An Emoticon-Based Novel Sarcasm Pattern Detection Strategy to Identify Sarcasm in Microblogging Social Networks

M. Nirmala, Amir H. Gandomi, *Senior Member, IEEE*, Madda Rajasekhara Babu, L. D. Dhinesh Babu, and Rizwan Patan, *Senior Member, IEEE*

***Abstract*— Online social networks are one of the prime modes of communication used by people to voice their opinions and sentiments, especially after the advancement of digital gadgets and overall technology. Mining such sentiments and analyzing the polarity of user opinions is a trending research issue with high business value. Identifying, detecting, and understanding sarcasm is an important topic in the field of sentiment analysis. Despite being complex and challenging, automated detection of sarcasm is also a relatively less explored research area. In this article, we present a novel sarcasm pattern detection technique using emoticons to identify sarcasm in microblogging social networks like Twitter. Initially, we classify the tweets only with emoticons based on a decision tree classification approach. Afterward, we incorporate the SentiWordNet library and a separate emoticon library to find the polarities of the tokenized words and emoticons. Finally, we present a comparison of the polarity of the tweets and the polarity of the emoticons to detect sarcasm in tweets.**

***Index Terms*— Emoticon in tweets, sarcasm detection, sentiment analysis and tweet analysis.**

I. Introduction

IN THIS technologically advanced and modern world, people rely on the internet to get all the needed information to take care of their personal affairs; product reviews; education; communication; or online transactions. The advent of the internet had a positive impact on the usage of social networks. The success of Facebook, Twitter, LinkedIn, and other online social media are the finest examples that indicate how the public relies on the use of social media. Researchers try to make the best use of online social network information to study future perspectives and to mine the opinions of people toward a particular topic or product. Research on people's sentiments has gained major interest, not only from industry but also from academia. Researchers are successful in deciphering the user information as positive or negative but with certain limitations that are induced by the sarcasm in the information thereby making the task of sentiment analysis extremely complex. Sarcasm is a way of mocking someone using a contrasting meaning [1]. Even for intelligent human beings, the identification of sarcastic context is not as easy as the identification of positive or negative context [2].

Sarcastic context is not as easy as the identification of positive or negative context [2]. For machines, the mining of a sarcastic sentence is a tough and complex task. Because of the high business value in sentiment analysis, the focus of researchers moves toward sarcasm detection in spite of the technical challenges involved in it. In this work, a novel strategy to identify sarcasm in the popular social networking microblogging platform Twitter is proposed. Twitter continues to dominate the microblogging social network space, with a huge user base across the globe. This platform helps users to convey their opinions, thoughts, or any other information on a particular topic in the form of tweets. In this article, tweets that were tweeted by the users are initially categorized into positive or negative tweets based on the sander's algorithm, following the use of the proposed technique, which is used to identify the sarcastic nature of the sentences based on the used emoticons in the tweets. Analyzing the sarcasm in tweets by giving adequate weightage to the emoticons present in the tweets helps in arriving at better inferences. This article is structured as follows. Section I provides a brief introduction to sarcasm in sentimental analysis, Section II discusses the literature of available works on sentiment mining based on sarcasm, Section III elaborates on the novel sarcasm-detecting algorithm, and Section IV details the evaluation of the proposed algorithm. Section V concludes the work with some of the highlights and remarks.

II. Related Works

Automated detection of sarcasm in tweets is relatively less explored in the sentiment analysis research area when compared with other fields of study [3], [4]. Generally,

Manuscript received 21 October 2022; revised 31 July 2023; accepted 4 August 2023. Date of publication 7 September 2023; date of current version 2 August 2024. *(Corresponding authors: Amir H. Gandomi; Rizwan Patan.)*

M. Nirmala and L. D. Dhinesh Babu are with the School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu 632014, India (e-mail: mnirmala@vit.ac.in; lddhineshbabu@vit.ac.in).

Amir H. Gandomi is with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia, and also with the University Research and Innovation Center, Óbuda University, 1034 Budapest, Hungary (e-mail: gandomi@uts.edu.au).

Madda Rajasekhara Babu is with the School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu 632014, India (e-mail: mrajasekharababu@vit.ac.in).

Rizwan Patan is with the College of Computing and Software Engineering, Kennesaw State University, Marietta, GA 30060 USA (e-mail: rpatan@kennesaw.edu).

Digital Object Identifier 10.1109/TCSS.2023.3306908

in sentimental analysis, most of the sarcasm detection techniques involve analyzing the lexicon patterns [4], [5], [6], [7]. Davidov et al. [5] framed a sarcasm detection technique using a semi-supervised learning algorithm on Amazon books and other product reviews. Their algorithm uses a semi-supervised learning technique to mine, select, and match the patterns that tell which of the k-nearest neighbor model can be used to detect the sarcasm [5]. Strapparava and Valitutti [8] used WordNet libraries along with the predefined emotions to parse the text. González-Ibánez et al. [6] used WordNet effect [8] and linguistic count [9] methodologies to detect sarcasm based on supervised learning. They employed their supervised classification techniques of sarcasm using practical and textual classification to find the sarcasm, and finally compared the results with human performance [6]. Maynard and Greenwood [10] studied the impact of sarcasm in tweets, concluding that a lot of features are yet to be explored in the analysis of sarcasm. It is also stated that improved algorithms are required in opinion mining to differentiate the polarity of the tweets. Justo et al. [11] used supervised learning methodologies to detect sarcasm in online texts. Their methodology has three steps: classification, feature extraction, and comparison to find the nasty language, and consequently, find the sarcasm in online texts.

Rajadesingan et al. [4] came up with a behavioral approach to detect sarcasm on Twitter. The article states that the sarcastic tweets are knowingly created by the user to express their opinion in a sarcastic way. SentiStrength [12], a lexicon-based sentiment tool, is used to discover the sentiment weightage of each word in the tweets. A probability distribution on the length of user's past and present tweets is built to calculate the divergences. Based on the sentiment weightage, divergence, and user mood, the tweets are classified as sarcastic or not. Mukherjee and Bala [13] used a naive Bayes classifier together with fuzzy c means clustering to classify the sarcasm from tweets. Ryes et al. [14] developed a classifier to detect humor and irony from the text based on the sentence polarity, the emotional state of the user, and the uncertainty of the sentence. Rilof et al. [15] developed a sarcasm-detecting lexicon technique that uses the dissimilarities between positive sentiments and negative situations.

Davidov et al. [16] used smileys and hashtags on the Twitter dataset to find the sentiment tags. The methodology is based on both k-means neighbor classification and feature extraction, like Davidov et al.'s [5] framework. The previously mentioned works generally use lexical analysis, classification, feature extraction, and selection strategies to detect sarcastic sentences or sarcasm in tweets. Read [17] uses emoticon classifiers to find the sentiments in the tweets, stating that the classifier has 70% of accuracy on tweets with emoticons. Emoticons are not used in any of the above methods in the automated detection of sarcasm. Though Read [17] uses the emoticons classifier, the corpus of emoticons or smiley is lesser when compared with the prevailing scenario, with the additional fact that the experiments of Jonathon Read are in a controlled architecture. Emoticons or smileys are computer-encoded blocks that convey the emotions of the user through a graphical representation of the face [18]. Though there is no clarity on the evolution of emoticons, the way messages are communicated has seen a major transformation with the increased use of emoticons [19].

Research by Provine et al. [20] states that the formal text typing regularity in messages is converted into linguistic conversation with the usage of emoticons. Provine [21], [22] explains that emoticons convey emotions through text. He also argues that without laugher symbol, the textual meaning will be lost in the message. Advancements in technologies made users to convey the state of their emotions through emoticons. Currently, emoticons are used as a rating tool to get the user's opinion on an article or any other product, and even digital surveys are conducted by using emoticons [23]. The prevailing digital society forces every individual to use digital media in one way or the other, which paves way for researchers to analyze the traces of users and to know the users' behavior with emotion analysis and intention detection [23]. Recently, MIT has come out with a proposed technique to analyze tweets and detect sarcasm. The algorithm trains the neural network using deep learning technique, while emotional content is used to recognize sarcasm [24]. Another work that uses emoticons to detect sentiments and sarcasm is presented in [25]. Though several works have been carried out on sarcasm detection to date, there is no solid work on detecting sarcasm through emoticons to the best of our knowledge. In this work, emoticons in tweets are used to detect sarcasm.

III. Emoticons-Based Sarcasm Detection Methodology

Emoticons in Twitter feeds can be used as an effective means for detecting sarcasm as part of sentiment analysis. To implement the proposed algorithm for the detection of sarcasm using emoticons, the tweets are first extracted. Once tweets are extracted, tweets are classified into tweets with emoticons and tweets without emoticons. Since our proposed approach totally depends on emoticons, tweets with smileys are added to the bag of tweets for further proceedings. After segregating the tweets with emoticons, a pattern is framed for detecting sarcasm in tweets.

A. Classification of Tweets

The tweets fetched from Twitter are analyzed to identify the presence of emoticons is clearly explained in Algorithm 1. Initially, tweets are classified based on the decision tree process. This classification is then compared with the predefined emoticon corpus presented in Table I. In the case emoticons are present in the list of tweets of a classified tweet bag, the tweet is appended to the analysis bucket, as depicted in Fig. 1.

B. Tokenizing the Tweets

The separated tweets cannot be used in their current form for the analysis to make the tweets into machine-readable forms that were tokenized. Tokenization is the process of slicing the sequence of alphabetic characters into every single token so the sentimental strength can be computed. The natural language processing package of Python programming language is used on the fetched tweets for tokenization. Let us consider that f

Algorithm 1 Emoticon-Based Sarcasm Detection Algorithm

Step 1: Extract the tweets
Step 2: Check for the emoticons in the tweets
  if the emoticon is present, **then** classify that as a tweet with an emoticon; **else** ignore
Step 3: Add the tweets with emoticons to the main corpus
Step 4: Tokenize each tweet for further comparison with SentiWordNet
Step 5: Compare the Tokenized with SentiWordNet
  Get the polarity of each tweet
Step 6: Check for the Emoticons
  Compare the emoticon with the emoticon corpus
  Assign the polarity based on the emoticon
Step 7: If there is a presence of both positive and negative polarity of emoticons, **then** categorize that tweet as Sarcastic
Step 8: List the Sarcastic Tweets

TABLE I
LIST OF EMOTICON CORPUS

Happy	":-)", ":)", ":]", "=)", ";-)" ,";)", ":-D", ":D", "=D", ":-*", ":*", "O:)", "O:-)", ":-P", ":P", ":-p", ":p", "=P", ":3"
Sad	":-(", ":(", ":[", "=(", ":-O", ":O", ":-o", ":o",":'("

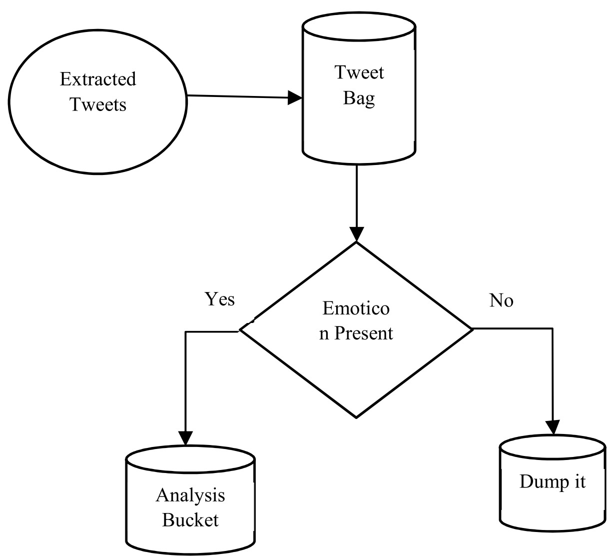


Fig. 1. Classification of tweets.

is the tweet with the total length of l in the analysis bucket ab. All the tweets in the analysis bucket will be tokenized as given in (1), where T denotes tokenization and lower-case t stands for token. In (2), the tokenization of the tweet f_1 is represented by assuming that the length of tweet f_1 is 4

$$\sum_{j=0}^{\text{length of } ab} T(f_j) = \sum_{j=\text{length of } ab} \left(\sum_{i=0}^{l} f_j t(i) \right) \tag{1}$$

$$T(f_1) = \{f_1 t(1), f_1 t(2), f_1 t(3), f_1 t(4)\}. \tag{2}$$

In the process of tokenization, the stop words and punctuations are neglected, so the accurate weightage of each word is achieved. After tokenization, the weightage for each token is assigned based on SentiWordNet. The flow diagram of the proposed approach is depicted in Fig. 2.

C. SentiWordNet Weightage

After tokenization, the weightage is mapped to each word in the analysis bucket. To calculate the sentiment of each word, SentiWordNet [24] is used in our work. SentiWordNet is used widely in the field of opinion mining to differentiate words into positive, negative, and objective. Our proposed work compares each token with the SentiWordNet (SN) library to get the token weightage (t_W).

The SN library consists of positive and negative bags of words. These bags of words are compared with the tokenized word to get the weightage for each token. Considering the positive bag of words as P_L and the negative bag of words as N_L; each token $t(i)$ of the tweet f_j gets compared with both P_L and N_L. If the token $t(i)$ is present in positive bag P_L, it takes the value 1 else it takes the value 0, as depicted in (3). On the contrary, if the token is in negative bag N_L then the value is 1 and 0, as depicted in the following equation:

$$P_{t(i)} = \begin{cases} 1, & \text{if } t(i) \text{ in } P_L \\ 0, & \text{if } t(i) \text{ not in } P_L \end{cases} \tag{3}$$

$$N_{t(i)} = \begin{cases} 1, & \text{if } t(i) \text{ in } P_L \\ 0, & \text{if } t(i) \text{ not in } P_L. \end{cases} \tag{4}$$

Positive and negative values are assigned for the entire tokenized tweet in the analysis basket. Equations (5) and (6) show the representation of positive and negative values, respectively, for each token in the tokenized tweet; where $l(ab)$ denotes the total length of the analysis basket. Finally, (7) and (8) depict the calculation of the total positive and negative weightage of the tweet $W(f_j)$, by computing the summation of positive and negative values of the tweets, respectively

$$f_j(P_{t(i)}) = \{f_j(P_{t(0)}, P_{t(1)}, P_{t(2)}, \ldots, P_{t(l)})\}_{\substack{j=0 \text{ to } l(ab) \\ i=0 \text{ to } l}} \tag{5}$$

$$f_j(N_{t(i)}) = \{f_j(N_{t(0)}, N_{t(1)}, N_{t(2)}, \ldots, N_{t(l)})\}_{\substack{j=0 \text{ to } l(ab) \\ i=0 \text{ to } l}} \tag{6}$$

$$W_P(f_j) = \sum_{i=0}^{l} f_j(P_{t(i)})_{j=0 \text{ to } l(ab)} \tag{7}$$

$$W_N(f_j) = \sum_{i=0}^{l} f_j(N_{t(i)})_{j=0 \text{ to } l(ab)}. \tag{8}$$

D. Emoticon Weightage

Emoticons in the tweets are compared with the emoticon corpus to get the emoticon weight for each tweet. As in Table I, emoticons are classified as happy and sad. Happy emoticons are considered positive ones, while sad emoticons are considered negative ones. For the emoticon-based weightage, the non-tokenized tweets are analyzed using both modified and general regular expressions, as follows:

$$c(eP) = \text{count}(eP) \text{ in } f_j \tag{9}$$

$$c(eN) = \text{count}(eN) \text{ in } f_j \tag{10}$$

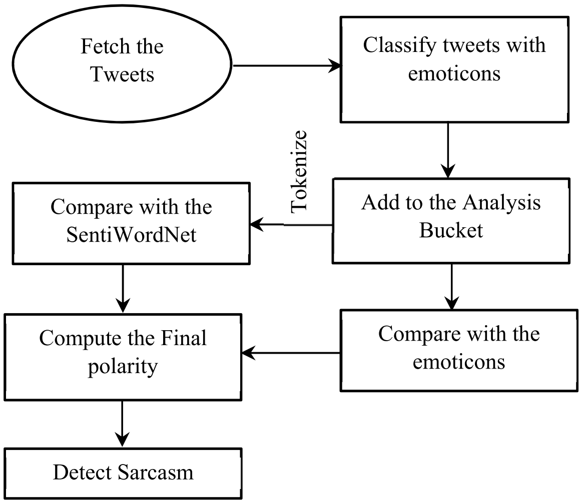


Fig. 2. Flow diagram of the proposed approach.

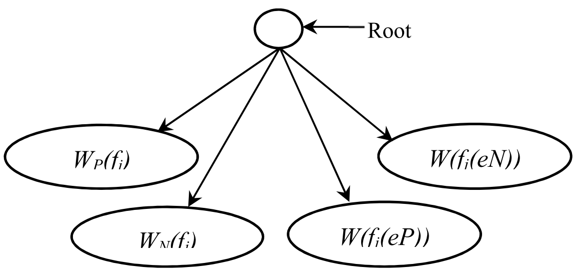


Fig. 3. Tree-based sarcasm detection.

$$W(f_j(eP))_{j=0 \text{ to } l(ab)} = \sum_{k=0}^{c(eP)} V(eP) \tag{11}$$

$$W(f_j(eN))_{j=0 \text{ to } l(ab)} = \sum_{k=0}^{c(eN)} V(eN). \tag{12}$$

In order to calculate the weightage of emoticons present in a tweet, *eP* is considered a positive emoticon, and *eN* is a negative emoticon. For each tweet in the length of the analysis basket, the presence of positive and negative emoticons is counted as in (9) and (10), respectively. Once the counting process is completed, the emoticon weightage of each tweet, on the basis of emoticons, is calculated for positive and negative emoticons, as shown in (11) and (12), respectively. Here, $V(eP)$ denotes the value for the positive emoticon, which is 1, and $V(eN)$ stands for the negative emoticon value, which is also 1. While calculating emoticon weight on tweets, the positive weight $W(f_j(eP))$ of the emoticon in the tweet f_j is calculated by summing up the $V(eP)$ value up to the limit of $c(eP)$, as shown in (11). Similarly, the negative emoticon weightage $W(f_j(eN))$ is calculated as shown in (12). After the computation of emoticon weightages, the process leads to the detection of sarcasm, as depicted in Fig. 3.

E. Sarcasm Detection

Detecting sarcasm is a highly complicated task. In the proposed approach, this detection is computed using the tree kernel method. In sentimental analysis, the efficiency of the tree kernel technique outsmarts other available baseline techniques [25]. The tree is designed to inherit the weightage based on SentiWordNet and weightage based on emoticons.

The root node of the tree represents the tweet from the analysis basket, while the four branches of the root represent the positive weightage of tokens, negative weightage of tokens, positive weight of emoticons, and negative weight of emoticons. In case the root has the entire four branch nodes i.e., if all the values of branch nodes are non-zero, then the tweet is considered as containing sarcasm. The tree is computed to check whether the sarcasm is present or not for all the tweets, in the length of the analysis basket. If any of the weightage values is zero, then the tree would not have four leaf nodes, representing that the tweet is not sarcastic. The efficiency of the proposed algorithm in detecting sarcasm is defined in Section IV.

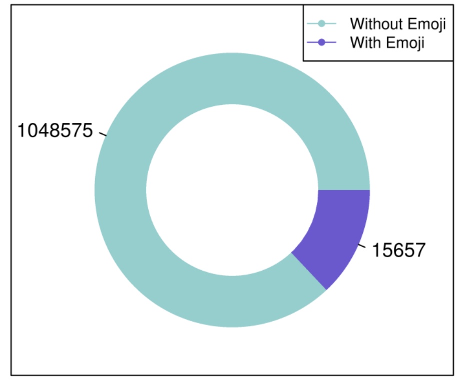


Fig. 4. Total number of emoticons in the total tweets.

IV. Evaluating the Sarcasm Detection Algorithm

The proposed novel sarcasm pattern detection technique is evaluated based on the fetched random tweets. For the purpose of this evaluation, 1 064 232 tweets were randomly fetched from Twitter. Out of this dataset, 15 657 tweets were identified as tweets that have emoticon(s). This is depicted in Fig. 4.

Though the presence of emoticons is much lesser in proportion when compared with the overall fetched tweets, the evolving trends in the digital world clearly indicate that in the future, people will rely more on expressing their emotions using emoticons, and also because this use is much more popular among the younger generation of Twitter users. Among the tweets with emoticons, the SentiWordNet-based analysis is done to find the positive and negative tweets, which is depicted in Fig. 5. From Fig. 5, it can be clearly visualized that in the extracted tweet with emoticons, both positive and negative tweets have an equal half-share each.

The sarcasm in the given tweet is calculated from positive and negative tweets based on the sentiment weightage of emoticons. According to the proposed strategy, the sarcastic sentence should have both positivity and negativity in it. For example, if a positive sentence emotes negativity or when a negative sentence has a positive emotion, then the chance of finding sarcasm is high since there is a contradiction between the two of them.

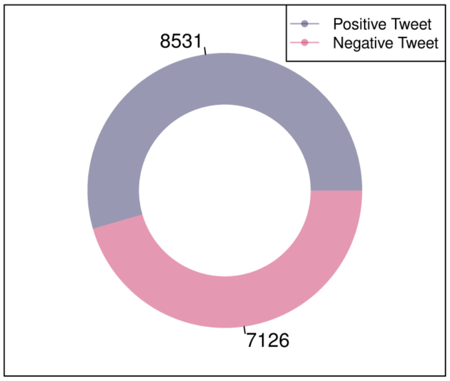


Fig. 5. Polarity of the tweets with emoticons.

TABLE II
SAMPLE OF SARCASTIC TWEETS DETECTED BY THE ALGORITHM

1	Is off to bed, I have work at 9 so I'll be up at 7… Feel for me guys lol:'(Haha .... NIGHT =D
2	I need to do lots of work tomorrow jealous of joely meeting George Michael :O so cool.. Interesting place too ;) carpark... subtle lol
3	Crap! Karate Tomorrow! : [4 more months till' I'm a balack belt! I'm a brown belt right now.:] I could hurt you. Lol. But I wouldn't
4	@abc aww. me too =(I've been 3 times but a fourth time wouldn't hurt best place in the whole world.. ahahaha =D
5	@xyz LOL - that's preferable anyway Be a great shirt to have at W:O:A ;-) hehe
6	@abc seriously?! :O ur so lucky, i hate you. joking. HAVE FUN! oh and buy a souvenir for me ;) wink wink.;
7	@xyz you went to school today??? :O come tmrw. ;) I have band. be ungrounded already!
8	@abc < -- EATS OREOS MADE FROM FECIES!! :o Hehe Cool + I have a headache YES YOU ARE ;)

Based on this pattern, sarcasm is detected in the tweets, as shown in the following equation:

$$\text{Sarcasm} = \text{Positive}_{\text{Weight of the Sentence}} + \text{Positive}_{\text{Emoji}} + \text{Negative}_{\text{Weight of the sentence}} + \text{Negative}_{\text{emoji}}. \quad (13)$$

Out of the total 15 657 tweets, the proposed algorithm detects 1409 tweets as sarcastic tweets. A sample of the detected sarcastic tweets is shown in Table II.

Among the total tweets with emoticons, the proposed approach has found that 9% of the tweets have sarcastic meanings in them. The proposed algorithm for detecting the sarcastic tweet that does not follow any pattern is computed for all the 1409 sarcastic tweets and compared with other sarcasm detection approaches. In Figs. 6 and 7, the proposed emoticon-based sarcasm detection approach (EBSD) accuracy is compared with other approaches, like the pattern-based approach (PBD) [26], contextual model approach (CMD) [27],

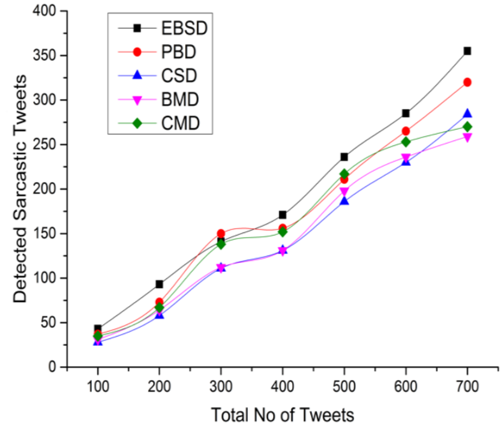


Fig. 6. Detection of sarcastic tweets.

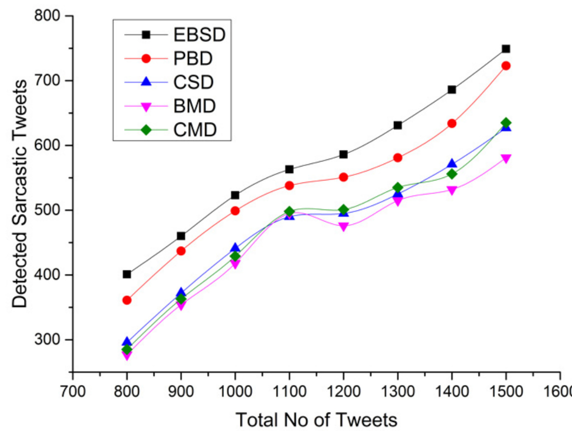


Fig. 7. Detection of sarcastic tweets.

behavioral model (BMD) [28], and contrast sarcasm detection (CSD) approach [28]. The EBSD outperforms all the other approaches in detecting sarcasm from tweets with emoticons. It is observed that the detection rate of the proposed approach is steady and outperforms all the other approaches for a different number of tweets.

In order to analyze the error percentage, the mean absolute error is computed and depicted in Fig. 8. The overall MAE score is lower for the EBSD approach, when compared to other approaches, though there is a slight variation for 200 tweets. This leads to the conclusion that the error percentage is minimum in the proposed approach than the compared approaches. Prediction accuracy of the proposed approach is evaluated by computing Precision, Recall and F1 scores, which can be seen in Figs. 9–11, respectively. It can be concluded from these figures that the prediction accuracy of the EBSD is higher than the other approaches. Though there is a slight variation between the PBD and EBSD approaches, the overall results prove that the proposed approach outperforms all the other approaches.

In order to analyze the efficiency of the proposed EBSD approach in detecting the learned patterns, the algorithm is evaluated on a manually constructed dataset. The dataset consists of three test sets, each having 150 sarcastic tweets of the same pattern as mentioned in (13) and where some examples are shown in Table II. The 150 sarcastic tweets of know the n pattern are classified into five test cases, namely

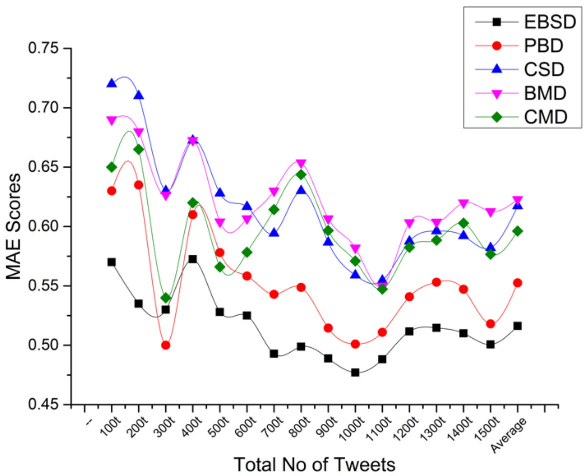


Fig. 8. Comparison of MAE scores of several algorithms.

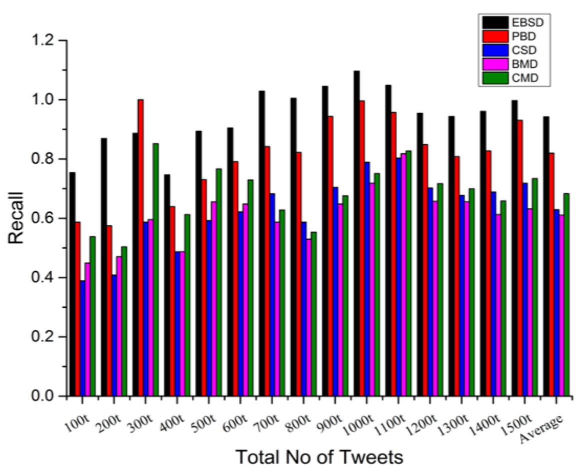


Fig. 9. Comparison of recall values of several algorithms.

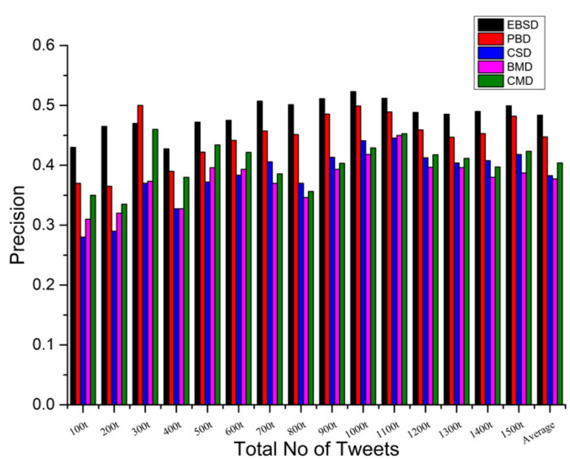


Fig. 10. Comparison of precision values of several algorithms.

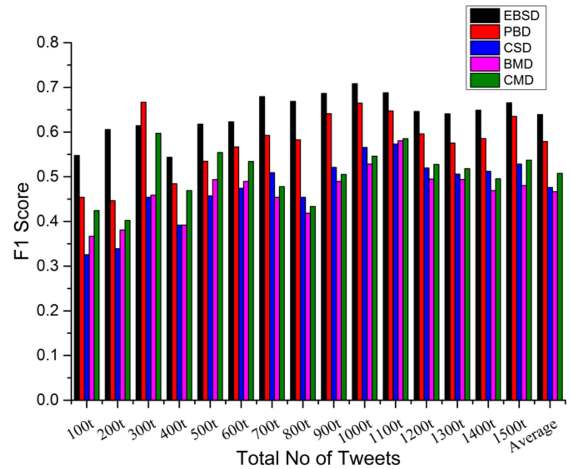


Fig. 11. Comparison of F1 scores of several algorithms.

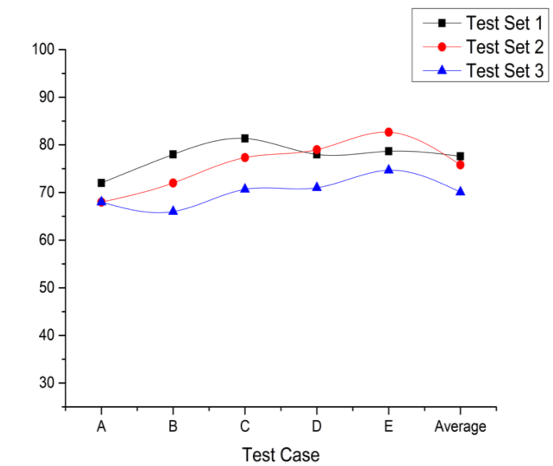


Fig. 12. Comparison of test set accuracy in detecting the known pattern.

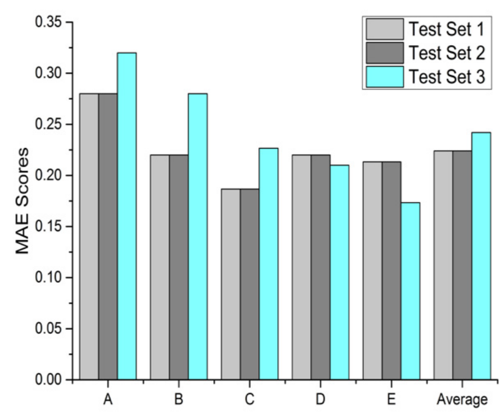


Fig. 13. Comparison of MAE score in detecting the known pattern.

A, B, C, D, and E, which consist of 25, 50, 75, 100, and 150 sarcastic tweets, respectively. The algorithm's accuracy in predicting the known patterns is comparatively higher than identifying the normal sarcastic tweets without any pattern. The result of the algorithm's accuracy in detecting the sarcastic tweet of the known pattern is depicted in Fig. 12. The EBSD approach's accuracy percentage appears to be relatively similar for all the test cases since the average of all the test cases collides with each other. The results of the mean absolute error of the algorithm in detecting the known pattern are depicted in Fig. 13, which shows that the algorithm performs better with the known pattern.

The comparative study of the algorithm's accuracy and error percentage is evaluated on test cases and the results are shown in Figs. 14 and 15, respectively. For comparing the unknown pattern, the test set with random sarcastic tweets is considered. The results show that the accuracy of the algorithm is good for known pattern detection when compared with the unknown

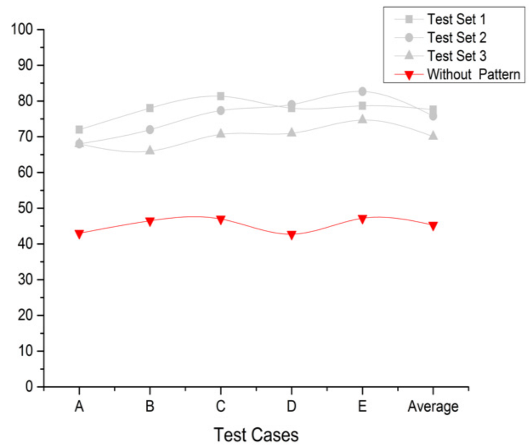


Fig. 14. Comparison of accuracy in detecting the known and unknown patterns.

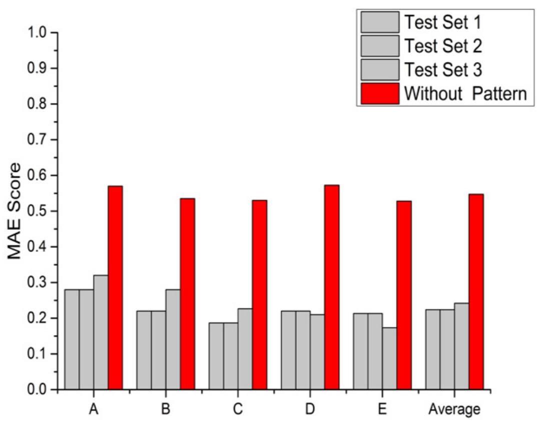


Fig. 15. Comparison of MAE Score in detecting the known and unknown patterns.

pattern sarcastic detection that is shown in Fig. 14. The error percentage score for the known pattern is much smaller than the MAE score for the unknown pattern, depicted in Fig. 15.

This shows that the efficiency of the algorithm is high for the known sarcasm pattern defined in this work, though the accuracy and MAE score for the unknown pattern is much smaller than that of the known pattern. The efficiency of the proposed EBSD approach is relatively better when compared with the other approaches. The detection accuracy can also be increased when the algorithm learns the pattern. The analysis of the overall tweets with emoticons shows that the novel sarcastic pattern detection algorithm can accurately find the sarcastic sentence among the tweets. This work creates new dimensions in both pattern-based sarcasm detection and also in the use of emoticons in detecting sarcasm. In the near future, emoticons are likely to be used in effectively exhibiting emotions by the upcoming trendy tech-savvy digital generation of modern times when compared to the usage of words. Therefore, this work paves the way for further novel and in-depth detection strategies in the near future.

V. Conclusion

Detecting sarcasm from text is one of the complex tasks in the field of sentimental analysis, due to the involvement of uncertainties in it. Our work uses emoticons, which can exhibit the emotions of users, to effectively detect sarcasm in tweets. This is the first and novel work to fully use emoticons in detecting sarcasm in tweets, to the best of our knowledge. The proposed approach paves the way for future research on sarcasm based on emoticons in microblogs and tweets since it has shown improvements in performance when compared with similar methods in the literature. The proposed approach works effectively when there are emoticons in the tweets. As a future extension, the proposed work can be further extended to recognize sarcasm in tweets, even in case of deviation from the pattern mentioned in this algorithm. Considering the complexity of detecting sarcasm, the proposed algorithm proves effective for sarcasm detection in tweets.

References

[1] R. Gibbs, "Irony in talk among friends," *Metaphor Symbol*, vol. 15, no. 1, pp. 5–27, Apr. 2000.

[2] G. A. Bryant and J. E. Fox Tree, "Recognizing verbal irony in spontaneous speech," *Metaphor Symbol*, vol. 17, no. 2, pp. 99–119, Apr. 2002.

[3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, p. 135, 2008.

[4] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on Twitter: A behavioral modeling approach," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, Feb. 2015, pp. 97–106.

[5] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in Twitter and Amazon," in *Proc. 14th Conf. Comput. Natural Lang. Learn.*, 2010, pp. 107–116.

[6] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in Twitter: A closer look," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol.*, 2011, pp. 581–586.

[7] R. J. Kreuz and G. M. Caucci, "Lexical influences on the perception of sarcasm," in *Proc. Workshop Comput. Approaches Figurative Lang.*, 2007, pp. 1–4.

[8] C. Strapparava and A. Valitutti, "WordNet affect: An affective extension of WordNet," in *Proc. LREC*, 2004, pp. 1083–1086.

[9] J. W. P. Francis, M. E. Francis, and R. J. Booth, *Linguistic Inquiry and Word Count*, vol. 71. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Apr. 2001, pp. 1–28.

[10] D. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis," in *Proc. LREC*, 2014, pp. 4238–4243.

[11] R. Justo, T. Corcoran, S. M. Lukin, M. Walker, and M. I. Torres, "Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web," *Knowl.-Based Syst.*, vol. 69, pp. 124–133, Oct. 2014.

[12] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 61, no. 12, pp. 2544–2558, Dec. 2010.

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

[14] A. Reyes, P. Rosso, and D. Buscaldi, "From humor recognition to irony detection: The figurative language of social media," *Data Knowl. Eng.*, vol. 74, pp. 1–12, Apr. 2012.

[15] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 704–714.

[16] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using Twitter hashtags and smileys," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 241–249.

[17] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proc. ACL Student Res. Workshop*, 2005, pp. 43–48.

[18] Wiki. *Emoticons (Unicode Block)*. Wikipedia, Accessed: Apr. 18, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Emoticons_(Unicode_block)

[19] L. K. Kaye, H. J. Wall, and S. A. Malone, "'Turn that frown upside-down': A contextual account of emoticon usage on different virtual platforms," *Comput. Hum. Behav.*, vol. 60, pp. 463–467, Jul. 2016.

[20] R. R. Provine, R. J. Spencer, and D. L. Mandell, "Emotional expression online: Emoticons punctuate website text messages," *J. Lang. Social Psychol.*, vol. 26, no. 3, pp. 299–307, Sep. 2007.

[21] R. R. Provine, "Contagious laughter: Laughter is a sufficient stimulus for laughs and smiles," *Bull. Psychonomic Soc.*, vol. 30, no. 1, pp. 1–4, Jul. 1992.

[22] R. R. Provine, "Laughter punctuates speech: Linguistic, social and gender contexts of laughter," *Ethology*, vol. 95, no. 4, pp. 291–298, Apr. 2010.

[23] L. K. Kaye, S. A. Malone, and H. J. Wall, "Emojis: Insights, affordances, and possibilities for psychological science," *Trends Cognit. Sci.*, vol. 21, no. 2, pp. 66–68, Feb. 2017.

[24] MIT. (Aug. 2017). *Deepmoji*. [Online]. Available: http://google.com/newsstand/s/CBIw8tD-7jk

[25] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations," 2017, *arXiv:1708.00524*.

[26] A. Esuli and F. Sebastiani, "SENTIWORDNET: A high-coverage lexical resource for opinion mining," *Evaluation*, vol. 17, no. 1, p. 26, 2007.

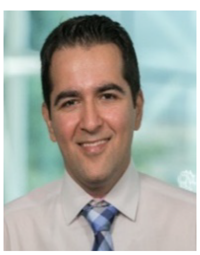
[27] S. A. El Rahman, F. A. AlOtaibi, and W. A. AlShehri, "Sentiment analysis of Twitter data," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, Apr. 2019, pp. 1–4.

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

[29] D. Bamman and N. A. Smith, "Contextualized sarcasm detection on Twitter," in *Proc. ICWSM*, 2015, pp. 574–577.

M. Nirmala received the bachelor's degree in electronic science from the University of Madras, Chennai, India, in 2000, the master's degree in computer applications from Madurai Kamaraj University, Chennai, in 2003, and the Master of Technology degree in computer science and engineering and the Ph.D. degree from the Vellore Institute of Technology, Vellore, Tamil Nadu, India, in 2010 and 2019, respectively.

She is currently an Associate Professor with the School of Information Technology and Engineering, Vellore Institute of Technology. Her research interests include data analytics, social computing, sentiment analysis, databases, data mining, and data warehousing.



Amir H. Gandomi (Senior Member, IEEE) is currently a Professor of data science and an ARC DECRA Fellow with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. He is also affiliated with Óbuda University, Budapest, Hungary, as a Distinguished Professor. Prior to joining UTS, he was an Assistant Professor at the Stevens Institute of Technology, Hoboken, NJ, USA, and a Distinguished Research Fellow at BEACON Center, Michigan State University, East Lansing, MI, USA. He has published over 300 journal articles and 12 books which collectively have been cited more than 43 000 times (H-index = 93). He has been named as one of the most influential scientific minds and received the Highly Cited Researcher Award (top 1% publications and 0.1% researchers) from Web of Science for six consecutive years, from 2017 to 2022. In the recent most impactful researcher list, done by Stanford University and released by Elsevier, he is ranked among the top 1 000 researchers (top 0.01%) and top 50 researchers in AI and Image Processing Subfield in 2021. He also ranked 17th in GP bibliography among more than 15 000 researchers. He is active in delivering keynotes and invited talks. His research interests include global optimization and (big) data analytics using machine learning and evolutionary computations in particular.

Prof. Gandomi has received multiple prestigious awards for his research excellence and impact, such as the 2023 Achenbach Medal and the 2022 Walter L. Huber Prize, the highest-level mid-career research award in all areas of civil engineering. He has served as an Associate Editor, an Editor, and a Guest Editor in several prestigious journals, such as an Associate Editor for *IEEE Network* and IEEE INTERNET OF THINGS JOURNAL.

Madda Rajasekhara Babu received the B.Tech. degree in electronics and communication engineering from S.V. University, Tirupathi, Andhra Pradesh, India, in 1998, the M.Tech. degree in computer science and engineering from R.E.C (presently known as NIT), Calicut, Kerala, in 2001, and the Ph.D. degree from VIT University, Vellore, Tamil Nadu, India, in 2011.

He is currently a Professor with the School of CSE, VIT University. He has published more than 100 papers in journals and conferences, produced more than ten Ph.D. scholars and filed three patents. He has authored edited books published by *Computer Architecture* (Allied), *Compiler Design* (Macmillan), and *Grid Computing books* (Springer), and. He established state-of-art Multicore Architecture, Embedded Systems, and Internet of Things (IoT) Laboratories with the financial grant from Intel, India. He served in various prestigious positions such as a Division Leader (TCS and LT) and a Program Manager. His research interests include the Internet of Things (IoT), high performance computing (HPC), and data analytics.

L. D. Dhinesh Babu received the Bachelor of Engineering degree in electrical and electronics engineering and the Master of Engineering degree in computer science from the University of Madras, Chennai, India, and the Ph.D. degree from the Vellore of Institute of Technology (VIT), Vellore, Tamil Nadu, India.

He is currently a Professor and an Associate Dean with the School of Information Technology and Engineering, VIT. He has 25 years of Academic, Research, and Administrative Experience. He has served in various administrative capacities at VIT. He has published extensively in journals with good impact factor. His research interests include cloud computing, online social network analysis, recommender systems, big data analytics, machine learning, and software engineering.

Dr. Babu is a reviewer for several journals of repute.

Rizwan Patan (Senior Member, IEEE) received the Ph.D. degree from the School of Computing Science and Engineering from the Vellore Institute of Technology, Vellore, India, in 2017.

He is currently working as an Assistant Professor with the Department of Software Engineering and the Game Development with Kennesaw State University, Marietta, GA, USA. He has more than 100 research contributions to his credit, which are published in refereed and indexed journals, conferences, and book chapters. He has authored and edited more than five books published by IGI Global, and Springer in cybersecurity and big data area. His research interests include blockchain technology, cybersecurity, big data, and the Internet of Things (IoT).

Dr. Patan has received several recognitions and best papers awards at top international conferences. His paper was also nominated in 2022 for the prestigious Best Research Paper Award from the IEEE INTERNET OF THINGS JOURNAL. He received the Best Reviewer Award from *Computer Communication* (Elsevier), IEEE INTERNET OF THINGS JOURNAL. In addition, he secured Excellent Researcher Awards from VIT University in 2016, World Research Council (WRC) and United Medical Council in 2019, and VRS Engineering College in 2021. He played a key role in a number of international symposia and workshops at various IEEE and Springer Conferences. He currently serves as an Editor and the Lead Guest/Associate Editor for several well-reputed journals, including *Information Medical Unlock* (Elsevier) and *Neural Computing and Applications* (Springer).