Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2026.1993

Research Article

Unmasking sarcasm to understand depression: A machine learning exploration using news headlines

Vanita Ganesh KSHIRSAGAR[1,*], Sunil Kumar YADAV[2], Nikhil KARANDE[3]

[1]*Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University, Rajasthan, India; Dr. D. Y. Patil Institute of Technology, Pune, 411018, India*

[2]*Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University, Rajasthan, 110061, India*

[3]*Cyber Security Consultant, Stickman Cyber, Karnataka, 560037, India*

ARTICLE INFO

Article history
Received: 05 June 2024
Revised: 03 August 2024
Accepted: 26 December 2024

Keywords:
Bidirectional Long Short-Term Memory (BiLSTM); Convolutional Neural Network (CNN) Sarcasm Detection; Long Short-Term Memory (LSTM); News Headlines Dataset

ABSTRACT

People often express thoughts and emotions through text, which can include sarcasm. Sarcasm detection is a difficult task because of noisy social media labels and a lack of high-quality datasets. It is necessary to identify sarcastic texts that affect mental health. The proposed novel method, Convolutional Neural Network and Long-Short Term Memory CNN+LSTM, which utilizes selectively frozen LSTM input layers, enhances sarcasm recognition to identify potential signs of depression, offering a unique approach for more precise mental health evaluation. For identification of sarcastic text, the News headline dataset and Reditt dataset were used together. For identification of sarcastic text, datasets were taken from Kaggle. To evaluate the efficiency of different approaches on sarcastic text, the suggested hybrid approach was compared with state-of-the-art methods like machine learning ML and deep learning DL. The ML and DL methods like logistic regression, decision tree, support vector classifier, multinomial Naïve Bayes, XG Boost, CNN, and LSTM were compared. The hybrid dataset was applied to all ML and DL methods. The proposed method outperformed, with an accuracy of 84%, recall of 83%, precision of 84%, and F1 score of 83% on the hybrid dataset, which is higher than the accuracy obtained from traditional ML methods. The hybrid method is well suited for the complex text. It performs well for detecting sarcastic language features.

Cite this article as: Kshirsagar VG, Yadav SK, Karande N. Unmasking sarcasm to understand depression: A machine learning exploration using news headlines. Sigma J Eng Nat Sci 2026;44(1):478−489.

INTRODUCTION

The sarcasm identification in literature for depression analysis [1,2] can be done in many ways. The conventional ML methods were widely employed for text classification; nevertheless, they frequently proved ineffective when handling sarcastic content. The effectiveness of DL techniques,

***Corresponding author.**
*E-mail address: vanita.kshirsagar@gmail.com

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

particularly CNN and LSTM, has improved when it comes to finding patterns in text data. The majority of earlier research concentrated on techniques for smaller data sets, which constrained the range of text areas. The methods XG Boost and random forest combinely enhance the performance [3,4]. Sarcastic text detection may seem simple, but it's not always that way. Due to the lack of high-quality datasets and noisy labeling from social media sites like Twitter, Instagram, and Facebook, sarcasm detection has proven difficult. These networks frequently employ colloquial language and hashtags, which makes it challenging for DL and ML methods to recognize sarcasm with good accuracy. Dataset labeling by hand can be expensive and lead to short annotator agreements. There is no hybrid dataset; sarcasm detection and depression address the small dataset, and there is also a dearth of research on hybrid approaches [3,4]. Past work on sarcasm detection focused more on accuracy metrics with little attempt of the process's models use to detect sarcasm [5]. This research covers a gap by exploring how CNN+LSTM models identify sarcastic patterns in complex datasets.

This research will make use of advanced machine learning algorithms such as Bi-LSTM and CNN + LSTM. These algorithms have some important architectures used in natural language processing and sequential data analysis. Bi-LSTM enables the model to use the past and future context while processing a sequence and better relate the context to a given task, which could be sentiment analysis, language translation, or even anything of that kind. Another architecture which has garnered attention because of its adaptability in image processing is CNN. Successfully applied to sequential data analysis [6,7]. In NLP, it captures local patterns in text that would be helpful to tasks such as text classification or sentiment analysis [8,9]. Among other modern technologies, CNN, LSTM, and BERT advance the complexity of finding these precise patterns in text and improve text understanding through the context. Techniques in sequential modeling could capture complexities of contextual use in sarcasm, and pre-trained models may be fine-tuned using datasets specifically designed to emphasize sarcasm. Sarcastic language processing and the application of appropriate measures for determining the performance of the model make the systems for sarcasm detection even more efficient [10,11]. LSTM is a type of recurrent neural network (RNN) designed to capture long-term dependencies within sequences. It is significantly better for language modeling or text generation by retaining and utilizing information over longer spans of text. These architectures were trained on extensive datasets comprising annotated examples of sarcastic and non-sarcastic headlines. By utilizing the capabilities of these architectures to capture hidden patterns and context dependencies in text data. The aim is to enhance the computational understanding of sarcasm.

The comparison of classic ML methods like multinomial naïve bayes classifiers, decision trees, random forests, logistic regression, and support vector machines.

- By doing a comparative study and evaluating the performance of advanced architectures alongside traditional methods to demonstrate their effectiveness in recognizing sarcasm.
- A new dimension can be added to the existing work to treat complexities involving sarcasm identification in short and rich context data by focusing on the frozen input layer of LSTM.

The paper is sectioned into four sections which include literature review, dataset description, research methodology, model interpretation and result analysis with implications of our findings for the future of sarcasm detection.

Literature Review

This has become a significant challenge for the detection of sarcasm in text due to the indirect nature of expression found in sarcastic speech often made in informal language, especially on social media sites. Various angles have been addressed by researchers through both machine learning (ML) and deep learning (DL) techniques in addition to rule-based systems in order to improve identification of sarcastic content.

Author found the use of rhetorical questions to attempt to solve sarcasm in text; on the other hand, since much of sarcasm relies on nuances of the greater context of communication, it is quite hard for machines to perform such acts. It has been observed that techniques like SVM and Random Forests have also been used to identify sarcasm, which however loses precision when it is deployed in the case of multiturn conversations where the context itself changes with the passage of time [4]. In order to solve this issue, it was shown that implementing deep learning techniques significantly upgraded the detection of sarcasm because these techniques can grasp nuances of the discourse as well as features over multiple turns in a conversation [2].

This has been further refined with annotated datasets from platforms such as Twitter to improve the performance of deep learning models. Attention-based models were adopted as a way of improving this, significantly enhancing the accuracy of sarcasm detection compared with traditional models and outperforming these traditional models in complex data environments [12]. Additionally, a behavioral approach combined with machine learning was used to detect sarcasm in tweets, highlighting the value of historical context and conversation history in improving detection [13,14].

The growing need to detect sarcasm in social media posts as an indicator of mental health, particularly depression, has been recognized in several studies. By analyzing datasets from platforms like Reddit and Twitter, models were able to achieve an 80% accuracy rate in detecting depressive posts using ML methods [1]. Sarcastic seentence tend to mislead machines and humans due to their irony behaviour. The methods using SVM and Random Forests have shown accuracy rates of 70 to 75% for identifying sarcasm in tweets [2]. The combined convolutional neural networks (CNN) with RNNs, specifically LSTMs, in an ensemble model. Which has demonstrated an 83% accuracy rate in detecting sarcasm [3]. In contrast, the CASCADE method, which integrates context with conversation history, outperformed traditional methods, achieving an accuracy of 85% in sarcasm detection [2].

Yet another research approach developed an innovative hybrid model that incorporated the CNN and LSTM network combining word and emoji embeddings for context-aware sarcasm detection. The model was able to achieve great success with an accuracy of 97.35% with F-score, precision, and recall highly outperforming the traditional models, especially by utilizing emojis to illustrate intent while interpreting sarcastic tweets [15]. In this region, more research is carried out in combining deep learning with the NLP techniques to perform better. MFO has been utilized with a MHSA-GRU model with the task of sarcasm detection that outperformed all models, and further results were checked on Kaggle's News Headlines dataset [16].

Apart from these developments, a many other studies have tested the strength of different feature extraction methods. One work proposed a hybrid combination of TF-IDF, dependency parsing bigrams, and lexicon-based features. This achieved 95.60% accuracy for detecting sarcasm using SVM over social media datasets including Twitter [17,18]. Another was the detection of sarcasm in the MUStARD dataset using an attention-based LSTM model, which obtained an F1 score of 60.1, that beat other deep learning models for sarcasm-context detection [19]. Key role of context was observed in sarcasm detection as shown by the research on sarcasm detection in Hinglish, which included sarcastic posts from social media posts. The Extra Tree Classifier and the Gradient Boosting Classifier ranked high. Their F-scores were 95.43 and 95.29, respectively. These studies highlight that in the detection of sarcasm, besides a linguistic feature representation, some sort of social feature has to be taken into consideration as well [20,21].

One major problem with sarcasm detection lies in the case of news headlines, where sarcasm goes unnoticed and sometimes results in false results in false information. The researchers found it possible to use neural networks particularly RNN and LSTMs for detecting sarcasm among news headlines. The current model achieved 85 percent, showing a high accuracy of LSTM networks while reading out sarcasm from short, context-dependent texts of news headlines [22]. Also, a few have proposed architectures in deep learning for detecting sarcasm in news headlines applying multiple datasets such as The Onion and HuffPost. The applied models used LSTM towards temporal dependencies capturing in various architectures. They obtained state-of-the-art accuracy of 0.925 in their test data. These models drastically outperformed the other classifiers which included decision trees, random forests, SVMs and established new benchmarks for the news article classification with respect to sarcasm [23]. Others have examined the integration of deep learning with handcrafted contextual features. Interesting research on the integration of CNN architectures along with crafted contextual features demonstrates that the deep learning along with handcrafted features highly boosts the accuracy, precision, and recall of sarcasm detection in intricate textual data [24]. Finally, a different set of studies provided evidence for further refining sentiment analysis and sarcasm detection by moving beyond more traditional contextual variables, such as the relationship between authors and audiences. A GAT-based framework achieved some breakthroughs by taking into account not only user history but also social dynamics in a conversation. This context-aware model holds immense promise in the prediction of sarcastic intent with an implication toward the significance of social context in the detection of online sarcasm [25].

The other challenge lately identified by studies is the procedure of handling imbalanced datasets, especially when performing sarcasm detection on datasets where there is an imbalance distribution of sarcastic and non-sarcastic examples. An example that used an agglomerative clustering method to identify some of its outliers and eliminate them along with using ensemble techniques like Bi-LSTM networks had an accuracy of 99.3%, which sets up a new benchmark for determining performance in sarcasm detection [26,27].

The issue of distinguishing sarcasm from other types of sentiment, especially in user reviews and posts, was also addressed through the development of new classification systems based on decision trees and neuro-fuzzy rules. These methods, tested on Twitter data. Which helps to produced promising results in classifying sarcastic and non-sarcastic sentences, contributing to the broader field of sarcasm detection [28]. Therefore, the sentiment analysis and sarcasm detection are often treated as separate tasks, but researchers have begun integrating them. A multi-task learning approach proposed by one study achieved an F1-score of 94%. It outperforming traditional methods and showing potential for applications in behavior analysis and sentiment classification [29,30].

Several studies have also examined sarcasm through the lens of its emotional and contextual components, classifying sarcasm into categories such as polite, rude, and raging. The results majorly helpws to identifies the sracsm. Utilizing ensemble feature selection methods, for example, authors were able to achieve 92.7% to 99.79% accuracy across other classes of sarcasm. This further informs emotional subtleties of sarcasm in social media platforms [31].

However, this remains an ongoing challenge but keeps changing to the implementation of new techniques of machine and deep learning. The efforts ranged from the use of attention-based models up to hybrid CNN-LSTM networks and context-aware frameworks that showed context as well as feature extraction to be as integral a part of the algorithm for improving the precision of sarcasm detection. As these methods advance, they promise to improve applications in mental health monitoring, sentiment analysis, even the development of more human-like conversational agents.

Problem Statement

Challenges in sarcasm detection datasets available for the development of the system are of a small size and lack diversity; the contextual factors that are included are very narrow; also, there is a huge impact because of cultural and linguistic variations. All these variations are important, as there is a difference in presentation of sarcasm across cultures, contexts, and communication styles. Differences have been addressed within these presentations to generally extend the capability of the model in being sarcastic in more real scenarios, thus making it more robust and adaptable in different populations. To this end, the News Headlines and Reddit datasets were added, which showed greater diversity and have helped the model capture more nuanced contexts to improve its performance in different styles of communication.

About Dataset

The sarcasm detection dataset is fetched from Kaggle. The news headlines dataset was downloaded from two websites, TheOnion and HuffPost [32,33]. Another dataset is added and named Reditt dataset. The combination of both News Headlines and Reddit datasets uses formal and informal language styles. News Headlines offer fo rmalised, professional language, while Reddit gives it a more casual, everyday tone. This blend results in a balanced dataset that captures a wide range of language, hence leading to a model that understands and adapts better to varied communication styles. The dataset contains a mix of non-sarcastic and sarcastic comments [33]. Twitter datasets are generally noisy regarding labels and text, which makes it hard to classify sarcasm with high accuracy. The intention behind the dataset is to provide good-quality labels with as less noise as possible, since headings of news are written by experts in a formal manner. In the dataset, each record has the following attributes:

'is_sarcastic': Indicates whether the headline is_sarcastic- yes that means 1 or is_sarcastic-no that means 0 (0).

'headline': The heading of the news article.

'article_link': It is the Link to the news article, which helps to gather supplementary data.

The dataset offers a resource for training and evaluating sarcasm detection models. It provide a more reliable and structured source of data compared to Twitter datasets.

Research Methodology

The methodology for the conduct of the study on sarcasm detection in news headline is briefly depicted. This includes many crucial steps such as data collection and preprocessing. The data investigation and imagining model development, and assessment and analysis, as showed by Fig. 1. Each step was carefully designed to ensure that our sarcasm-detection model is robust and effective.

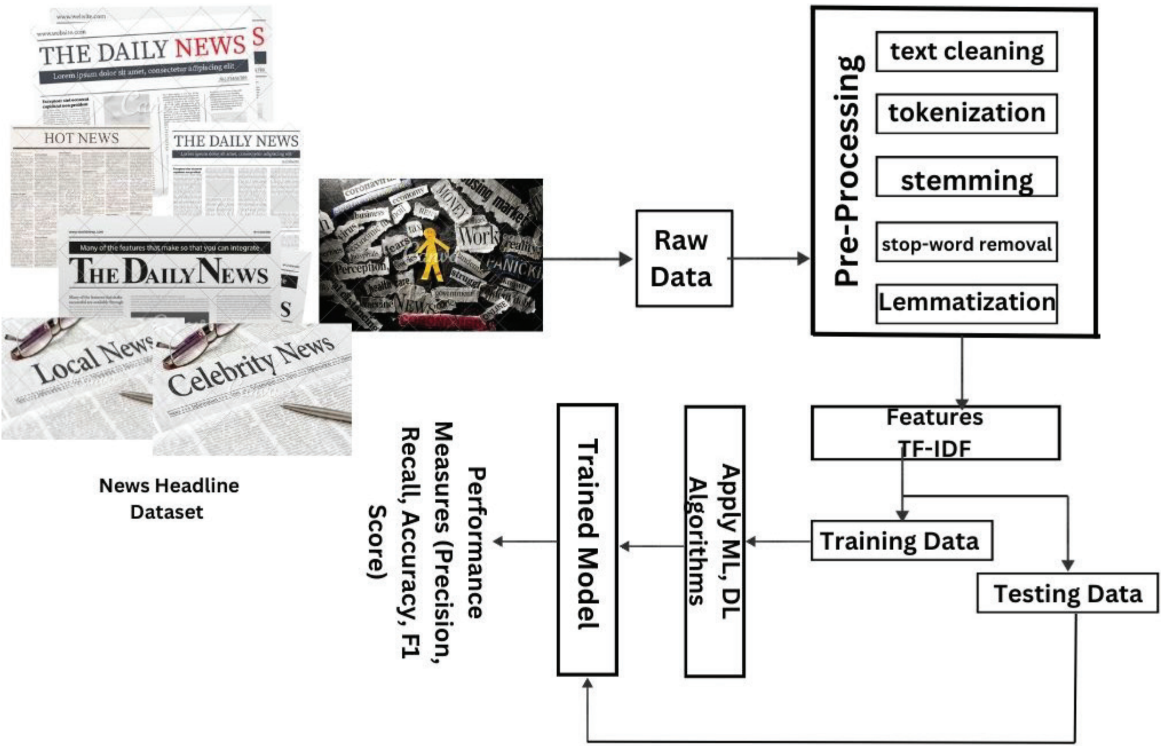

Figure 1. System architecture.

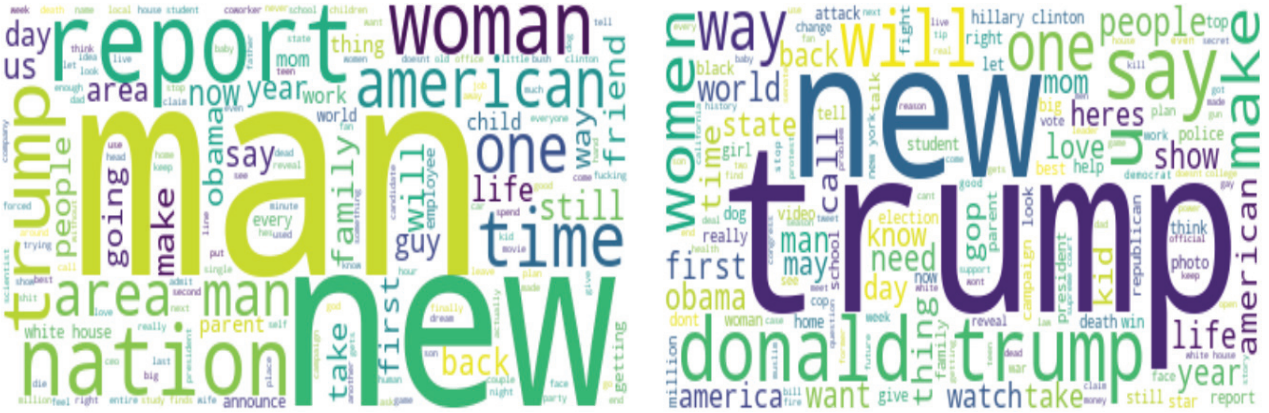

Figure 2. Word cloud of sarcastic headlines (left) and non-sarcastic headlines (right).

Data Collection and Preprocessing

Two such datasets of news headlines were annotated for sarcasm detection from Kaggle according to the study. These datasets are properly analyzed for relevance and content diversity in headlines. After collection, extensive preprocessing steps applied for cleaning textual data. This includes a process such as lowercasing; removal of punctuation, URLs, digits, and emojis; followed by tokenization and stop-word removal for improving the text's suitability for further analysis.

Data Exploration and Visualization

After preprocessing techniques, analyze the nature of datasets. Imagining of data through count plots, pie charts, and word clouds. The distribution of sarcastic and non-sarcastic headlines as represented in Figure 2. The visualizations served an overall description of the dataset and emphasized common words associated with each class of headlines.

Model Development

Construct a sophisticated model with an indepth understanding of the data to detect sarcasm. A custom architecture of the neural network has been developed, which combines embedding layers, bidirectional LSTM networks, dense layers with both batch normalization and dropout, and sigmoid output layer. This architecture is well-designed to catch all the intricate patterns and much-needed context in news headlines.

Embedding Layer

In this layer, it converts the input text data X of shape (batch_size, sequence_length) to dense vectors of the shape (batch_size, sequence_length, embedding_dim) with the help of embedding matrix E.

The word index in the Input mapping will map to the embedding space-dense vector. Construct a sarcasm detection model with a detailed understanding of the information. Developed a custom neural network architecture to fit in embedding layers, bidirectional LSTM networks, dense layers with batch normalization and dropout, and a sigmoid output layer. This architecture was accurately designed to capture the complex patterns and context inherent in news headlines.

BiLSTM architecture consists of several major layers that sequentially process the given input text data to detect sarcasm. The architecture includes the following layers for the performance of sarcasm detection:

Bi-LSTM Architecture

- **Embedding Layer:** In this layer, it converts the input text data X of shape (batch_size, sequence_length) to dense vectors of the shape (batch_size, sequence_length, embedding_dim) with the help of embedding matrix E. The word index in the input mapping will map to the embedding space-dense vector.
  For example, if X is the input text [I, love, sarcasm] and E is an embedding matrix, then the output X_embedded will be a tensor containing dense vectors for each word.
  In this embedding freezing the input layer means leveraging the knowledge embedded in those embeddings. This can help maintain the richness of the pre-trained representations, which could be especially useful for detecting nuanced features in sarcasm and depression.
- **Spatial Dropout1D Layer:** This layer randomly drops entire 1D feature maps (words) along the sequence dimension. Let X_embedded be the input tensor with shape (batch_size, sequence_length, embedding_dim), and dropout_rate be the dropout rate.
  For example, if X_embedded is the embedded input tensor, applying spatial dropout would randomly drop certain words from the sequence.
- **Bidirectional LSTM Layers:** This layer processes the input sequence X_dropout in both forward and backward directions. Let h_t^f and h_t^b represent the hidden states at time t for the forward and backward LSTMs, respectively.
  For example, In the sequence "I love sarcasm", a bidirectional LSTM would process it both ways, forwards and in reverse. This will capture context from both the beginning and the end.

- **Dense Layers with Batch Normalization and Dropout:** Let h_concat of the forward and backward hideen states. Apply dense layers to this concatenated representation, followed by batch normalization and dropout. For instance, after concatenation of the forward and backward states of the hidden layers, the output representation is passed to dense layers to generate higher-level features.
- **Flatten Layer:** It will flatten the output of the dense layers into a one-dimensional array. For example, the output from the dense layers, being typically a multi-dimensional tensor, is flattened into a one-dimensional array.
- **Output Layer:** Here, there is only one neuron of type sigmoid. Let h_flat be the flattened representation. Example- For a dense layer, let us take the output. This needs to go through the sigmoid function in order to derive the final prediction probability for sarcasm.
- **Equations involved for BiLSTM layered architecture:** For example, Given the sequence "I love sarcasm", the bidirectional LSTM would process it in both forward and backward directions, capturing context from both ends. Equations representing BiLSTM layered architecture:

1. Embedding Layer:X_embedded = E(X) #(make trainable =False)
2. Spatial Dropout1D Layer: X_dropout = Spatial Dropout1D(X_embedded)
3. Bidirectional LSTM Layers: h_t^f = LSTM(X_dropout) and h_t^b = LSTM (reverse(X_dropout))
4. Dense Layers with Batch Normalization and Dropout: h_concat = Concatenate (h_t^f, h_t^b) and h_dense = Dense(h_concat)
5. Flatten Layer: h_flat = Flatten(h_dense)
6. Output Layer: y_hat = σ(Dense(h_flat))

The Table 1. Represents the Sequential BiLSTM model summary.

CNN and LSTM Architecture

The input layer freeze, which is trainable=False, within the embedding layer forms one of the very important architectural choices in our hybrid CNN-LSTM model for sarcasm detection [34]. This will preserve the semantic relationships of the pre-trained word embeddings while allowing subsequent layers to adapt to the features that comprise sarcasm [35,36]. The frozen embedding layer can therefore be represented as:

E(x) = W_e * x, where ∂E/∂W_e = 0

Where:

- E(x) represents the embedding function
- W_e represents the embedding weights matrix
- x represents the input token
- ∂E/∂W_e = 0 indicates the gradient with respect to embedding weights is zero during backpropagation

This frozen architecture allows the model to maintain stable semantic relationships while allowing the CNN and LSTM layers to focus on learning about higher-level sarcastic features and further optimization of the model's ability to detect subtlety in linguistic patterns and emotional undertones without loss of foundational language understanding encoded into these pre-trained embeddings [37].

This model should generalize well on different text samples without memorizing non-informative patterns. Freezing the input layer also reduces the number of

Table 1. BiLSTM sequential_2 Model summary

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 25, 100)	2253100
spatial_dropout1d_2 (SpatialDropout1D)	(None, 25, 100)	0
bidirectional_4 (Bidirectional)	(None, 25, 392)	465696
bidirectional_5 (Bidirectional)	(None, 392)	923552
dense_4 (Dense)	(None, 64)	25152
flatten_2 (Flatten)	(None, 64)	0
dense_5 (Dense)	(None, 1)	65

Table 2. CNN+LSTM sequential_2 Model summary

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 100, 32)	320000
conv1d (Conv1D)	(None, 96, 32)	5152
max_pooling1d (MaxPooling1D)	(None, 24, 32)	0
Lstm_2 (LSTM)	(None, 64)	24832
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65

parameters that the model needs to update during training, thus resulting in faster convergence and more efficient learning where the model can focus more on learning higher-level features by looking at the CNN and LSTM layers.

Equations representing CNN + LSTM layered architecture:

1. X_embedded = E(X) #(make trainable =False)
2. X_conv = Conv1D(X_embedded)
3. X_pooled = MaxPooling1D(X_conv)
4. h_t^f = LSTM(X_pooled)
5. y_hat = σ(Dense(h_flat))

The Table 2. Represents the CNN+LSTM sequential model summary.

RESULT AND DISCUSSION

The news headline contains a sarcastic statement. The dataset of news headlines is taken from Kaggle. Which is approximately 200000 news headlines. The dataset is the diverse dataset. Which contains news, including sports, technology, entertainment, and politics. The dataset was divided into training data and test data. The preprocessing methods were used, such as text cleaning, tokenization, stemming, and stop-word removal. To convert text into vectors, the TF-IDF method is used. This section discusses the results achieved with BiLSTM, CNN + LSTM, and other machine learning models.

- The BiLSTM model exhibited promising performance over 10 epochs, with initial accuracy of 66.3% on training data and 74.5% on validation data, improving to 85.1% and 84%, respectively, by the end of the training period, as shown in Figure 3. This indicates that the model effectively learned from the data and generalized well to unseen examples, as evidenced by the comparable performance on both training and validation datasets. The loss function decreased consistently throughout training, further demonstrating the model's learning capability. Figure 4. represents the confusion matrix with true label and corrected label.

Over 10 epochs, the BiLSTM model showed strong improvement. Starting with 40.45% accuracy on training data and 62.24% on validation, it reached 97.51% and 91.91%, respectively, by the final epoch (Fig. 5). With a larger dataset, these results surpassed those in Figure 4.

The BiLSTM model outperforms traditional models like Decision Trees and Random Forests because it can capture the sequential nature of text, understand context, and learn complex relationships between words. Traditional models, though effective for some tasks, struggle to grasp the

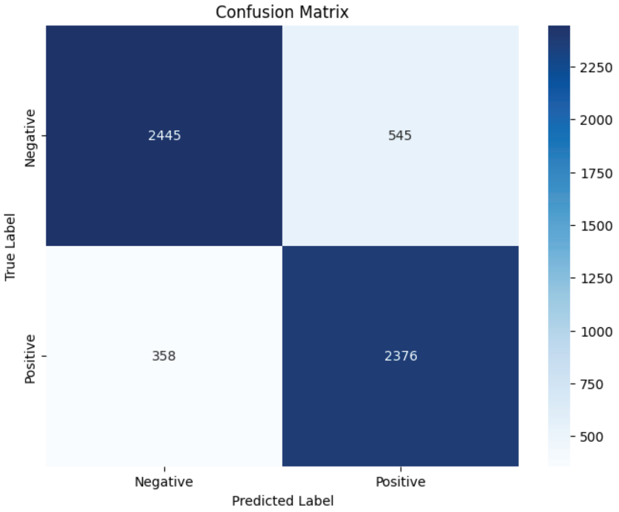

Figure 4. Confusion matrix for true and predicted Label.

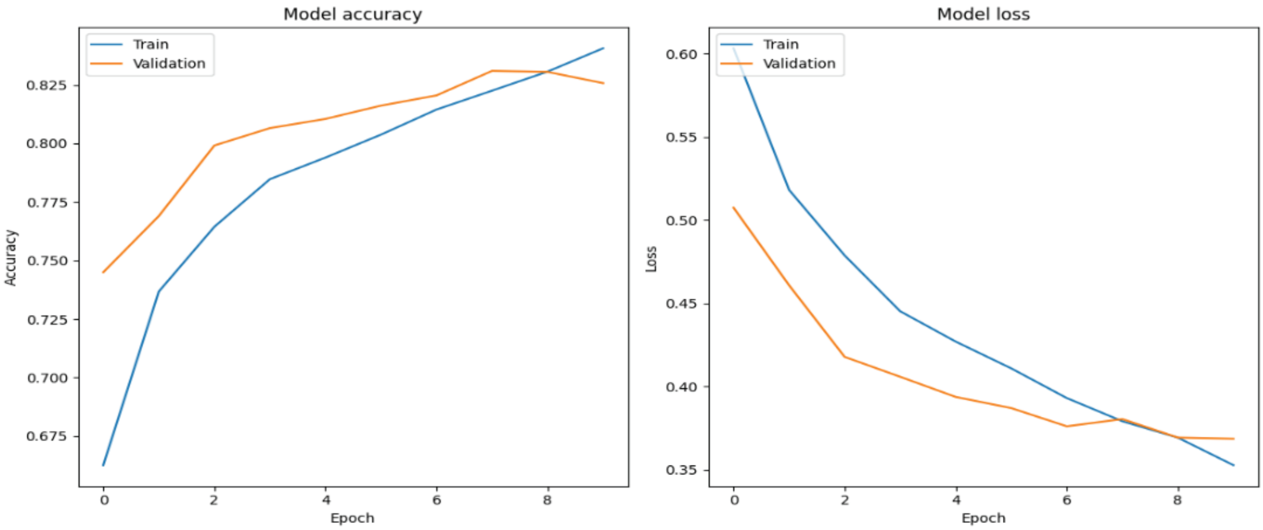

Figure 3. BiLSTM model train verses validation – accuracies (left) and losses (right).

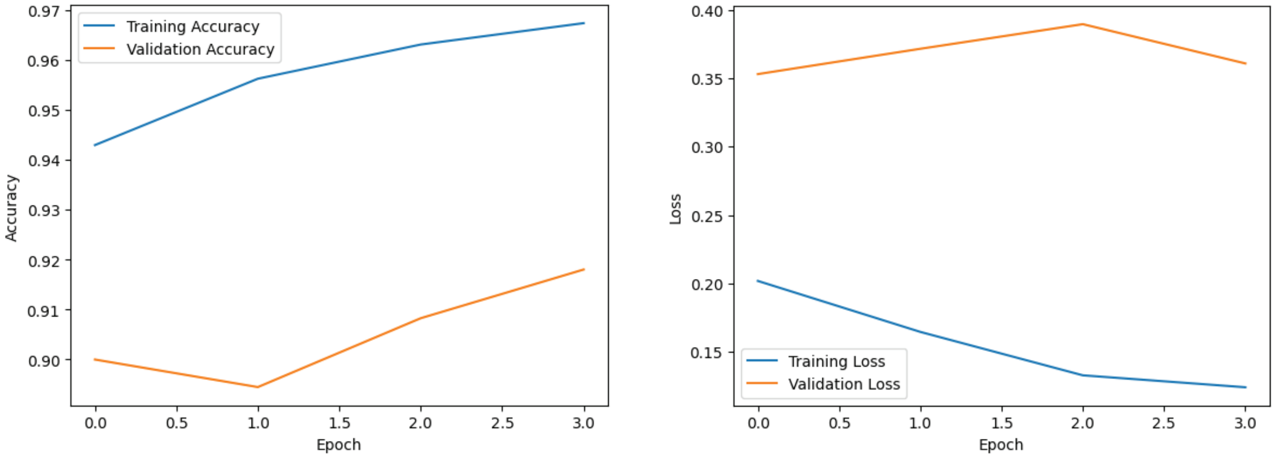

Figure 5. BiLSTM model train verses validation – accuracies (left) and losses (right) hybrid dataset.

nuances of text data. BiLSTM, with its deep learning capabilities, is better suited for sarcasm detection, where meaning depends on context, tone, and subtle word connections.

6.2 The CNN + LSTM model performed well across five training epochs. Initially, it achieved 76.3% accuracy on the training data and 84.6% on the validation data. By the final epoch, these scores reached 99.3% for training and 83% for validation, though minor overfitting was noted. The loss function stayed low, reflecting the model's consistent accuracy, as illustrated in Figure 6.

The classification reports of all models evaluated in sarcasm detection provide valuable insights into their performance (Fig. 7). A comprehensive overview of each model's recall, precision, accuracy, and F1-score, comparison is shown in the Table 3 to show their effectiveness in detecting sarcasm.

From this analysis, several key observations can be made:

1. Logistic Regression: It was able to yield 83.23% accuracy. Both the precision, recall, and F1-score were in balance with respect to both classes. Distribution in the confusion matrix was highly reasonable. This would suggest it generally performed better.
2. Decision Tree: The accuracy of the model was 73.43%. The precision, recall, and F1 scores were acceptable but had false positives and false negatives.

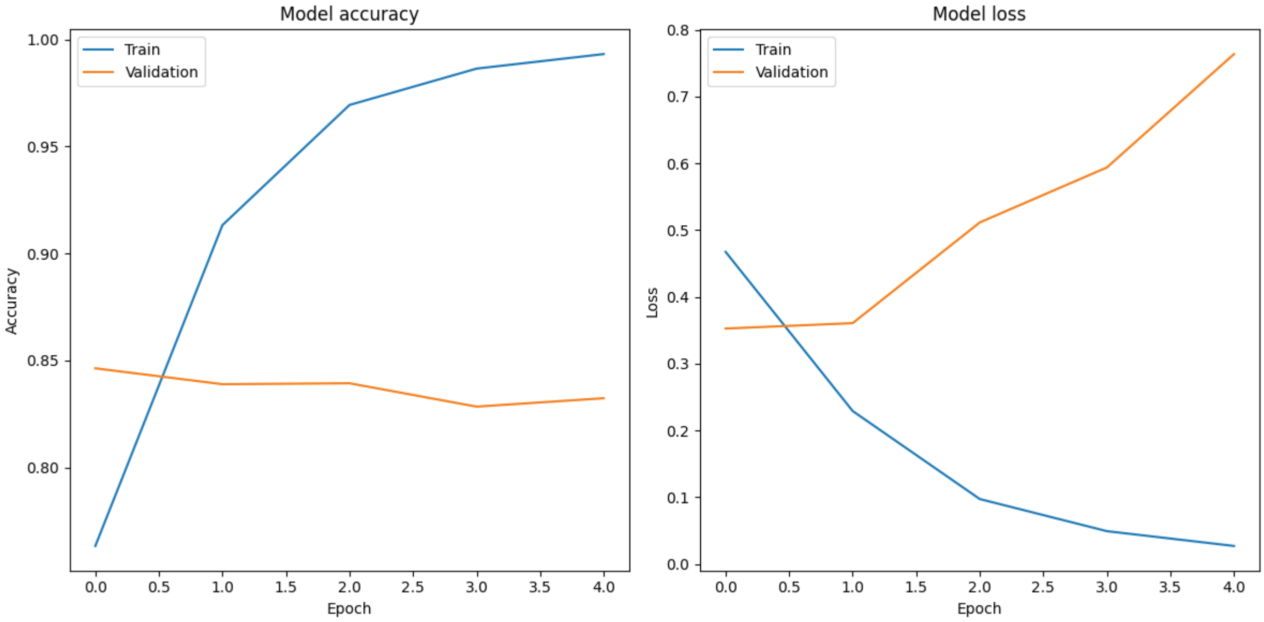

Figure 6. CNN + LSTM model train verses validation – accuracies (left) and losses (right).

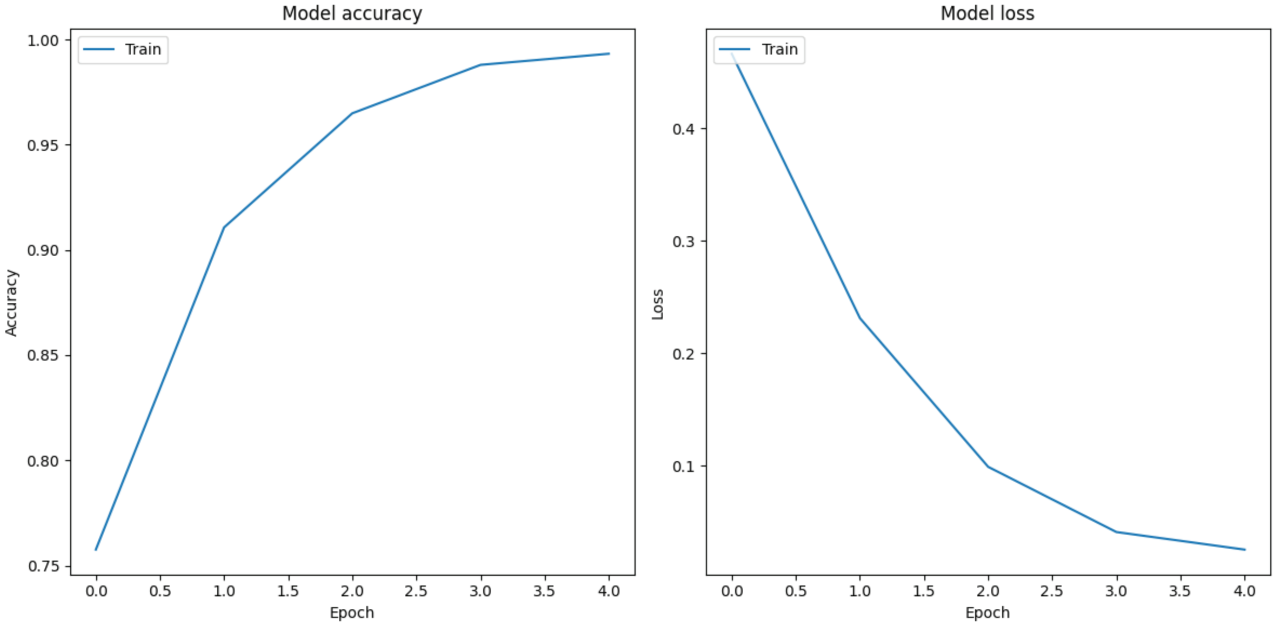

Figure 7. CNN+LSTM Train data accuracy and loss for 5 epoch.

Table 3. Performance Comparison of Classification Algorithms for Sarcasm Detection

Algorithm	Precision	Recall	F1-Score	Accuracy
Logistic regression	0.83	0.83	0.83	0.83
Decision tree	0.73	0.74	0.74	0.74
SVM classifier	0.82	0.82	0.82	0.82
MNB classifier	0.82	0.82	0.82	0.82
Random forest	0.79	0.79	0.79	0.79
LSTM	0.72	0.76	0.72	0.72
CNN+LSTM Classifier	**0.84**	**0.83**	**0.83**	**0.83**
Bi-LSTM	0.83	0.82	0.82	0.82
CNN+LSTM+Bi-LSTM	0.83	0.82	0.82	0.82

3. Support Vector Machine also closely had similar accuracy with logistic regression. It obtained an accuracy of 82.77% and had balanced performance metrics. The confusion matrix showed a balanced number of true positives and true negatives, which confirmed the robustness of the model in the classification process.
4. Multinomial Naive Bayes had an accuracy of 82.41%. The performance metrics were just about as good as logistic regression and SVM. The confusion matrix also showed that there were good numbers of true positives and true negatives in it.
5. The result of the random forest test is 79.35% accuracy, which is lesser than in any of the previous models. A greater number of false positives and false negatives are seen in the confusion matrix. Thus, it so happens that results obtained through random forest happen to be less optimal as compared to logistic regression and SVM.
6. With an accuracy of 98.00% in the model's trainability and an accuracy of 84.00% on testing CNN + LSTM Classifier, the balancing in its precision, recall, and F1-scores for both classes shows a great generalization to unknown data.
7. The model overfits a little just because of the lesser set of trainings and sometimes even makes judgement by tone or topic or context.
8. BiLSTM Classifier gave 84.00% accuracy, and precision and recall are balanced with F1-scores for both classes. There was a slight imbalance in the precision and recall of both classes, but the model had overall robust performance.

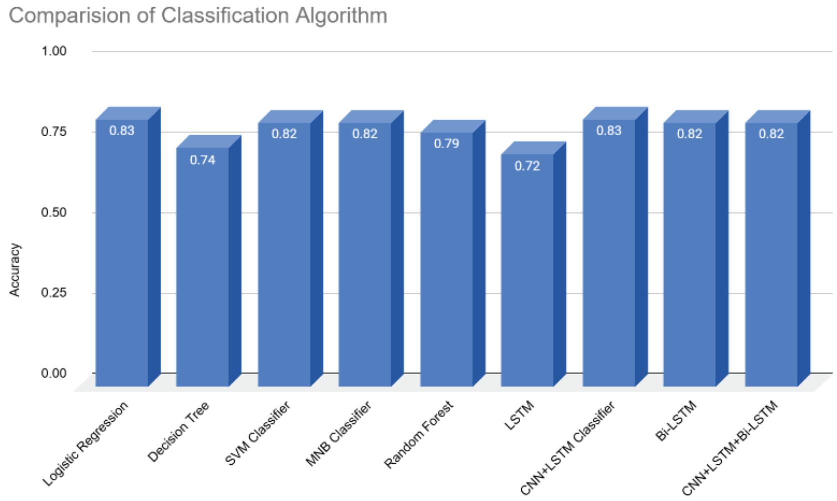

Figure 8. Accuracy comparison of classification algorithms for sarcasm detection.

9. BiLSTM Classifier gave 98.00% accuracy, and precision, recall, along with F1-scores for both classes were there as their concatenation with the Raditt dataset is available. The Training accuracy is 91.80%.

The related comparison of the accuracy from classification algorithms as presented in Fig. 8 gives a relatively clear view related to the performance regarding the detection of sarcasm from each model. It is thus crucial to determine the strength and weaknesses of a particular model so further efforts in optimizing and refining the efforts in detecting sarcasm can be made.

Logistic regression achieved an accuracy of 83.23% with relatively balanced precision, recall, and F1 scores for both classes. Balancing characteristics are portrayed by the conf When all classifiers are considered, the BiLSTM model falls in the range of well performing algorithms along with logistic regression, SVM, multinomial naive Bayes, and the CNN + LSTM Classifier. Decision tree and random forest algorithms showed potential issue areas with relatively lower performance.

CONCLUSION

Hence, it can be concluded that the study demonstrates the effectiveness of the BiLSTM and CNN + LSTM models in identifying sarcasm with maximum accuracy. These models have a good understanding and analysis of attaining complex patterns of sarcasm in textual data by developing great robustness and reliability. Traditional machine learning approaches, though outperformed by deep learning models, emphasized exploitation of advanced techniques in tasks that involve sarcasm detection. Hybrid models and ensemble techniques are candidates for finding even better accuracy and generalization in future sarcasm detection systems. New possibilities open up for improvements in natural language understanding with the help of overcoming the above limitations. Future directions are outlined here, but can be even more specific in listing several ways that will make sarcasm detection even better. Traditional machine learning approaches were outperformed by deep learning models.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article. ETHICS There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

[1] Tadesse MM, Lin H, Xu B, Yang L. Detection of depression-related posts in social media using machine learning algorithms. Int J Med Inform 2019;127:317–327.

[2] Hazarika D, Poria S, Cambria E, Zimmermann R, Mihalcea R. Cascade: Contextual sarcasm detection in online discussion forums. In: Proc 27th ACM Int Conf Inf Knowl Manag; 2018. p. 1835–1838.

[3] Ghosh D, Veale T. Fracking sarcasm using neural network ensembles In Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. San Diego, California: Association for Computational Linguistics; 2016. p. 161–169. [CrossRef]

[4] Oraby S, Harrison V, Reed L, Walker M. Are you serious?: Rhetorical questions and sarcasm in social media dialog. In: Jokinen K, Stede M, DeVault D, Louis A, editors. Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue. San Diego, California: Association for Computational Linguistics; 2017. p. 11–17.

[5] Misra R, Arora P. Sarcasm detection using news headlines dataset. AI Open 2023;4:13–18. [CrossRef]

[6] Ahmed MF, Mohanta JC, Sanyal A. Inspection and identification of transmission line insulator breakdown based on deep learning using aerial images. Electr Power Syst Res 2022;211:108199. [CrossRef]

[7] Kshirsagar VG, Yadav S, Karande N. Feature fusion and early prediction of mental health using hybrid squeeze-MobileNet. In: Garg D, Rodrigues JJPC, Gupta SK, Cheng X, Sarao P, Patel GS, editors. Advanced computing. Commun Comput Inf Sci. Cham: Springer; 2024. p. 2053. [CrossRef]

[8] Amasala L, Ponnuru M, Srideviponmalar P. Secure goods storage and anti-theft approach using Ethereum blockchain. Procedia Comput Sci 2024;233:1–11. [CrossRef]

[9] Kshirsagar VG, Sharma S. Review paper on sentimental analysis for recommendation system. In: Patil VH, Dey N, Mahalle PN, Pathan MS, Kimbahune VV, editors. Proc First Doctoral Symp Nat Comput Res. Lect Notes Netw Syst. Singapore: Springer; 2021. p. 169.

[10] Kiran SK, Shashi M, Madhuri KB. Multi-stage transfer learning for fake news detection using AWD-LSTM network. Int J Inform Technol Comput Sci 2022;14:58–69.

[11] Sirra K, Mogalla S, Madhuri KB. CSSLnO: Cat swarm sea lion optimization-based deep learning for fake news detection from social media. Int J Inf Technol 2024;16:4225–4241. [CrossRef]

[12] Tay Y, Tuan LA, Hui SC. Reasoning with sarcasm by reading in-between. In: Proc 56th Annu Meet Assoc Comput Linguist; 2018. p. 1010–1020. [CrossRef]

[13] Rajadesingan A, Zafarani R, Liu H. Sarcasm detection on Twitter: A behavioral modeling approach. In: Proc 8th ACM Int Conf Web Search Data Min; 2015. p. 97–106. [CrossRef]

[14] Khattri S, Koul A, Bansal A, Shrivastava S. Sarcasm detection: A computational approach. arXiv 2015. Preprint. doi: 10.48550/arXiv.2107.02276

[15] Pane SF, Ramdan J, Putrada AG, Fauzan MN, Awangga RM, Alamsyah N. A hybrid CNN-LSTM model with word-emoji embedding for improving the Twitter sentiment analysis on Indonesia's PPKM policy. In: Proceeding 6th International Conference on Information Technology, Information Systems and Electrical Engineering. Yogyakarta, Indonesia: IEEE; 2022. p. 1–6. [CrossRef]

[16] Sait ARW, Ishak MK. Deep learning with natural language processing enabled sentimental analysis on sarcasm classification. Comput Syst Sci Eng 2023;44:2553–2567. [CrossRef]

[17] Rajani B, Saxena S, Kumar BS. Design and analyze the machine learning based sarcasm prediction model for social media context. Res Artic Spec Issue Appl Adv Eng Technol 2023;11:156–163.

[18] Kshirsagar VG, Yadav S, Karande N. HMHDTML: Human mental health detection using text and machine learning model. In: Kumar R, Verma AK, Verma OP, Wadehra T, editors. Soft computing: Theories and applications. Lect Notes Netw Syst. Singapore: Springer; 2024. p. 970. [CrossRef]

[19] Băroiu AC, Trăuşan-Matu Ş. Comparison of deep learning models for automatic detection of sarcasm context on the MUStARD dataset. Electronics 2023;12:666. [CrossRef]

[20] Singh R, Srivastava R. Extracting contextual feature from Hinglish short text by handling spelling variation at character and word level. Int J Intell Syst Appl Eng 2023;11:713–719.

[21] Olaniyan D, Ogundokun RO, Bernard OP, Olaniyan J, Maskeliūnas R, Akande HB. Sarcasm and irony detection in social media. Computers 2023;12:231. [CrossRef]

[22] Sinha S, Choudhary M. Sarcasm detection using deep learning approaches: A review. Int J Recent Technol Eng 2023;11:50–58.

[23] Ali R, Farhat T, Abdullah S, Akram S, Alhajlah M, Mahmood A, Iqbal MA. Deep learning for sarcasm identification in news headlines. Appl Sci 2023;13:5586. [CrossRef]

[24] Razali MS, Halin AA, Ye L, Doraisamy S, Norowi NM. Sarcasm detection using deep learning with contextual features. IEEE Access 2021;9:68609–68618. [CrossRef]

[25] Plepi J, Flek L. Perceived and intended sarcasm detection with graph attention networks. In: Moens M-F, Huang X, Specia L, Yih SW, editors. Findings of the Association for Computational Linguistics: EMNLP. Punta Cana, Dominican Republic: Association for Computational Linguistics; 2021. p. 4746–4753. [CrossRef]

[26] Maheswari SU, Dhenakaran SS. Improved ensemble based deep learning approach for sarcastic opinion classification. Multimedia Tools Appl 2023;83:38267–38289. [CrossRef]

[27] Prasanna MSM, SG SS, Ayyasamy V. Polarity classification on Twitter data for classifying sarcasm using clause pattern for sentiment analysis. Multimedia Tools Appl 2023;82:1–37. [CrossRef]

[28] Tan YY, Chow CO, Kanesan J, Chuah JH, Lim Y. Sentiment analysis and sarcasm detection using deep multi-task learning. Wirel Pers Commun 2023;129:2213–2237. [CrossRef]

[29] Pandey AC, Seth SR, Varshney M. Sarcasm detection of Amazon Alexa sample set. In: Advances in signal processing and communication. Singapore: Springer; 2019. p. 559–564. [CrossRef]

[30] Sundararajan K, Palanisamy A. Multi-rule based ensemble feature selection model for sarcasm type detection in Twitter. Comput Intell Neurosci 2020;2020:2860479. [CrossRef]

[31] Misra R, Arora P. Sarcasm detection using news headlines dataset. AI Open 2023;4:13–18. [CrossRef]

[32] Misra R, Grover J. Sculpting data for ML: The first act of machine learning. 1st ed. Independently published. 2021.

[33] Khodak M, Saunshi N, Vodrahalli K. A large self-annotated corpus for sarcasm. arXiv 2017. Preprint. doi: 10.48550/arXiv.1704.05579

[34] Gupta I, Chatterjee I, Gupta N. Enhanced deep learning-based model for sentiment analysis to identify sarcasm appeared in the news. Comput Artif Intell 2023;42:1091–1113. [CrossRef]

[35] Galal AM, Yousef MH, Zayed HH, Medhat W. Arabic sarcasm detection: An enhanced fine-tuned language model approach. Ain Shams Eng J 2024;15:102736. [CrossRef]

[36] Najafabadi MK, Ko TZC, Chaeikar SS, Shabani N. A multi-level embedding framework for decoding sarcasm using context, emotion, and sentiment feature. Electronics 2024;13:4429. [CrossRef]

[37] Aksan F, Li Y, Suresh V, Janik P. CNN-LSTM vs LSTM-CNN to predict power flow direction: A case study of the high-voltage subnet of northeast Germany. Sensors 2023;23:901. [CrossRef]