Damiani, and K. T. Chui, Eds., pp. 246–272, IGI Gobal, Hershey, PA, USA, 2019.

[4] K. Ravi and V. Ravi, "A novel automatic satire and irony detection using ensembled feature selection and data mining," *Knowledge-Based Systems*, vol. 120, pp. 15–33, 2017.

[5] M. Bouazizi and T. Ohtsuki, "Opinion mining in Twitter: how to make use of sarcasm to enhance sentiment analysis," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, Vancouver, Canada, August 2015.

[6] M. Khokhlova, V. Patti, and P. Rosso, "Distinguishing between irony and sarcasm in social media texts: linguistic observations," in *Proceedings of the International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT)*, IEEE, Saint-Petersburg, Russia, August 2016.

[7] P. Rockwell, "Vocal features of conversational sarcasm: a comparison of methods," *Journal of Psycholinguistic Research*, vol. 36, no. 5, pp. 361–369, 2007.

[8] J. R. Ragini, P. M. R. Anand, and V. Bhaskar, "Big data analytics for disaster response and recovery through sentiment analysis," *International Journal of Information Management*, vol. 42, pp. 13–24, 2018.

[9] M. El-Masri, N. Altrabsheh, H. Mansour, and A. Ramsay, "A web-based tool for Arabic sentiment analysis," *Procedia Computer Science*, vol. 117, pp. 38–45, 2017.

[10] V. K. Neppalli, C. Caragea, A. Squicciarini, A. Tapia, and S. Stehle, "Sentiment analysis during Hurricane Sandy in emergency response," *International Journal of Disaster Risk Reduction*, vol. 21, pp. 213–222, 2017.

[11] J. Kim, "How Korean EFL learners understand sarcasm in L2 English," *Journal of Pragmatics*, vol. 60, pp. 193–206, 2014.

[12] S. Yoo, J. Song, and O. Jeong, "Social media contents based sentiment analysis and prediction system," *Expert Systems with Applications*, vol. 105, pp. 102–111, 2018.

[13] A. Musolff, "Metaphor, irony and sarcasm in public discourse," *Journal of Pragmatics*, vol. 109, pp. 95–104, 2017.

[14] Y. Ren, D. Ji, and H. Ren, "Context-augmented convolutional neural networks for twitter sarcasm detection," *Neurocomputing*, vol. 308, pp. 1–7, 2018.

[15] J. Lei, Y. Rao, Q. Li, X. Quan, and L. Wenyin, "Towards building a social emotion detection system for online news," *Future Generation Computer Systems*, vol. 37, pp. 438–448, 2014.

[16] M. Ghiassi and S. Lee, "A domain transferable lexicon set for twitter sentiment analysis using a supervised machine learning approach," *Expert Systems with Applications*, vol. 106, pp. 197–216, 2018.

[17] S. Xiong, H. Lv, W. Zhao, and D. Ji, "Towards twitter sentiment classification by multi-level sentiment-enriched word embeddings," *Neurocomputing*, vol. 275, pp. 2459–2466, 2018.

[18] J. Karoui, F. B. Zitoune, and V. Moriceau, "Soukhria: towards an irony detection system for Arabic in social media," *Procedia Computer Science*, vol. 117, pp. 161–168, 2017.

[19] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: a new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[20] S. M. Jiménez-Zafra, M. T. Martín-Valdivia, M. D. Molina-González, and L. A. Ureña-López, "How do we talk about doctors and drugs? Sentiment analysis in forums expressing opinions for medical domain," *Artificial Intelligence in Medicine*, vol. 93, pp. 50–57, 2019.

[21] Y. Kim, K. A. Schug, and S. B. Kim, "An ensemble regularization method for feature selection in mass spectral fingerprints," *Chemometrics and Intelligent Laboratory Systems*, vol. 146, pp. 322–328, 2015.

[22] S. K. Bharti, B. Vachha, R. K. Pradhan, K. S. Babu, and S. K. Jena, "Sarcastic sentiment detection in tweets streamed in real time: a big data approach," *Digital Communications and Networks*, vol. 2, no. 3, pp. 108–121, 2016.

[23] S. Mukherjee and P. K. Bala, "Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering," *Technology in Society*, vol. 48, pp. 19–27, 2017.

[24] D. Voyer and J. P. Vu, "Using sarcasm to compliment: context, intonation, and the perception of statements with a negative literal meaning," *Journal of Psycholinguistic Research*, vol. 45, no. 3, pp. 615–624, 2016.

[25] N. Bhan and M. D'silva, "Sarcasmometer using sentiment analysis and topic modeling," in *Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3)*, IEEE, Mumbai, India, December 2017.

[26] P. Chaudhari and C. Chandankhede, "Literature survey of sarcasm detection," in *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, IEEE, Chennai, India, March 2017.

[27] P. Dharwal, T. Choudhury, R. Mittal, and P. Kumar, "Automatic sarcasm detection using feature selection," in *Proceedings of the 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, IEEE, Tumkur, India, December 2017.

[28] E. Gupta, G. Rathee, P. Kumar, and D. S. Chauhan, "Mood swing analyser: a dynamic sentiment detection approach," *Proceedings of the National Academy of Sciences, India Section A: Physical Sciencesm*, vol. 85, no. 1, pp. 149–157, 2015.

[29] L. Lagerwerf, "Irony and sarcasm in advertisements: effects of relevant inappropriateness," *Journal of Pragmatics*, vol. 39, no. 10, pp. 1702–1721, 2007.

[30] M. T. AL-Sharuee, F. Liu, M. Pratama, and M. Pratama, "Sentiment analysis: an automatic contextual analysis and ensemble clustering approach and comparison," *Data & Knowledge Engineering*, vol. 115, pp. 194–213, 2018.

[31] A. Persicke, J. Tarbox, J. Ranick, and M. St. Clair, "Teaching children with autism to detect and respond to sarcasm," *Research in Autism Spectrum Disorders*, vol. 7, no. 1, pp. 193–198, 2013.

[32] M. Dynel, "Academics vs. American scriptwriters vs. academics: a battle over the etic and emic "sarcasm" and "irony" labels," *Language & Communication*, vol. 55, pp. 69–87, 2017.

[33] E. Ager, "Sarcasm's so simple," in *Proceedings of the International Conference on Web-Age Information Management*, pp. 459–471, Macau, China, June 2014.

[34] E. Gent, "AI learns to spot sarcasm in your tweets," *New Scientist*, vol. 231, no. 3086, p. 22, 2016.

[35] A. Fernández-Caballero, A. Martínez-Rodrigo, J. M. Pastor et al., "Smart environment architecture for emotion detection and regulation," *Journal of Biomedical Informatics*, vol. 64, pp. 55–73, 2016.

[36] A. Porshnev and I. Redkin, "Analysis of Twitter users' mood for prediction of gold and silver prices in the stock market," in *Proceedings of the International Conference on Analysis of Images, Social Networks and Texts*, Springer, Moscow, Russia, July 2014.

[37] S. Zhang, Z. Wei, Y. Wang, and T. Liao, "Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary," *Future Generation Computer Systems*, vol. 81, pp. 395–403, 2018.

[38] Z. Xiaomei, Y. Jing, Z. Jianpei, and H. Hongyu, "Microblog sentiment analysis with weak dependency connections," *Knowledge-Based Systems*, vol. 142, pp. 170–180, 2018.

[39] S. Schuch, J. Zweerings, P. Hirsch, and I. Koch, "Conflict adaptation in positive and negative mood: applying a success-failure manipulation," *Acta Psychologica*, vol. 176, pp. 11–22, 2017.

[40] K. Song, T. Yao, Q. Ling, and T. Mei, "Boosting image sentiment analysis with visual attention," *Neurocomputing*, vol. 312, pp. 218–228, 2018.

[41] H. M. Fardoun, S. R. López, D. M. Alghazzawi, and J. Castillo, "Monitoring students moods for the detection of weaknesses in secondary schools," *Procedia—Social and Behavioral Sciences*, vol. 47, pp. 2049–2054, 2012.

[42] S. Swami, A. Khandelwal, V. Singh, S. S. Akhtar, and M. Shrivastava, "A corpus of English-Hindi code-mixed tweets for sarcasm detection," 2018, https://arxiv.org/abs/1805.11869.

[43] K. Rajeswari and P. ShanthiBala, "Recognization of sarcastic emotions of individuals on social network," *International Journal of Pure and Applied Mathematics*, vol. 18, no. 7, pp. 253–259, 2018.

[44] A. D. Tran, S. Arch-int, and N. Arch-int, "A rough set approach for approximating differential dependencies," *Expert Systems with Applications*, vol. 114, pp. 488–502, 2018.

[45] A. K. Tiwari, S. Shreevastava, T. Som, and K. K. Shukla, "Tolerance-based intuitionistic fuzzy-rough set approach for attribute reduction," *Expert Systems with Applications*, vol. 101, pp. 205–212, 2018.

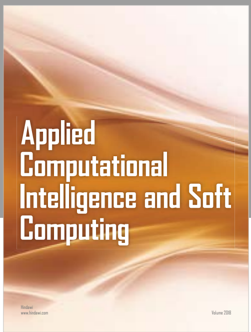
[46] S. Hiai and K. Shimada, "A sarcasm extraction method based on patterns of evaluation expressions," in *Proceeding of the 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, IEEE, Kumamoto, Japan, July 2016.

[47] B. W. Fang and B. Q. Hu, "Granular fuzzy rough sets based on fuzzy implicators and coimplicators," *Fuzzy Sets and Systems*, vol. 359, pp. 112–139, 2019.

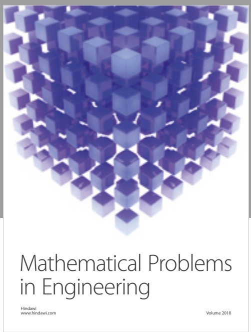
[48] Y. Qian, X. Liang, Q. Wang et al., "Local rough set: a solution to rough data analysis in big data," *International Journal of Approximate Reasoning*, vol. 97, pp. 38–63, 2018.

[49] S. Jayasanka, T. Madhushani, E. Marcus, I. Aberathne, and S. Premaratne, "Sentiment analysis for social media," 2013, https://www.researchgate.net/profile/Saminda_Premaratne/publication/268817500_Sentiment_Analysis_for_Social_Media/links/5478afec0cf293e2da2b2aa0.pdf.

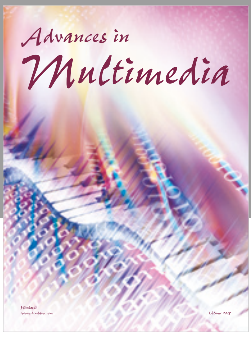
[50] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on twitter: a behavioral modeling approach," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, ACM, Shanghai, China, February 2015.

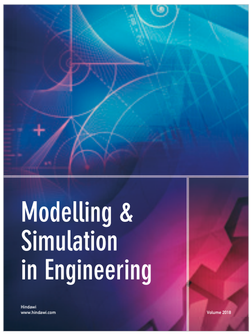

Applied Computational Intelligence and Soft Computing

The Scientific World Journal


Mathematical Problems in Engineering

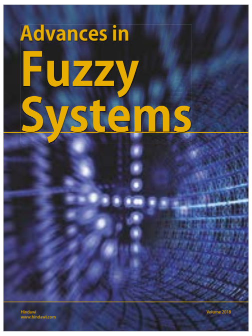
Journal of Engineering

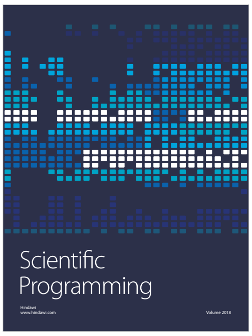

Advances in Multimedia

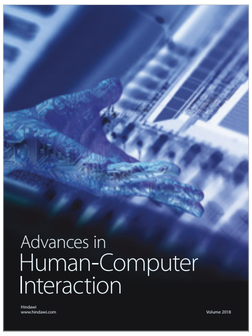

Modelling & Simulation in Engineering

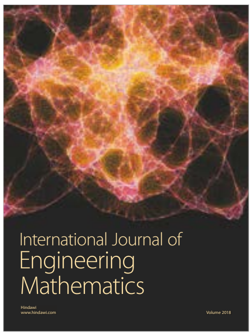
International Journal of Reconfigurable Computing

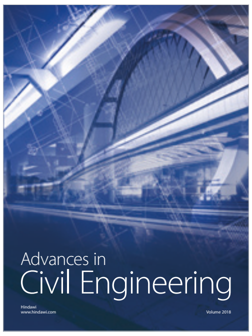
Advances in Artificial Intelligence

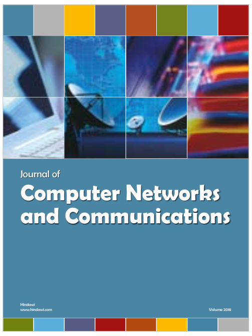

Advances in Fuzzy Systems


Scientific Programming


Advances in Human-Computer Interaction

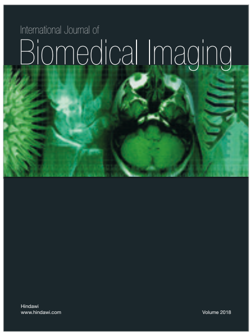

International Journal of Engineering Mathematics


Advances in Civil Engineering

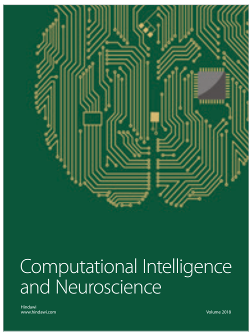

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology


International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience