*2023 4th International Conference for Emerging Technology (INCET)*
*Belgaum, India. May 26-28, 2023*

# Detecting Sarcasm in Reddit Comments: A Comparative Analysis

Babita Sonare
*Department of Information Technology*
*Pimpri Chinchwad College of Engineering*
Pune, India
sonare.babita@gmail.com

Jaya H. Dewan
*Department of Information Technology*
*Pimpri Chinchwad College of Engineering*
Pune, India
jaya.h.dewan@gmail.com

Sudeep D. Thepade
*Department of Computer Engineering*
*Pimpri Chinchwad College of Engineering*
Pune, India
sudeepthepade@gmail.com

Vedang Dadape
*Department of Information Technology*
*Pimpri Chinchwad College of Engineering*
Pune, India
vedang12d@gmail.com

Tejas Gadge
*Department of Information Technology*
*Pimpri Chinchwad College of Engineering*
Pune, India
tejasgadge2882@gmail.com

Aditya Gavali
*Department of Information Technology*
*Pimpri Chinchwad College of Engineering*
Pune, India
adityagavali18@gmail.com

2023 4th International Conference for Emerging Technology (INCET) | 979-8-3503-3575-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/INCET57972.2023.10170613

***Abstract*—Sarcasm is the use of sarcastic words to mock or mockingly show disdain for something. Several people frequently use it on social media sites like Reddit and Twitter. This study investigates the effectiveness of deep learning and machine learning algorithms in detecting sarcasm using SARC dataset consisting of 1.3 million Reddit comments with almost equal amounts of sarcastic and neutral comments. We compare several well-known machine learning classification methods, including Logistic Regression, Naïve Bayes, Decision Tree Classifier, and Convolutional Neural Networks (CNN). Our results, with an accuracy of 73.2%, demonstrate that the model designed using a fusion of CNN and Long Short-Term Memory Networks (LSTM) techniques performed better than alternative classification algorithms. Our findings show how machine learning techniques will be used in the future to identify sarcasm on social networking websites.**

***Keywords— Sarcasm, Reddit, Natural Language Processing, Classification, Machine Learning, Deep Learning***

## I. INTRODUCTION

The language used to express irony is known as sarcasm, and it frequently involves using words that mean the opposite of what is intended. Sarcasm can be used to convey comedy, rage, or to make a point. Sarcasm is typically communicated in face-to-face interactions by tone, gestures, and facial expressions, but in written communication, it is frequently challenging to discern sarcasm without extra contextual information. Finding sarcastic expressions in text and figuring out what they mean is known as sarcasm detection.

Users can publish material about their areas of interest on Reddit [1], an online discussion forum. Subreddits are a category for the interest areas. Many different subreddits exist, including r/AskReddit, r/worldnews, and r/politics. The data is collected from the users of the subreddit r/sarcasm, who publish comments from other subreddits that have been identified as sarcastic by appending a '/s' to the text.

Sentiment analysis is the process of examining user comments to ascertain their emotional undertone, such as whether they are positive, negative, or neutral [2]. However, sarcasm conveys a different message than the text itself. Thus, sarcasm detection can improve and streamline the current sentiment analysis method.

Sarcasm detection has been tackled by machine learning (ML) algorithms in a variety of methods. To determine whether the text is sarcastic or not, supervised learning algorithms like Support Vector Machines, Decision Trees, Naïve Bayes, and Logistic Regression, as well as deep learning techniques like Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM), are used. These algorithms generate their predictions using variables, including word frequencies, sentiment ratings, and lexical clues. They are trained using a dataset of labeled sarcastic and neutral text examples.

Another method is to find patterns and connections in the data that are suggestive of sarcasm using unsupervised learning algorithms, like clustering and dimensionality reduction. Latent variables in the data that can be used as features for sarcasm detection are uncovered using these techniques.

In this study, sarcasm detection using sentiment and syntactic analysis is researched. Performance comparisons between various machine learning algorithms and deep learning approaches on the SARC dataset [3] are demonstrated. The objective is to determine the best model to detect sarcasm in online communities. The models' overall accuracy, precision, recall, and F1 score are analyzed.

## II. LITERATURE SURVEY

A self-annotated corpus designed for studying sarcasm in natural language was introduced in [3]. The corpus comprises diverse text samples from multiple sources, and the authors have annotated each sample for sarcasm. The corpus provides a significant contribution to the study of sarcasm in social media. Through analyzing the corpus, the authors examine the distribution, frequency, and linguistic markers of sarcasm, which can provide valuable insights for developing computational models for detecting sarcasm in natural language. In [4] and [5], supervised machine learning algorithms have been used to detect sarcasm in Reddit and Twitter data, respectively. The authors have proposed methods for detecting sarcasm through the utilization of sentiment and syntactic features to increase accuracy. The potential applications of this approach include social media analysis, customer feedback analysis, and online opinion mining.

In a recent study, automatic sarcasm detection is analyzed [6]. The review covers various techniques utilized in sarcasm detection, such as feature-based, corpus-based, and deep-learning approaches. The review also addresses the limitations of current approaches and proposes directions for

future research, such as improving datasets and exploring cross-lingual sarcasm detection. [7] examines various approaches to detecting sarcasm in online conversations and highlights limitations, including the difficulty in detecting sarcasm in short messages and context-dependent sarcasm. They also provide recommendations for future research, such as more accurate labeling of sarcasm and developing more sophisticated algorithms. An approach for detecting sarcasm on Twitter, combining linguistic and contextual features, is proposed in [8]. The method involves identifying sarcastic patterns in the text and analyzing the contextual features to determine the presence of sarcasm. The proposed approach can also be applied to other social media platforms, making it a promising solution for detecting sarcasm in online conversations.

Researchers investigated deep learning and contextual features' effectiveness in detecting sarcasm on a publicly available dataset [9]. Training and testing on various algorithms, including CNN, SVM, and logistic regression, are performed. The results demonstrate that the logistic regression algorithm outperforms the other algorithms in terms of accuracy. The researchers also noted the significance of contextual features in detecting sarcasm. [10] proposed a method that involves utilizing dense neural networks, which are designed to contain multiple hidden layers, with pre-trained word embeddings to extract features and classify sarcastic comments accurately. The outcomes demonstrate that the approach performs better than the earlier approach in accuracy, precision, recall, and F1 score levels. The study concludes that the proposed approach can potentially detect sarcasm in various contexts.

Comparison in the performance of supervised machine learning models using a comprehensive set of linguistic and contextual features, including sentiment analysis, word frequency, and part-of-speech tagging, was done in [11]. The findings demonstrated that the logistic regression model performed better compared to the other models. Researchers reviewed the current machine learning models trained to detect sarcasm on Twitter. They reviewed 18 studies and found that most were feature-based approaches to detect sarcasm [12] and concluded that a combination of CNN and SVM algorithms achieved the best results with accuracy of 97.71 %.

In [13], authors proposed a novel method for sarcasm detection that uses Bidirectional Encoder Representations from Transformers (BERT) for encoding the input text into vectors and then using these to predict the next word in sequence and Graph Convolutional Networks (GCN) to propagate information through the graph and make predictions about nodes in the graph. An accuracy of 88.3% was achieved using this approach to detect sarcasm in News headlines. An approach for sarcasm detection using a stacked bidirectional LSTM model is proposed in [14]. Text is represented as vectors using pre-trained word embeddings and then fed into the model to learn long-range dependencies between words in the text. An accuracy of 82.4 % was on tweets data making it significantly more accurate than other baseline methods.

## III. Research Methodology

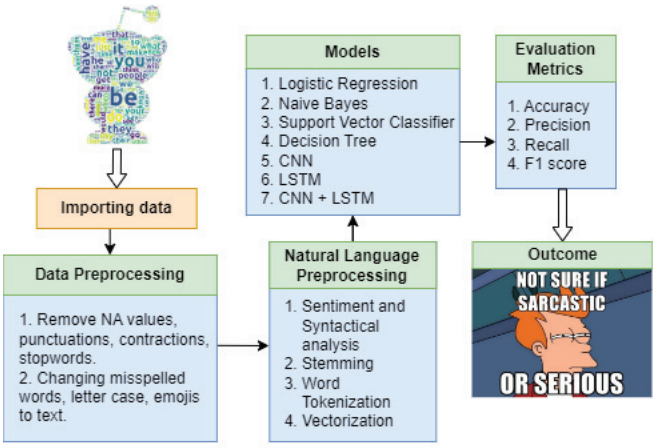


Fig. 1. Proposed Methodology

Fig. 1 demonstrates the steps used in the proposed system.

### A. Data Preprocessing

Data is preprocessed to remove irrelevant information and convert the text into a numerical representation that can be input into the ML model. Data preprocessing involves:

- *Removing NA Values:* NA values are missing/ unavailable, and since there is no use for missing comments, these NA values must be removed.
- *Change to Lowercase:* Comment must be converted to lowercase to reduce index size so that two exact words - one in uppercase and the other in lowercase should not be indexed twice.
- *Changing Misspelled words*: Users use acronyms instead of words (ex: good night → gn). These words must be mapped correctly to preserve their correct meaning.
- *Converting Emojis to Text*: Emojis are converted to textual format to extract meaning using NLP.
- *Removing Contractions*: Removing words formed by shortening and combining multiple words (ex: don't → do not).
- *Removing Punctuations*: Punctuations like '.', '!', and '?' do not contribute to the meaning of the comment and need to be removed since they are in abundant amounts.
- *Removing Stop words*: Commonly occurring words like 'and', 'the', and 'I' do not contribute to the sentence's meaning and must be removed as they make the data noisy.
- *Train Test Split*: The dataset is divided into a training set and a test set in the proportion of 70:30. The training set is used to train the ML model, while the test set is used for evaluating that model.

### B. Natural Language Preprocessing

Text data is transformed into a format that can be processed by ML algorithms using natural language processing (NLP).

- *Syntactic Analysis*: Statistical models are used to analyze the context of each word in a sentence and predict the probability that a given word belongs to a particular part of speech (ex: noun, verb, adjective, adverb).
- *Sentence Sentiment score*: Reflects the emotional tone of a sentence (positive, negative, or neutral).
- *Stemming*: Stemming removes affixes from words to extract their root form. Similar words can be indexed together instead of storing all forms of the word. Ex: (swimming, swimmer, swim → swim)
- *Word Tokenization*: It splits user comments into an array of words. Tokenization is needed as further analysis needs to be done on each word.
- *Feature Extraction*: Since many ML algorithms require performing several mathematical operations on data, tokenized data must be transformed into a numerical format. Employed techniques are:

(a) *Count Vectorization*: Represents each word as a vector of its frequency across the entire corpus of documents.

(b) *TF-IDF Vectorization*: Calculates a weighted score for each word based on its frequency in a document and its rarity in the entire corpus.

(c) *Word2Vec Embeddings*: Uses a neural network to learn vector representations of words by predicting the context in which words appear in a text corpus.

### C. Model Training

The following models are analyzed in this study:

1 *Logistic Regression* [15]: A classification-focused supervised ML algorithm used to model the connection between a dependent binary variable and single or multiple independent variables. Its output represents the probability of the dependent variable belonging to a particular class.

2 *Decision Tree* [15]: Decisions are made by recursively splitting the data into smaller subsets based on the features of the data. The algorithm creates a tree-like model, where each internal node represents a feature, and each leaf node represents a decision. The algorithm chooses the feature that best splits the data into homogeneous subsets based on a Gini Impurity index or Information gain.

3 *Support Vector Machines (SVM)* [15]: The algorithm divides data points into two or more classes by constructing an ideal hyperplane (boundary) that maximally separates the classes based on the support vectors closest to the hyperplane.

4 *Naïve Bayes* [15]: Based on the Bayes theorem, this probabilistic ML method presumes that the features used for classification are unrelated. It determines the likelihood that a given sample belongs to a specific class and then chooses the class with the greatest likelihood.

5 *Convolutional Neural Networks (CNN)* [16]: A deep learning model with many layers of convolutional filters separate from the input data's features and layers of pooling that reduce the dimensionality. After being flattened, the convolutional layers' output is sent through the connected layers for classification. They are used for identifying key features and learning hierarchical representations of text data. Model architecture is displayed in Fig. 2.

6 *Long Short-Term Memory Networks (LSTM)* [16]: A recurrent neural network (RNN) intended to avoid the vanishing gradient issue affecting conventional RNNs. LSTM is suited for speech recognition and NLP tasks because it can manage long-term dependencies in input sequences. Based on input gates and output gates, LSTMs have an internal cell state that can selectively retain or forget information. They also have an ignore gate that enables them to remove redundant data from the cell state. In model architecture, the Conv1D layer of CNN architecture is replaced with the LSTM layer.

7 *Combining CNN and LSTM techniques* [16]: The LSTM layer stores temporal dependencies, whereas CNNs extract significant features from the text. Concatenated output is provided to the connected layers for classification. Model architecture after combining CNN and LSTM layers is shown in Fig. 3.

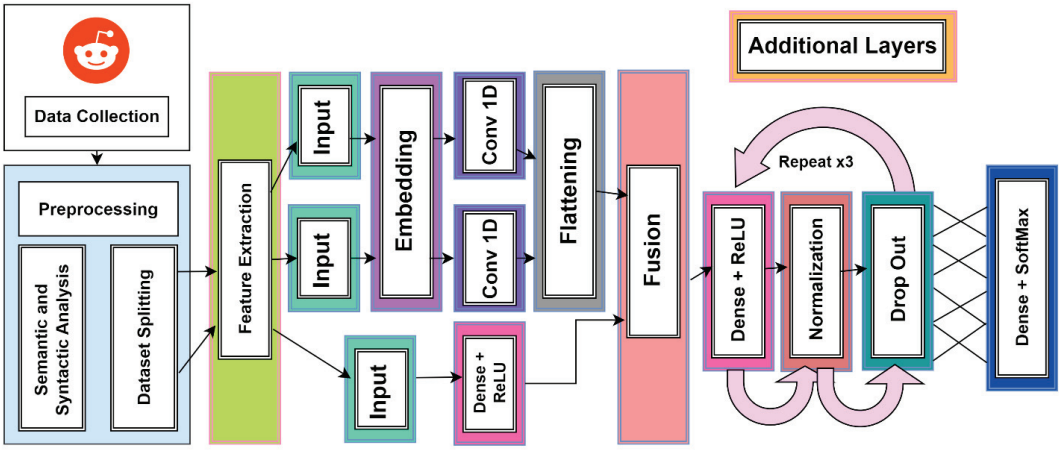


Fig. 2. CNN 1D Model Architecture.

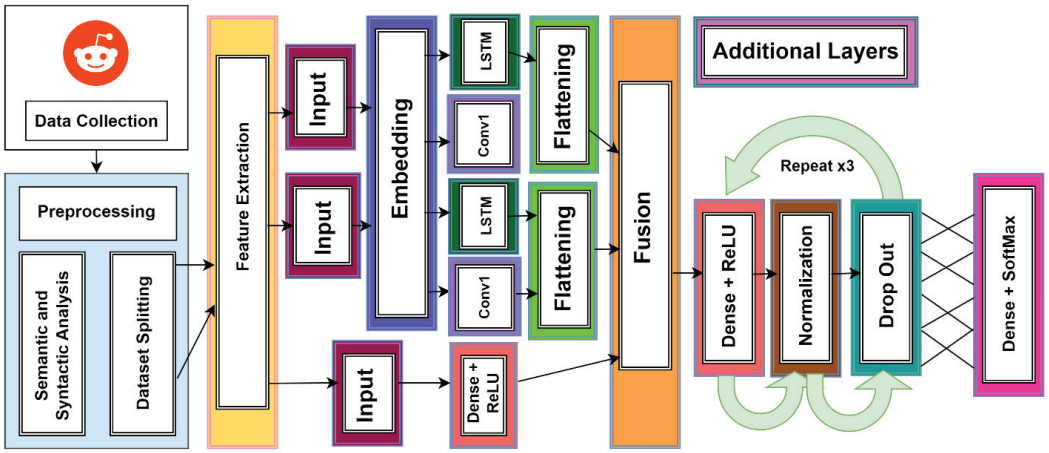


Fig. 3. CNN 1D + LSTM Model Architecture

## IV. Experimental Setup

The models have been trained on Jupyter Notebook, which allows the use of popular Python libraries like SkLearn and Tensorflow. Matplotlib and Seaborn libraries are used to plot graphs. ML models have been trained using AMD Ryzen 9 4900HS processor with eight cores and 16 threads. The device was equipped with 16 GB of RAM and a 1 TB M.2 NVMe solid-state drive. Nvidia RTX 2060 MaxQ GPU has been used to accelerate the training process. The training process was run on a Windows 10 operating system.

### A. Dataset Overview

The SARC dataset contains sarcastic comments and is compiled by Mikhail Khodak et al. for their article [3]. The comments were collected by web scraping from various subreddits on the online community Reddit.

The dataset contains about one million data entries and ten attributes. The attributes or columns are:

1 label: To classify whether the comment is sarcastic or not.

2 comment: Textual/Emoji response to original Reddit post.

3 author: Reddit username of the individual who posted the comment.

4 subreddit: Topic-specific forum where the comment was posted.

5 score: Difference between the number of upvotes and the number of downvotes the comment has received.

6 ups: The number of upvotes the comment has received.

7 downs: The number of downvotes the comment has received.

8 date: Month and year the comment was posted in YYYY-MM format.

9 created_utc: Date and time the comment was posted in YYYY-MM-DD HH:MM:SS format.

10 parent_comment: If the comment is part of a Reddit discussion, it refers to the thread's original root comment.

TABLE I. Classification of Comments in Dataset

| Classification | Number of Comments in Dataset |
|---|---|
| Sarcastic | 505368 |
| Neutral | 505405 |

### B. Visualizations

The word frequency of the data is displayed in the word cloud [17]. The word clouds in Fig. 4. and Fig. 5. show the most frequently used words in sarcastic and neutral comments, respectively.

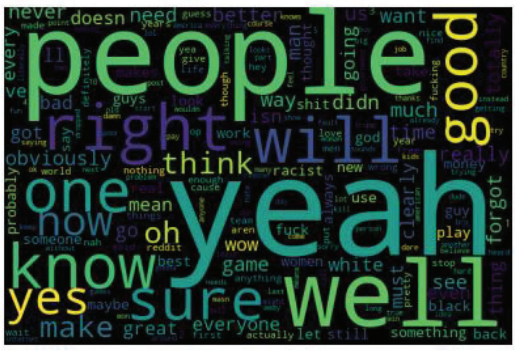


Fig. 4. Sarcastic word cloud.


Fig. 5. Neutral word cloud.

It is observed that ‘yeah’ and ‘one’ are the most used words in sarcastic and neutral comments, respectively, and ‘people’ is prevalent in both sarcastic and neutral comments.

The bar graph in Fig. 6. displays ten subreddits with the highest percentage of sarcastic comments.

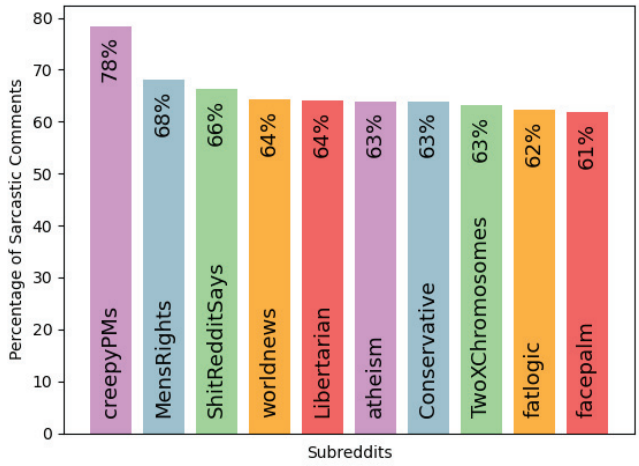


Fig. 6. Top ten Sarcastic Subreddits.

Evidently, ‘r/creepyPMs’ is the most sarcastic subreddit.

As the data is in skewed format, the grouped bar graph in Fig. 7. is used to depict natural log lengths of comments instead of comment lengths.

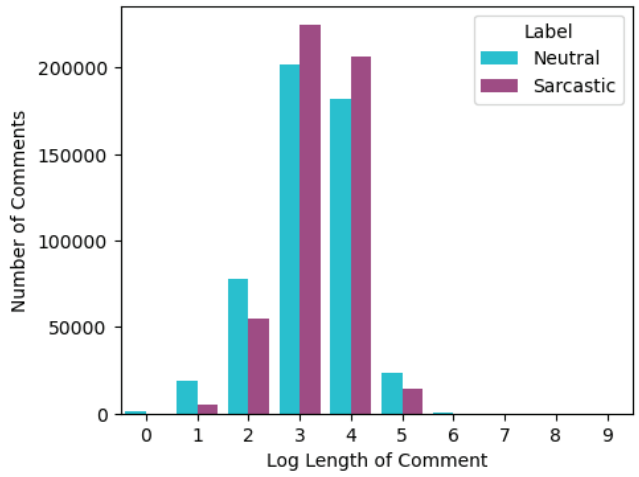


Fig. 7. Log Lengths of all comments.

It is inferred that most comments have log lengths between three and four.

### C. Evaluation Metrics

Many different types of metrics can measure performance. For example, the original research paper of Mikhail Khodak et al. [3] uses a Normal Classification score.

If a class imbalance exists in a dataset, accuracy may not be an appropriate metric, as the model can become biased towards the majority class. However, in the current dataset, nearly equal numbers of data points exist for both class labels. Thus, accuracy can be used as an evaluation metric. Evaluation of models is done on the basis of four metrics:

- *Accuracy*: Used to measure the percentage of accurate predictions a model makes. The formula is given in (1).

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ predictions} \quad (1)$$

- *Precision*: Measures the proportion of accurate positive predictions among positive model predictions. The formula is given in (2).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (2)$$

- *Recall*: The proportion of correctly predicted positive instances among all accurate predictions. The formula is given in (3).

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (3)$$

- *F1 Score*: Harmonic mean of precision and recall is used to provide more information about classifier accuracy when both evaluation metrics are significant. The formula is given in (4).

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

## V. Results and Discussion

A study has been performed to compare metrics of different machine learning models and deep learning techniques to detect sarcasm in Reddit comments using SARC dataset.

Table 2. displays the performance of various models, including Logistic Regression, Support Vector Machines, Decision Tree, Naïve Bayes, CNN, LSTM, and a combination of CNN and LSTM to detect sarcasm using multiple evaluation metrics accuracy, precision, recall, and F1 score.

TABLE II. Comparison of Performance of Multiple Models for Detecting Sarcasm using various Metrics

| Models | Evaluation Metrics | | | |
|---|---|---|---|---|
| | *Accuracy* | *Precision* | *Recall* | *F1 Score* |
| **Support Vector Machines** | 0.7068 | 0.6896 | 0.7149 | 0.7021 |
| **Naïve Bayes** | 0.7085 | 0.6923 | 0.7162 | 0.7041 |
| **Decision Tree** | 0.6425 | 0.6265 | 0.6481 | 0.6371 |
| **Logistic Regression** | 0.7227 | 0.6908 | 0.7387 | 0.7139 |
| **CNN** | 0.7292 | 0.6987 | 0.7339 | 0.7159 |
| **LSTM** | 0.7213 | 0.6719 | 0.7295 | 0.6995 |
| **CNN + LSTM** | 0.7322 | 0.7009 | 0.7316 | 0.7159 |

Fig. 8. displays the evaluation metrics of different models on the validation set.

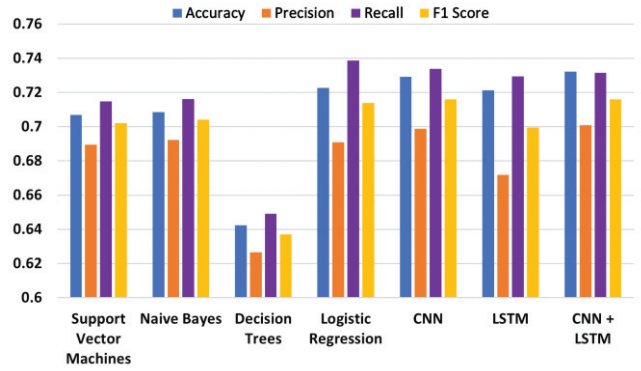


Fig. 8. Evaluation Metrics of Classifiers on Validation Data.

Overall, the model designed using a combination of CNN and LSTM techniques outperformed the other models in terms of accuracy, with a score of 73.22%. The logistic regression model, on the other hand, had a relatively high recall of 73.87%, which may be more important depending on the application. The CNN + LSTM model achieved the highest precision of 70.07%. The F1 Scores of all models were relatively similar, with CNN and CNN + LSTM models achieving the highest F1 score of 71.59%. Therefore, the choice of model would depend on the specific goals of the project and the tradeoffs between different evaluation metrics.

## VI. Conclusion

According to our findings, deep learning techniques like CNN and LSTM perform better than more established supervised machine learning algorithms for detecting sarcasm in social media communities and online forums.

Our findings have implications in the field of natural language processing (NLP) and sentiment analysis, as sarcasm detection is an important and challenging problem in these areas because it can help us understand the underlying sentiment and tone of a message, which is critical for effective communication and interpretation of online content preventing misunderstandings and promoting empathy. Our results suggest that deep learning approaches may be comparatively more effective for sarcasm detection in Online Communities like Reddit, where sarcasm is abundant and difficult to detect using traditional approaches.

Future research could explore different techniques, such as using unsupervised learning approaches, deep learning techniques, different types of feature extractions, and incorporating transfer learning techniques. Overall, our research indicates the potential for further study in this field, as well as the effects of deep learning in sarcasm detection on social media platforms.

## References


[1] "Reddit - Dive into anything." [Online]. Available: https://www.reddit.com/

[2] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artif Intell Rev*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.

[3] M. Khodak, N. Saunshi, and K. Vodrahalli, "A Large Self-Annotated Corpus for Sarcasm." arXiv, Mar. 2018. [Online]. Available: http://arxiv.org/abs/1704.05579

[4] F. Hasnat *et al.*, "Understanding Sarcasm from Reddit texts using Supervised Algorithms," *IEEE Region 10 Humanitarian Technology Conference, R10-HTC*, vol. 2022-September, pp. 1–6, 2022, doi: 10.1109/R10-HTC54060.2022.9929882.

[5] P. Nagwanshi and C. E. Veni Madhavan, "Sarcasm Detection Using Sentiment and Semantic Features," in *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1*, in IC3K 2014. Setubal, PRT: SCITEPRESS - Science and Technology Publications, Lda, 2014, pp. 418–424. doi: 10.5220/0005153504180424.

[6] A.-C. Băroiu and S. Trausan-Matu, "Automatic Sarcasm Detection: Systematic Literature Review," *Information*, vol. 13, p. 399, Apr. 2022, doi: 10.3390/info13080399.

[7] F. B. Kader, N. H. Nujat, T. B. Sogir, M. Kabir, H. Mahmud, and K. Hasan, "Computational Sarcasm Analysis on Social Media: A Systematic Review," Sep. 2022, Accessed: Apr. 23, 2023. [Online]. Available: https://arxiv.org/abs/2209.06170v2

[8] M. Bouazizi and T. Otsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," *IEEE Access*, vol. 4, pp. 5477–5488, 2016, doi: 10.1109/ACCESS.2016.2594194.

[9] 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.

[10] D. Pelser and H. Murrell, "Deep and Dense Sarcasm Detection," Nov. 2019, Accessed: Apr. 23, 2023. [Online]. Available: https://arxiv.org/abs/1911.07474v2

[11] A. Rahaman and M. Kabir, "Sarcasm Detection in Tweets: A Sarcastic Feature Based Approach Using Supervised Machine Learning Model," 2021.

[12] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright, "Sarcasm detection using machine learning algorithms in Twitter: A systematic review," *International Journal of Market Research*, vol. 62, no. 5, pp. 578–598, 2020, doi: 10.1177/1470785320921779.

[13] A. Mohan, A. M. Nair, B. Jayakumar, and S. Muraleedharan, "Sarcasm Detection Using Bidirectional Encoder Representations from Transformers and Graph Convolutional Networks," *Procedia Comput Sci*, vol. 218, pp. 93–102, Jan. 2023, doi: 10.1016/J.PROCS.2022.12.405.

[14] D. M. Kumar and A. Patidar, "Sarcasm Detection Using Stacked Bi-Directional LSTM Model," *Proceedings - 2021 3rd International Conference on Advances in Computing, Communication Control and Networking, ICAC3N 2021*, pp. 1–5, 2021, doi: 10.1109/ICAC3N53548.2021.9725488.

[15] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput Sci*, vol. 2, no. 3, p. 160, 2021, doi: 10.1007/s42979-021-00592-x.

[16] J. Zhang, Y. Li, J. Tian, and T. Li, "LSTM-CNN Hybrid Model for Text Classification," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2018, pp. 1675–1680. doi: 10.1109/IAEAC.2018.8577620.

[17] J. H. Dewan and S. D.Thepade, "How Scopus is Shaping the Research Publications of Feature Fusion-Based Image Retrieval," *Library Philosophy and Practice*, vol. 2021, pp. 2–15, 2021.