**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 23, No. 2, August 2021, pp. 993~1001
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v23.i2.pp993-1001


# Sarcasm detection of tweets without #sarcasm: data science approach

**Rupali Amit Bagate[1], R. Suguna[2]**
[1,2]Department of Computer Science & Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Chennai, India
[1]Department of Information Technology, Army Institute of Technology, Dighi Hills, Pune, India

**Article Info**

***Article history:***

Received Feb 5, 2021
Revised May 24, 2021
Accepted Jun 1, 2021

***Keywords:***

Dataset
Deep learning
Hashtag
LSTM
Machine learning
Neural network
Sarcasm detection

**ABSTRACT**

Identifying sarcasm present in the text could be a challenging work. In sarcasm, a negative word can flip the polarity of a positive sentence. Sentences can be classified as sarcastic or non-sarcastic. It is easier to identify sarcasm using facial expression or tonal weight rather detecting from plain text. Thus, sarcasm detection using natural language processing is major challenge without giving away any specific context or clue such as #sarcasm present in a tweet. Therefore, research tries to solve this classification problem using various optimized models. Proposed model, analyzes whether a given tweet, is sarcastic or not without the presnece of hashtag sarcasm or any kind of specific context present in text. To achieve better results, we used different machine learning classification methodology along with deep learning embedding techniques. Our optimized model uses a stacking technique which combines the result of logistic regression and long short-term memory (LSTM) recurrent neural net feed to light gradient boosting technique which generates better result as compare to existing machine learning and neural network algorithm. The key difference of our research work is sarcasm detection done without #sarcasm which has not been much explored earlierby any researcher. The metrics used for evolution is F1-score and confusion matrix.



***Corresponding Author:***

Rupali Amit Bagate
Department of Computer Science & Engineering
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Chennai
No. 42, Avadi-Vel Tech Road, Vel Nagar Chennai, Tamil Nadu 600062, India
Email: rupali.bagate@gmail.com

## 1. INTRODUCTION

Beginning of the internet gave a new vision to the world by changing the way people around the world interact. Now, people started expressing their feeling in front of other people to whom they even don't know. Also, people gather the opinion of each-others feeling, for a particular thing. It may be noted that for humans, it is easy to understand the opinion of other people. However, for a machine, it is very difficult to understand what people are says and how they feel. Sentiment analysis helps machine to analyze the written sentence and classifies it as a positive, negative, or neutral. Sentiment analysis gathers and recognizes attitudes and opinions depicted by users in social media toward a definitive topic. Research on sentiment analysis made machines capable of detecting whether a sentence is positive, negative, or neutral with a good accuracy depending upon the dataset. However it is very difficult to findthe exact sentiment, when the present sentence is layered with sarcasm, thus making it extremely difficult to find out whether the sentence is said in sarcastic manner or not [1].

Embarking with the original well-known research by [2] which handles sarcasm detection in speech, the field has witnessed ample significance in the sentiment analysis community. Sarcasm, which is both assuredlyabsurd and negatively awful, handles a vital part of human's social intercommunication. Sarcasm can also be used for purposes like blame, censure, or comedic relief. For example, 'nice perfume. How long did you marinate in it?'. It is very difficult to identify whether this sentence is written in a sarcastically manner. Numerous researches are going on sarcasm detection. Twitter dataset can be viewed as an important tool for sentiment analysis or sarcasm detection [3]. Twitter exhibits a plentiful terrain for the depiction of enormous views and assessments. In a previous works, #sarcasm was used to detect the sarcasm [4], [5]. This work identify the sentence is said sarcastically without using #sarcasm. Here, a stacking technique is used to analyze the result in a better way. In stacking technique, we trained our Twitter dataset through a logistic regression and evaluated the output. Similarly, we trained the same twitter dataset again on neural network along with LSTM layer to find the output [6]. This output is used as an input to train dataon extreme gradient boosting (XGBoost) forfinal prediction. The proposed methodology achieves better results without #sarcasm.

This research paper is organized into following sections. Section 2 presents survey of research work carried out on related topic. Section 3 describes the dataset which is used as input to proposed model. Section 4 briefs about pre-processing techniques, which is an integral part of machine learning technique. Section 5 explains methodologies involved in the sarcasm detection of tweets by removing #sarcasm word. Section 6 explains the experimental results carried on a pre-processed tweet dataset after combining various machine learning and deep learning techniques.

## 2. RELATED WORK

This section describes research carried out on sarcasm detection work. Buschmeier *et al*. [7] articulated the problem of sarcasm detection as a supervised classification task and evaluated classifiers using logistic regression. They considered the difference between the star-rating and the overall polarity of words in the reviews given by the customer. Joshi [8] captured context incongruity using morphological acceptable similarity/differences between word embedding. The authors used the similarity score between the words in a sentence using word2vec. Poria *et al*. [9] developed a model based on a pre-trained convolutional neural network for extracting sentiment. This grasps sarcasm as a feature automatically using a convolutional neural network form sarcastic corpus. Sentiment shifting was observed in the tweets as it can indicate the presence of sarcasm. Shifting in the framework proposed, they train their model to extract sentiment-specific features. Ghosh and Veale [3] suggested a sarcasm detection using neural network semantic model using convolution neural network followed by an LSTM network and at end-side a deep neural network (DNN). Their model outruns state-of-the-art text-based methods for sarcasm detection, prove out an F-score of .92. They also executed the model against two publicly available datasets [10], [11] and seen that their model has met with a better f-score than previous systems but achieved a reduced precision value than semi-supervised sarcasm identification algorithm (SASI). Bagate and Suguna [12] have surveyed different approaches to sarcasm detection which articulated and compared various techniques for sarcasm detection. Thus, we tried to incorporate some models in our research. Bharathi *et al*. [13] have used the Twitter dataset for sentiment analysis for better stock prediction. As tweets are realistic and updated on a day-to-day basis. As observed in related work, researchers have done work by considering #sarcasm as a part of processing tweet dataset while feeding it to different machine learning models. Whereas our research contribution focuses on the prediction of tweets without #sarcasm which makes machines more intelligent to predicts tweets sarcastic even if a hashtag is absent. Next section describes about the dataset used for research work. We referred a short text for model training. Therefore, we choose twitter as dataset for better analysis and prediction.

## 3. DATASET

Propsed research uses a dataset of Twitter, collected from Kaggle [10], [11] which has two columns. Column 1 has labels of the tweet and column 2 contains tweets stated by different users on Twitter. In column 1, if the label indicates 0, means tweet is not sarcastic and if the labelindicates 1 tweet is sarcastic. The shape for our dataset is (51188, 2), as described the tweets which are marked as sarcastic is labeled as 1 and non-sarcastic tweets are marked as 0 in dataset. There are a total number of 24452 sarcastic and 26736 nonsarcastic tweets. Thus, can be considered as a balanced dataset. A balanced dataset is one that contains an approximatly equal number of samples for each class. It has numerous benefits in training as it reduces bias. Reason behind choosing tweeter as dataset for solving research problem to have standard comparison of results with latest trends. Next session is describing different preprocessing steps to feed noise free data for model building.

## 4. PRE-PROCESSING

In natural language processing (NLP) data preprocessing is the most essential part of any machine learning model. Good results are depending upon how well the data has been preprocessed [14], [15]. In proposed model #sarcasm from the tweet is removed and then data is pre processed. Mainly in four steps tokenization, stop words removal, noise removaland normalization. Below description shows systematic data preprocessing on sentence step by step:

### 4.1. Tokenization

In a nutshell, tokenization is about breaking strings of text into “tokens” means smaller pieces. Paragraphs gets tokenized into sentences and sentences gets tokenized into words. Tokenizer () function used in python for generation of tokens. After such tokenization, process will return a list of tokens. ['I', 'respect', 'my', 'parents', 'no', 'matter', 'what', 'we', 'fight', 'what', 'we', 'say', 'because', ‘'I', 'know', 'at', 'the', 'end', 'of', 'the', 'story', 'they', 'will', 'always', 'stood', 'by', 'me', ”#sarcasm’].

### 4.2. Noise removal

Remove punctuation symbols like a comma, exclamation point, quotation mark, and question mark. These word which does not have any significance in sentences. After noise removalwe get a list of clean tokens which can be processed further. We also converted all charcters or words into lowercase. The python translate () function used to remove punctuations and the lower () function used to lower all chacters from dataset. Outcome of noise removal is being as. ['i', 'respect', 'my', 'parents', 'no', 'matter', 'what', 'we', 'fight,' what', 'we',' say', 'because', ‘'I', 'know', 'at', 'the',' end','of','the','story','they','will','always','stood','by','me.',’#sarcasm’].

### 4.3. Stop words removal

A stop word is a frequently used word (such as "the", "a", "an", "in", "i"), we have ignored those words as they do not affect the polarity of the tweet and is of no use for us.We have written a python script for stop word removal. And tokens will be returned as: ['respect', 'parents.', 'matter', 'fight,', 'say', 'know,', 'end,', "story", 'always', 'stood.',’#sarcasm’].

### 4.4. Normalization

Normalization is the final step of preprocessing. This aims to put all text on a level playing field, e.g., transforming all characters to lowercase. The normalization process enhances text matching. E.g., there are so many ways that the term "bread-butter" can be represented as, like bread & butter, bread and butter, bread-butter, and bread/butter. Normalization provides words with a common and simpler form for further text processing. Then after gathering these tokens we will get a string: "respect parents matter fight say know end story they will always stood me #sarcasm". By doing so we have left with only significant data that will be used in further process. We used SnowballStemmer() function for stemming as a part of normalization. In proposed work, we removed hashtags present in sentences, this will help our model to train intelligently to maintain novelty of work. The only sentence we left with after preprocessing is “respect parents matter fight say know end story always stood me #sarcasm". Removal of #sarcasm.

### 4.5. Removal of #sarcasm

From the dataset will automate the prediction on an entire tweet more intelligently, instead of just checking they mentioned manually #sarcasm in tweet. This will help system to correctly identify tweets even if #sarcasm is not mentioned explicitly by tweet user while tweeting on social media. The sentence will be now "i respect my parents no matter what we fight what we say because i know at the end of the story they will always stood by me”. Figure 1 shows code snippet of #sarcasm removal.

```
gb =df
gb = gb["tweet"]
modified =[]
for i in gb.values :
    url = re.sub('\#sarcasm$', '', i)
    modified.append(url)
df["tweet"] = modified
```

Figure 1. Removal of #sarcasm

## 5. METHODOLOGY

Data pre-processing is an essential part of any machine learning model. Good results always depend upon how well the data has been pre-processed. In NLP text pre-processing is first step before building a model. Tokenization, lower casing, stop words removal and stemming come under pre-processing. Tokenization is the process of breaking the sentences into words and stemming is used to reduce the inflexions in the words. Stop words is used to remove the unwanted words, these are those word which does not add any significance to the sentences. By doing so we have left with only significant data that will be used in future.

Further ensemble models used to predict more accurate valuethan individual classifiers [16], [17]. The proposed architecture showed in Figure 2 uses a stacking technique, which ensemble machine learning models. It uses a meta-learning technique to combine best predictions from two or more base machine learning algorithms. Stacking uses weak models in parallel to predict and get better accuracy. Figure 2 describes architectural flow of proposed research. Logistic regression, neural network (using LSTM layer) and XGBoost model used for sarcasm classification.

Proposed technique splits the cleaned data into training and testing dataset. Based on training data logistic regression and neural network models are trained. Both model predictions are treated as metadata and XGBoost model is trained further. Large balanced dataset (24452 sarcastic, 26736 non sarcastic) is split into train and tested with 20% testing data which lately treated as metadata. Further this metadata is again split into train test data with 20% as testing data. The processed dataset is then fed for predicting tweets, in which user have not left any clue about whether tweet is sarcastic or not. We used two methodologies for sarcasm detection explained below logistic regression.

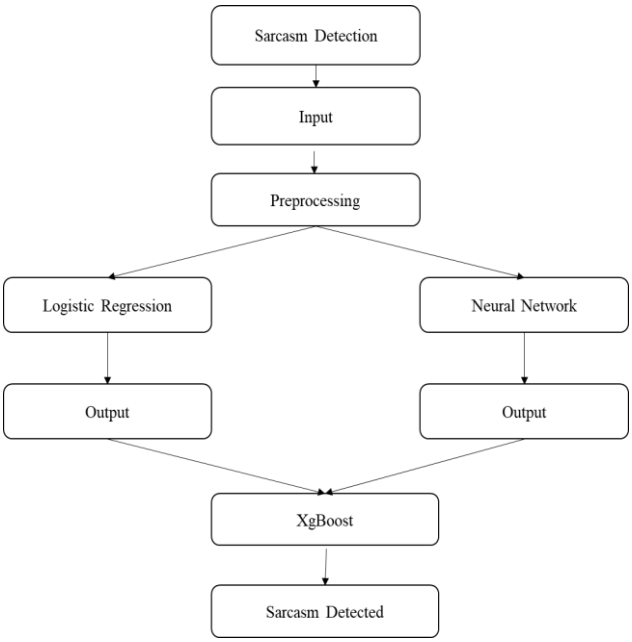


Figure 2. Flow diagram of sarcasm detection

### 5.1. Logistic regression

Logistic Regresssion is the simplest classification algorithm. Earlier many authors have used logistic regression for sarcasm detection on Twitter dataset [18]. They concluded that only #sarcasm is not the indicator for sarcasm which leads us to process our dataset and remove #sarcasm. Logistic regression is a supervised classification algorithm. In the classification problem, the target variable (y) can only take discrete values for a given set of input variables (x). The given problem is based on binary classification and its target variable (y) can take only two discrete values 0 or 1. Logistic regression algorithm uses the sigmoid function to classify into discrete values. Figure 3 shows a graphical view of the sigmoid function which shows how y varies with change in x.

$$g(z) = \frac{1}{1+e-z} \quad (1)$$

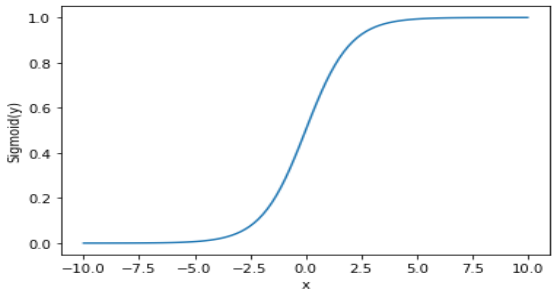


Figure 3. Sigmoid Function used by LR

### 5.2. Neural network

The neural network is gaining popularity and is the backbone of machine learning. It tends to mimic humans. Neural networks learn from the patterns found in the dataset. Figure 4 depicts the simple structure of neural networks. LSTM Layer is added in our proposed neural network model. LSTM layer retains the previous value into its memory therefore decision is basedon historical comments as well. It is very useful in the prediction of continuous data. Figure 5 shows a simple LSTM structure used for research. Various other authors used LSTM to solve different problems that depend on continuous data. Few authors have used the same technique in sarcasm detection. Since sarcasm depends on the entire sentence and may vary with a single change of word [19]. So, it needs to retain a previous text to give accurate results.

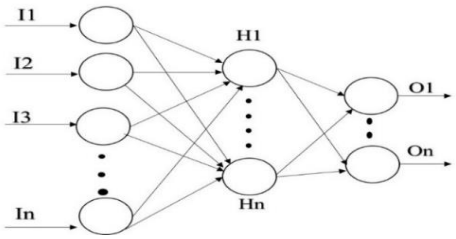


Figure 4. Structure of neural network

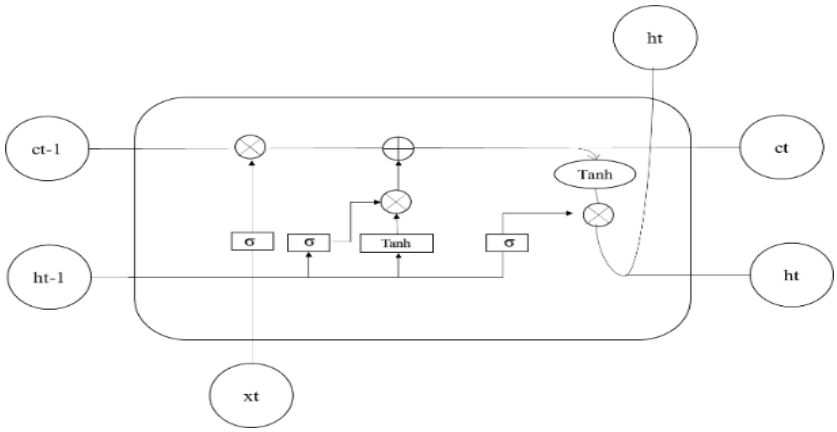


Figure 5. LSTM structure

### 5.3. XGboost

XGboost is a boosting technique and widely used nowadays. XGboost is an implementation of gradient boosting machines called "Extreme Gradient Boosting". As compared to other gradient boosting algorithms XGboost is very rapid and accurate [20]. It is one of the ensemble techniques which uses second partial derivatives of the loss function, which gives more information about the direction of gradients and how to achieve a minimum of our loss function. This technique is used in irony detection for the Twitter dataset and give good results [21]. As shown in Figure 6 shows the XGboost working style.

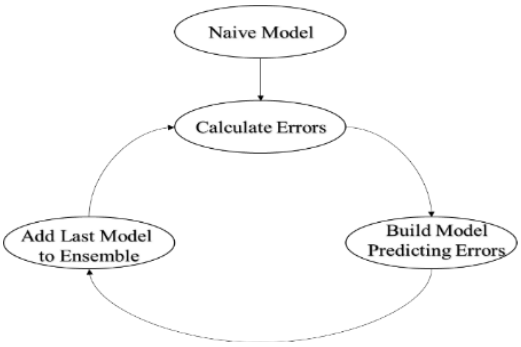


Figure 6. XGBoost working

## 6. EXPERIMENTAL RESULTS

Our research is using twitter dataset that is being created and used by researchers [10], [11] as an input dataset. Tweets being used by our model need to be pre-processed before feeding to our model for a good probable outcome. Confusion matrix measures the effectiveness of any machine learning model classification. Measuring the outcome of research, different measuring parameters are used such as precision, recall, accuracy, and f1-score. These all-mentioned measuring parameters are getting inputs from a confusion matrix. When a given problem statement can be classified into two or more classes, confusion matrix can be used as a measuring parameter. A confusion matrix is formed as a combination of two rows and columns total four elements named true positive (TP), false positive (FP), false negative (FN), and true negative (TN). Figure 7 shows a pictorial representation of the confusion matrix. As shown in below figure we have some terminologies such as TP, FP, FN, and TN are described, which are very important parameters while calculating the efficiency of classifiers. True positives (TP) mean if predicted outcome comes as a positive and actual outcome is also positive. True negative (TN) means if predicted outcome comes as a negative and actual outcome is opposite of that. False positive (FP), the prediction is positive and actual outcomes generates as false. Lastly false negative (FN), the prediction is negative and actual values also comes as false. We have taken F1-score as efficiency parameter check the performance of the classifier. F1-score is the harmonic mean of recall and precision [22]. Therefore, to calculate F1-score we should know precision and recall. As shown are the equations for recall and precision. In (2) shows recall and (3) shows precision and (4) shows F1-score.

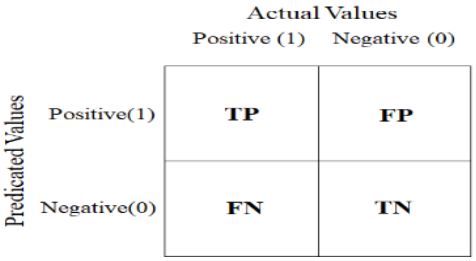


Figure 7. Confusion matrix

Ideally, recall and precision equal to 1 indicates accurate results. As FN and FP increases recall and precision decreases which is not good in terms of prediction. Whenever recall and precision comes one F1-score measures better prediction of a classifier. We extensively tried various combination of different models and we achieved maximum accuracy with the combination of logistic regression, neural network, and XGboost achieving F1 score value of 72.44 %.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$F1 - score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

### 6.1. Logistic regression

Since logistic regression is used with dichotomous dependent variables this is the first technique to be used in such type of problems [23]. We achieved an accuracy of F1 score value of 32.47%. Table 1 shows a confusion matrix for logistic regression.

Table 1. Confusion matrix LR

| | | Predicted | |
|---|---|---|---|
| | | Yes | No |
| Actual | Yes | 3930 | 1378 |
| | No | 1971 | 2959 |

### 6.2. Neural network

We have extensively experimented with the number of nodes, layers, epochs, optimizers, metrics, and loss functions. We have used advanced neural network technique with LSTM Layer to extract the features for training using 100 neurons and achieved an F1 score of 62.77%. Table 2 shows a confusion matrix for neural network with LSTM and Table 3 shows detailed model architecture of the same. We have extensively experimented with the 100 number of nodes, 4 layers, 50 epochs, sgd optimizers, embedding metrics, and loss functions. We have used advanced neural network technique with LSTM Layer using 100 neurons and achieved an F1 score of 62.77%.

Table 2. Confusion matrix NN

| | | Predicted | |
|---|---|---|---|
| | | Yes | No |
| Actual | Yes | 4469 | 639 |
| | No | 2627 | 2303 |

Table 3. Model architecture of NN

| Layer (type) | Output Shape | Param # |
|---|---|---|
| Embedding_1 (Embedding) | (None, 50,50) | 1000000 |
| Dropout (Dropout) | (None, 50,50) | 0 |
| Conv 1d_1(Conv 1D) | (None,46,64) | 16064 |
| Max_pooling 1d (Max Pooling 1D) | (None, 11,64) | 0 |
| Lstm_1 (LSTM) | (None,100) | 66000 |
| Dense_1(Dense) | (None,1) | 101 |
| Total Params: 1,082,165 | | |
| Trainable Params: 1,082,165 | | |
| Non-Trainable Params: 0 | | |

### 6.3. Stacking

Finally, we used the XGboost model which takes the input from both the models and we achieved an accuracy of F1 score value of 73.04%. Table 4 shows a confusion matrix of stacking which passed through XGBoost. As earlier described in subsection. XGboost is an ensemble technique with minimum loss achieves good F1 score.

Table 4. Confusion matrix of stacking

| | | Predicted | |
|---|---|---|---|
| | | Yes | No |
| Actual | Yes | 731 | 348 |
| | No | 221 | 748 |

Confusion matrix shown in Table 4 calculates F1-score of final classification of research. We calculate precision, recall, and f1-score from the above matrix by referring to (2)-(4). In Table 5 we have compared proposed work with previously existing work. Our proposed work is giving good F1-score 73.10%.

$$Recall = \frac{700}{895}\ Precision = \frac{700}{1079}\ F1 - score = 2 * \frac{0.78212 * 0.6487}{0.78212 + 0.6487}$$

Table 5. Comparitive analysis of existing work

| Authors | Dataset | Methodology | Language | Accuracy |
|---|---|---|---|---|
| Liebrecht, C. C *et al*. [24] | Tweets | Balanced Winnow | Dutch | 75% |
| González-Ibánez, *et al*. [25] | Tweets | Sequential minimal optimization (SMO) with LIWC+_P (Linguistic Inquiry and Word Count) and LIWC+ _F | English | 75.89% |

## 7. CONCLUSION

Sarcasm is a complicated term to analyze and understand. Sarcasm is rich in semantics. Many authors achieved a good F1-score for sarcasm detection with #sarcasm present in tweets. We observed and surveyed very few researchers have worked on without #sarcasm tag while identifying sarcasm from tweets. Identifying sarcasm without #sarcasm hashtag is difficult for a model as compared to with #sarcasm, as the system does not have a reference of hashtag while training. Therefore, the system tries to identify the presence of sarcasm from the present context. We have implemented a better stacking technique with a combination of weak base models and achieved an accuracy of 73.10%. For future work, we can use different features combination along with sentiment to identify sarcasm. So, we can improve the different measuring parameters such as precision, recall, and F1-score can benefit system to categorize sarcasm in a better way.

## ACKNOWLEDGEMENTS

Authors thanks to Department of Computer Science & Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Chennai, India, and Army Institute of Technology for providing an infrastructure to carry research on above mentioned topic.

## REFERENCES

[1] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining text data*, *Springer*, 2012, pp. 415-463, doi: 10.1007/978-1-4614-3223-4_13.
[2] J. Tepperman, D. R Traum, and S. Narayanan, "Yeah right: Sarcasm recognition for spoken dialogue systems," *Ninth international conference on spoken language processing*, 2006.
[3] A. Ghosh and T. Veale, "Fracking sarcasm using neural network," *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, 2016, pp. 161-169, doi: 10.13140/RG.2.2.16560.15363.
[4] S. Nadali, M. A. A. Murad, and N. M. Sharef, "Sarcastic tweets detection based on sentiment hashtags analysis," *Advanced Science Letters*, vol. 24, no. 2, pp. 1362-1365, 2018, doi: 10.1166/asl.2018.10750.
[5] A. G. Prasad, S. Sanjana, S. M. Bhat, and B. S. Harish, "Sentiment analysis for sarcasm detection on streaming short text data," *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, 2017, pp. 1-5, doi: 10.1109/ICKEA.2017.8169892.
[6] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 2016, pp. 1-5, doi: 10.1109/IISA.2016.7785373.
[7] K. Buschmeier, P. Cimiano, and R. K. Semantic, "An Impact Analysis of Features in a Classification Approach to Irony Detection in Product Reviews," *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2014, pp. 42-49.
[8] A. Joshi, P. Bhattacharyya, M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Computing*, vol. 50, no. 5, Nov. 2017, doi: 10.1145/3124420.
[9] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks," *arXiv preprint arXiv:1610.08815*, Oct. 2016.
[10] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as a contrast between a positive sentiment and negative situation," *Proceedings of the 2013 conference on empirical methods in natural language processing*, Oct. 2013, pp. 704-714.
[11] A. Ghosh and T. Veale, "Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal," *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*, Sep. 2017, pp. 482-491.
[12] R. A. Bagate and R. Suguna, "Different Approaches in Sarcasm Detection: A Survey," *International Conference on Intelligent Data Communication Technologies and Internet of Things*, Sep. 2019, pp. 425-433
[13] S. Bharathi, A. Geetha, and R. Sathiynarayanan, "Sentiment analysis of Twitter and RSS news feeds and its impact on stock market prediction," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 6, pp. 68-77, Jul. 2017, doi: 10.22266/ijies2017.1231.08.

[14] C. Sindhu, G. Vadivu, and V. Mandala, "A comprehensive study on sarcasm detection techniques in sentiment analysis," *Int J Pure Appl Math*, vol. 118, pp. 433-442, 2018.
[15] M. S. Razali, A. A. Halin, L. Ye, S. Doraisamy, and N. M. Norowi, "Sarcasm Detection Using Deep Learning with Contextual Features," *IEEE Access*, vol. 9, pp. 68609-68618, 2021, doi: 10.1109/ACCESS.2021.3076789.
[16] B. Peng, J. Wang, and X. Zhang, "YNU-HPCC at SemEval-2018 task 3: Ensemble neural network models for irony detection on Twitter," *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval'18). ACL*, Jun. 2018, pp. 622-627, doi: 10.18653/v1/S18-1101.
[17] E. Fersini, F. A. Pozzi, and E. Messina, "Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers," *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2015, pp. 1-8, doi: 10.1109/DSAA.2015.7344888.
[18] D. Bamman and N. A. Smith, "Contextualized sarcasm detection on Twitter," *Proceedings of the 9th International Conference on Web and social media AAAI*, Apr. 2015.
[19] A. Kumar and G. Garg G, "Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets," *Journal of Ambient Intelligence and Humanized Computing*, Aug. 2019, pp. 1-16, doi: 10.1007/s12652-019-01419-7.
[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785-794, doi: 10.1145/2939672.2939785.
[21] M. Khalifa and N. Hussein, "Ensemble Learning for Irony Detection in Arabic Tweets," *Proceedings of the IDAT@FIRE2019 Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings*, 2019, pp. 433-438.
[22] K. Rajeswari and P. Shanthibala, "Sarcasm detection using machine learning techniques," *Int J Recent Sci Res.*, vol. 9, pp. 26368-26372, 2018.
[23] A. Christmann, "Least median of weighted squares in logistic regression with large strata," *Biometrika*, vol. 81, no. 2, pp. 413-417, Jun. 1994, doi: 10.1093/biomet/81.2.413.
[24] C. C. Liebrecht, F. A. Kunneman, and A. P. J. Van Den Bosch, "The perfect solution for detecting sarcasm in tweets# not," *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2013.
[25] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in Twitter: a closer look," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 581-586.

## BIOGRAPHIES OF AUTHORS

**Rupali Amit Bagate,** is research scholarat department of Computer Science & Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai. The Author working as Assistant Professor at theDepartment of Information Technology, Army Institute of Technology, Dighi Hills, Pune. Area of intrest is Natural Language precessing and big data.

**R. Suguna**, Professor at department of Computer Science & Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai.