Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 235 (2024) 1058–1067

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering (ICMLDE 2023)

An Efficient Sarcasm Detection using Linguistic Features and Ensemble Machine Learning

Jitesh Pradhan*, Rajshree Verma, Sumit Kumar, Varun Sharma

Department of Computer Science and Engineering, National Institute of Technology (NIT) Jamshedpur, Jharkhand - 831014, India.

Abstract

Sarcasm detection in written text has emerged as a significant research area within natural language processing (NLP). Sarcasm, characterized by conveying the opposite of the intended meaning often for humor, irony, or ridicule, poses a challenge due to its contextual and tonal nuances. This study investigates the application of machine learning methods to detect sarcasm in text due to its potential to reverse the overall sentiment expressed in a sentence. A total of 13 linguistic manually crafted features related to text meaning, word usage, lexical diversity, and readability are extracted. These features are then employed to train a variety of machine learning models including Gradient Boosting, Decision Tree, Random Forest, Support Vector Machine, Gaussian Nave Bayes, K-Nearest Neighbor, and Logistic Regression classifiers. Additionally, an Ensemble Model and a Dense Neural Network is developed, both trained on the extracted handcrafted features to showcase performance. The results reveal that the suggested ensemble model achieves a peak accuracy of 93% in sarcasm detection. The amalgamation of these 13 linguistic features enhances model performance when compared to other contemporary models, exhibiting an improvement of up to 5% in terms of F1-score using the publicly available gold standard News Headline Dataset.

Keywords: Dense Neural Network; Ensemble Learning; Linguistic Features; Sarcasm; Semantic Meaning;

1. Introduction

The advent of the social media era has transformed communication, allowing individuals to express themselves effortlessly through a simple tap [1]. Platforms like Twitter and Facebook have become hubs for the pervasive use of sarcasm, a linguistic tool conveying humor, disdain, or negative sentiments via exaggerated expressions. While seemingly polite, sarcasm can unwittingly incite anger. It often targets political parties and celebrities, key influencers in public opinion. Interestingly, sarcasm correlates with psychological aspects; during periods of depression or moderate anxiety, individuals tend to employ sarcasm more in their online interactions [2]. Additionally, a recent study founds

* Corresponding author. Tel.: +91-958-963-3179.
E-mail address: jiteshpradhan.cse@nitjsr.ac.in, jitpradhan02@gmail.com

1877-0509
10.1016/j.procs.2024.04.100

that those delivering teasing remarks view them as less hurtful than perceived by recipients [2], despite these remarks often causing deeper emotional harm [3].

Recognizing sarcasm is straightforward in face-to-face conversations, aided by facial expressions, tone, and gestures. However, in written communication, especially on online platforms, detecting sarcasm becomes intricate due to the absence of these cues. The challenge escalates when identifying sarcasm within shared content like images, videos, or text on social media, where context hinges on additional elements like images or main text/comment/headlines [4, 3]. The significance of detecting online sarcasm has grown due to its relevance in tasks such as identifying fake news, sentiment analysis, opinion extraction, and pinpointing online trolls and cyberbullies on various online platforms, including social media sites, discussion forums, and e-commerce websites [4, 3, 5]. Consequently, the exploration of sarcasm detection has gained prominence as a focal point of contemporary research.

In previous studies, researchers have proposed a range of strategies, encompassing machine learning, deep learning, and hybrid combinations of these techniques, to identify sarcastic sentences [4, 3, 5]. Nonetheless, the direct application of deep learning models might not adequately capture the intricate and nuanced essence of sarcasm within text due to the absence of cues like facial expressions, tone, and gestures [6, 3]. Consequently, in this study, we introduce the concept of hand-crafted linguistic features. This involves selecting pertinent features that encapsulate the semantic and syntactic cues inherent in sarcasm. These features encompass elements like negation, contrast, and exaggeration, among others. The incorporation of hand-crafted features empowers the sarcasm detection system to grasp the sophisticated and delicate nature of sarcasm in textual content. This approach effectively captures the nuances of language that are often difficult to capture through purely data-driven methods. To implement this approach, we utilize the extracted features in an Ensemble Classification Model and a Dense Network for detecting sarcasm within the text. The key contribution of this paper can be succinctly summarized as follows:

- Deriving 13 unique handcrafted linguistic features aimed at capturing the semantic and syntactic cues associated with sarcasm present in the textual data.
- Utilizing a variety of machine learning models such as Support Vector Machine, Random Forest, Logistic Regression, K-Nearest Neighbor, Gaussian Nave Bayes, Gradient Boosting, and Decision Tree classifiers on the extracted features to identify instances of sarcasm within text.
- Proposing an Ensemble Learning and Dense Network-based classification framework to amplify the effectiveness of the extracted features in detecting sarcasm within textual data.

The remaining content of this paper is structured as follows: Section 2 provides a concise overview of the relevant literature concerning sarcasm detection. Section 3 delves deeply into the proposed method, which involves extracting handcrafted features and applying an ensemble learning-based classification scheme to detect sarcasm in text. The results stemming from this methodology are delineated in Section 4. Ultimately, Section 5 encapsulates the concluding remarks and outlines future prospects for this study.

2. Related Works

Sarcasm detection within the realm of natural language processing is a relatively new research field that has gained prominence in recent times. Prevailing methodologies for sarcasm detection have predominantly centered around identifying lexical and pragmatic attributes within a sentence to recognize sarcasm. Various approaches have been explored in the literature, showcasing encouraging strategies to unveil distinctive cues for sarcasm identification. Buschmeier *et al.* [7] employed a combination of 29 features encompassing elements like emotions, punctuation, interjection, and hyperbole. They applied logistic regression model on extracted features and achieved 0.75 F1-score on Amazon reviews dataset consisting of 1,254 review instances. Liu *et al.* [8] suggested an ensemble learning scheme that uses semantic imbalance rate, semantic features, lexical attributes, and punctuation symbol-based features for sarcasm detection. They utilized an English dataset to showcase the performance of their work and demonstrate 0.85 of AUC score. In their research, Shrawankar *et al.* [9] applied parsing-based lexicon features and employed Support Vector Machine (SVM) on a dataset they compiled, consisting of 41,227 individual employee feedback entries from the “100 Best Companies to Work for in America.” Their efforts resulted in an F1-score of 0.75. In their work [10], Khodak *et al.* employed features derived from bag-of-bigrams, sentence embeddings, and bag of words methodologies on a self-curated extensive sarcasm dataset. This study emphasized the significance of contextual information in achieving a higher level of accuracy in identifying sarcasm.

In recent times, the utilization of deep neural networks has gained traction in the field of sarcasm detection, owing to their inherent capability to autonomously learn underlying features. This intrinsic aptitude substantially bolsters the effectiveness of neural networks, facilitating the accurate identification of instances of sarcasm. Ghosh *et al.* [11] introduced a sarcasm detection system grounded in deep neural networks. Their architecture integrates a Long Short-Term Memory (LSTM) module beneath a Convolutional Neural Network (CNN) framework. On a similar note, Hazarika *et al.* [12] proposed a fusion-based methodology. By extracting contextual insights from discussion thread discourse sections and embedding user attributes, they effectively encode stylometric and personality characteristics. Their model demonstrated promising outcomes when applied to a substantial Reddit dataset. Context-awareness and GloVe embeddings played a crucial role in the work of Kumar *et al.* [13], who employed a bi-directional Long-Short-Term Memory (Bi-LSTM) model. Their approach achieved noteworthy accuracy scores of 0.86 and 0.83 on Reddit and Twitter corpus, respectively. Hiai *et al.* [14] devised a bi-directional LSTM (Bi-LSTM) model tailored to focus on relation information within role expression pairs like "boss and staff." By incorporating this relational context, they attained an F1-score of 0.80 for their sarcasm detection method. Jain *et al.* [15] curated a dataset comprising 30,000 code-mix English-Hindi tweets containing both non-sarcastic and sarcastic posts. Their hybrid model, which amalgamated attention CNN with Bi-LSTM, garnered an impressive F1-score of 0.89. In a similar vein, Ren *et al.* [16] proposed a hybrid model founded on a multilevel memory network framework. By separately applying LSTM and CNN to textual data, they effectively captured contrasting elements across various memory levels within sentences for sarcasm detection. Their findings, based on the Twitter datasets, Internet Argument Corpus (IAC-1) and IAC-2, yielded F1-scores of 0.87, 0.67, and 0.74, respectively. Recently, Pandey *et al.* [17] put forward another hybrid model suggestion that incorporates the attention mechanism of LSTM and handcrafted features to further enhance the accuracy of sarcasm detection. Moreover, a comprehensive overview of several potential approaches for identifying sarcasm is presented in Table 1.

Table 1: Brief description of various Sarcasm detection schemes proposed by the researchers in recent years.

Author	Year	Model	Features	Dataset	Performance
Buschmeier *et al.* [7]	2015	Liner SVM	Sentiment Word Count + Star Rating + BoW	Filatova (2012)	F1-Score = 0.713
		Random Forest			F1-Score = 0.482
Liu *et al.* [8]	2015	Ensemble Learning	Chinese and English Lingustic Attributes	Amazon Dataset	AUC = 0.85
				Twitter Dataset	AUC = 0.84
Ghosh *et al.* [11]	2016	LSTM + CNN + SVM	Recursive SVM Features	Own Dataset	F1-Score = 0.663
Khodak *et al.* [10]	2018	Manual Evalution	Sentence Embedding + BoW + Bag of Bigrams	Reddit Corpus with Self Annotation	F1-Score = 0.71
					F1-Score = 0.758
					F1-Score = 0.732
Hazarika *et al.* [12]	2018	Hybrid Approach + CNN	Contextual Information	SARC Dataset	F1-Score = 0.86
Shrawankar *et al.* [9]	2019	SVM	Parsing-based Lexicon Generation	Own Dataset	F1-Score = 0.73
Kumar *et al.* [13]	2019	Ensemble Voting	Term Frequency-Inverse Document Frequency (TF-IDF)	Reddit Dataset	F1-Score = 0.829
				Twitter Dataset	F1-Score = 0.863
Hiai *et al.* [14]	2019	RNN	Pair based Relation Information	Own Dataset	F1-Score = 0.802
Jain *et al.* [15]	2020	CNN + BiLSTM	Subjective Lexicon Hindi-SentiWordNet	Own Dataset	F1-Score = 0.694
Ren *et al.* [16]	2020	Memory Network	Word Embedding	IAC1 Datset	F1-Score = 0.676
				IAC1 Datset	F1-Score = 0.742
Pandey *et al.* [17]	2021	Hybrid Model	Word Embedding + Manual Features	Riloff Tweet	F1-Score = 0.99
				News Headlines	F1-Score = 0.88
Wen *et al.* [18]	2022	SAAG Model	Auxiliary Information	Chinese News	F1-Score = 0.732
Zhang *et al.* [19]	2023	BERT	Word Embedding	Own Dataset	F1-Score = 0.714
Dwivedy *et al.* [20]	2023	LSTM-based Multimodel	Text and Image Features	Multi-modal Dataset	F1-Score = 0.690

Most of the previous research employed either a machine learning model or a deep learning model to detect instances of sarcasm. A limited number of hybrid models were suggested, each incorporating varying combinations of deep learning architectures like LSTM/Bi-LSTM, CNN, RNN, and others. Nonetheless, these models are intricate and require substantial training resources. Moreover, they operate in a data-driven fashion, which may not fully encompass the intricate and nuanced aspects of sarcasm present in text, as they lack cues such as facial expressions, tone, and gestures. This prompted us to manually extract linguistic and semantic features, aiming to capture the intricate and delicate nature of sarcasm within textual content for more efficient sarcasm detection.

3. Proposed Methodology

This section will present a thorough explanation of the proposed framework designed for the detection of sarcasm. This framework operates through a two-fold approach. Firstly, it involves the extraction of a set of 13 linguistic features from the dataset, thereby forming a feature space. Following this, two distinct methodologies are employed to identify sarcasm within the text. The initial approach entails utilizing an Ensemble Learning-based technique, while the second approach involves deploying a novel Dense Network to conduct the classification task through an analysis of the extracted features. This section provides an intricate exploration of each constituent element within the proposed framework.

3.1. Dataset Description

In this research, a readily available and newly released benchmark dataset is utilized to evaluate the models' effectiveness. This dataset is known as the News Headlines dataset, which consists of 26,709 news headlines, containing 11,734 headlines that covey sarcasm and 14,985 headlines that are convey non-sarcastic. Furthermore, to enhance the dataset's scale and mitigate data imbalances, we chose to divide longer news headlines into several shorter segments. This strategy not only increases the dataset's magnitude but also intensifies the complexity of sarcasm detection due to the shortened length of headlines. An illustration of the original and processed datasets is depicted in Fig. 1(a). Following this, for conducting the experiments, we split each dataset into training and testing sets at an 80:20 ratio. Details regarding the training and testing samples are outlined in Fig. 1(b).

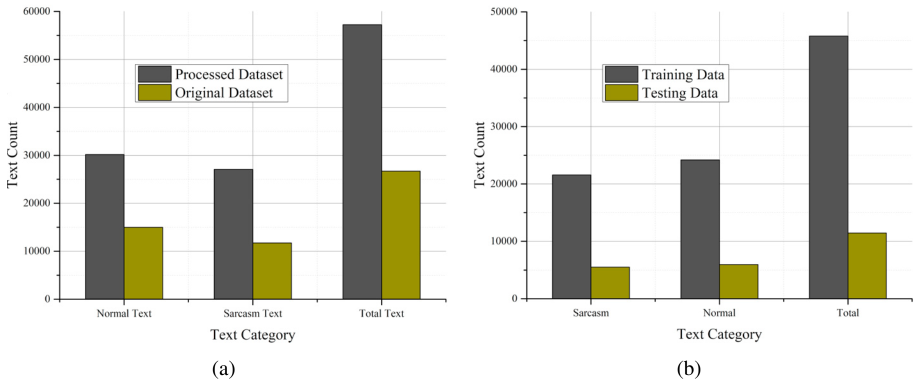

Fig. 1: Brief description of the dataset used for training and testing of the proposed framework.

3.2. Linguistic Feature Extraction

In this proposed sarcasm detection framework, we extracted 13 diverse linguistic features from the sentences. These extracted attributes encompass: Entropy, Lexical Diversity, Dale-Chall Readability Score, Flesch Reading Ease, Stop Words, Incorrect Words, Challenging Words, Lengthy Words, Two-Letter Words, Single-Letter Words, Verbs, Adjectives, and Nouns. The textual comments underwent preprocessing, involving the conversion of all text to lowercase and the exclusion of emojis and other hyperlinks. Elaborated information about these extracted attributes can be found in Table 2. In this Table 2, p represents the occurrence probability of any particular word w in the sentence. The α and β represents the the percentage of difficult words and average sentence length, respectively. Next, η is the symbolic representation of average number of syllables per word. Subsequently, all these linguistic features that are extracted will be employed to build a feature space within the pre-processed dataset. This feature space is then utilized for training and testing the classifiers.

Table 2: Brief description of the linguistic features used for sarcasm detection.

S. No.	Feature	Discription
1	Entropy	$-\sum(p \times \log_2(p))$, where p is the probability
2	Lexical Diversity	$\sum$ Unique Words/ $\sum$ Words
3	Dale-Chall Readability Score	$(0.1579 \times \alpha) + (0.0496 \times \beta)$
4	Flesch Reading Ease	$206.835 - (1.015 \times \beta) - (84.6 \times \eta)$
5	Stop Words	$\sum(S_w)$, where 'S_w' is a stop words in the text
6	Wrong Words	$\sum(W_w)$, where 'W_w' is not in a predefined list of correct words
7	Difficult Words	$\sum(w_1)$, where 'w_1' not in a predefined list of easy words
8	Lengthy Words	$\sum(w_2)$, where $\lvert w_2 \rvert > 2$
9	Two-Letter Words	$\sum(w_3)$, where $\lvert w_3 \rvert == 2$
10	Single-Letter Words	$\sum(w_4)$, where $\lvert w_4 \rvert == 1$
11	Verbs	$\sum(w_5)$, where w_5 is a Verb
12	Adjectives	$\sum(w_6)$, where w_6 is a Adjective
13	Nouns	$\sum(w_7)$, where w_7 is a noun

3.3. Ensemble Learning-based Detection

We employed a set of seven distinct classifiers: Gradient Boosting (GB), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Gaussian Nave Bayes (GNB), K-Nearest Neighbor (KNN), and Logistic Regression (LR) classifiers. These machine learning classifiers were trained using the feature space, which was divided into training and testing sets with an 80:20 ratio. The training samples were then used to train each of these classifiers. Following this, the top three performing classifiers were identified, and an Ensemble model was created using both Majority Voting and Averaging schemes. The comprehensive work-flow is depicted in Fig. 2.

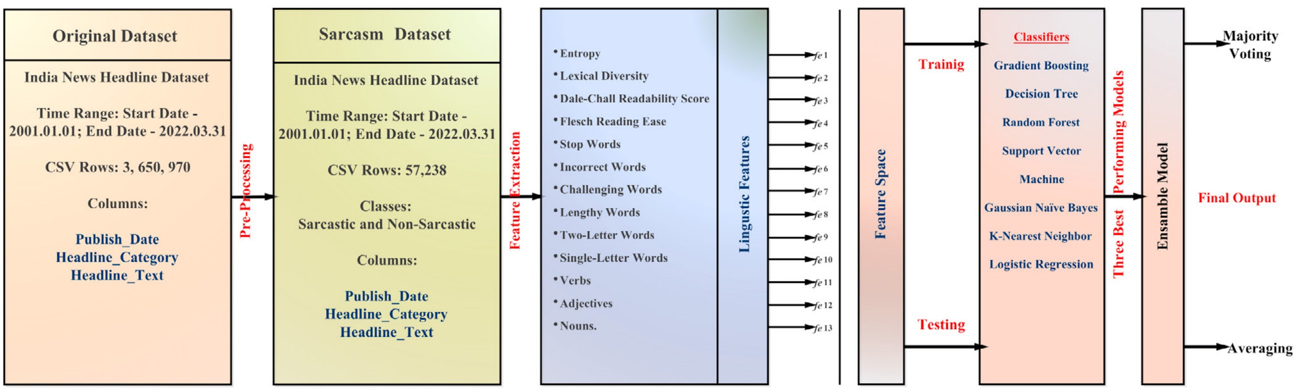

Fig. 2: Schematic flow diagram of the proposed Ensemble model-based sarcasm detection scheme.

3.4. Dense Network-based Detection

For the purpose of sarcasm detection, we formulated a sequential Dense Network, comprising an input layer in conjunction with seven additional layers. The input layer consists of 13 neurons, followed by three sets of Dense layers. Each of these dense blocks comprises one dense layer and one dropout layer, each with a dropout rate of 15%. The initial dense layer entails eight hidden units employing the ReLU activation function. The subsequent dense layer contains six hidden units also utilizing the ReLU activation function. The third dense layer accommodates four hidden units, again employing the ReLU activation function. Ultimately, the fourth dense layer encompasses two hidden units, activated by the Softmax function. The illustrative diagram portraying the proposed dense network-based approach for sarcasm detection is presented in Fig. 3.

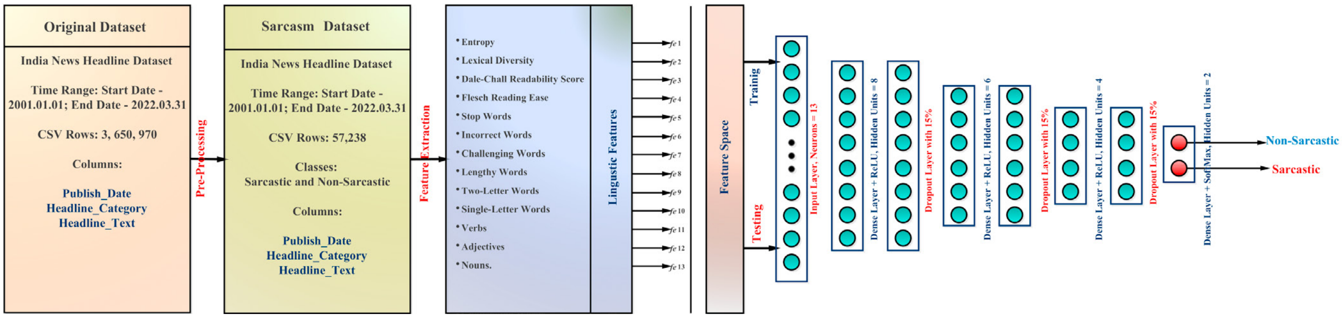

Fig. 3: Schematic flow diagram of the proposed Dense Network-based sarcasm detection scheme.

4. Results and Discussion

The proposed models have been developed in Google Colab using the TensorFlow along with the Keras framework. Extensive experiments have been conducted involving both implemented machine learning and deep learning models to validate and demonstrate their performance. The system's effectiveness has been assessed through metrics such as precision, recall, F1-score, AUC-ROC curves, and confusion matrix. The details of these matrices are as follows:

$$\mathcal{P}_r = \frac{t_p}{f_p + t_p} \tag{1}$$

$$\mathcal{R}_c = \frac{t_p}{f_n + t_p} \tag{2}$$

$$\mathcal{F}_s = \frac{2 \times \mathcal{P}_r \times \mathcal{R}_c}{\mathcal{P}_r + \mathcal{R}_c} \tag{3}$$

$$\mathcal{A}_c = \frac{t_p + t_n}{t_n + f_n + t_p + f_p} \tag{4}$$

In above context, t_p, t_n, f_p, and f_n symbolically denote true positive, true negative, false positive, and false negative, respectively. Similarly, $\mathcal{P}_r$ and $\mathcal{R}_c$ represent precision and recall, while $\mathcal{F}_s$ and $\mathcal{A}_c$ indicate the F1-score and accuracy, respectively.

Table 3 presents the results of various classifiers employing the extracted features for sarcasm detection. It's evident from Table 3 that Gradient Boosting, Decision Tree, and Random Forest stand out as the leading performers among the classifiers. Conversely, Support Vector Machine, Nave Bayes, and Logistic Regression exhibit limited effectiveness in detecting sarcasm. Notably, among all classifiers, Gradient Boosting achieves the highest accuracy, while Support Vector Machine attains the lowest accuracy. Subsequently, Table 4 showcases the performance of all seven classifiers in terms of class-wise performance.

Table 3: Performance of linguistic features along with different classifiers in sarcasm detection.

Models	$\mathcal{A}_c$	Macro Average			Weighted Average		
		$\mathcal{P}_r$	$\mathcal{R}_c$	$\mathcal{F}_s$	$\mathcal{P}_r$	$\mathcal{R}_c$	$\mathcal{F}_s$e
Gradient Boosting	0.9285	0.93	0.93	0.93	0.93	0.93	0.93
Decision Tree	0.9252	0.93	0.93	0.93	0.93	0.93	0.93
Random Forest	0.9241	0.92	0.92	0.92	0.92	0.92	0.92
Support Vector Machine	0.6750	0.67	0.67	0.67	0.67	0.67	0.67
Nave Bayes	0.6413	0.64	0.64	0.64	0.64	0.64	0.64
K-Nearest Neighbor	0.7454	0.75	0.75	0.75	0.75	0.75	0.75
Logistic Regression	0.6926	0.69	0.69	0.69	0.69	0.69	0.69

Table 4: Class-wise performance of linguistic features along with different classifiers in sarcasm detection.

Models	**Non-Sarcastic**			**Sarcastic**		
	$\mathcal{P}_r$	$\mathcal{R}_c$	$\mathcal{F}_s$e	$\mathcal{P}_r$	$\mathcal{R}_c$	$\mathcal{F}_s$
Gradient Boosting	0.93	0.93	0.93	0.93	0.92	0.93
Decision Tree	0.93	0.93	0.93	0.92	0.92	0.92
Random Forest	0.92	0.93	0.93	0.92	0.92	0.92
Support Vector Machine	0.68	0.71	0.69	0.67	0.64	0.65
Nave Bayes	0.66	0.64	0.65	0.62	0.64	0.65
K-Nearest Neighbor	0.76	0.74	0.75	0.73	0.75	0.74
Logistic Regression	0.70	0.70	0.70	0.68	0.68	0.68

Based on the results discussed above, we have identified Gradient Boosting, Decision Tree, and Random Forest as the three leading classifiers. Consequently, we employed these three classifiers to construct ensemble models using both the majority voting and average schemes. The overall performance of the ensemble learning-based approach and the dense network-based approach for sarcasm detection is summarized in Table 5. Notably, we observe that the Majority Voting-based Ensemble models consistently exhibit the most favorable outcomes across all scenarios. Conversely, the dense network model displays notably inferior detection performance. Next, Table 6, presents the class-wise performance of the ensemble and dense network-based models.

Table 5: Performance of Ensemble Learning and Dense Network in sarcasm detection.

Models	$\mathcal{A}_c$	**Macro Average**			**Weighted Average**		
		$\mathcal{P}_r$	$\mathcal{R}_c$	$\mathcal{F}_s$	$\mathcal{P}_r$	$\mathcal{R}_c$	$\mathcal{F}_s$e
Ensemble - Majority Voting	0.9375	0.94	0.94	0.94	0.94	0.94	0.94
Ensemble - Averaging	0.9351	0.94	0.94	0.94	0.94	0.94	0.94
Dense Network	0.7647	0.76	0.76	0.76	0.77	0.76	0.76

Table 6: Class-wise performance of Ensemble Learning and Dense Network in sarcasm detection.

Models	**Non-Sarcastic**			**Sarcastic**		
	$\mathcal{P}_r$	$\mathcal{R}_c$	$\mathcal{F}_s$e	$\mathcal{P}_r$	$\mathcal{R}_c$	$\mathcal{F}_s$
Ensemble - Majority Voting	0.94	0.95	0.94	0.94	0.93	0.93
Ensemble - Averaging	0.94	0.93	0.94	0.94	0.94	0.94
Dense Network	0.79	0.76	0.78	0.74	0.77	0.75

Following this, Fig. 4 illustrates the confusion matrix for each standalone classifier, ensemble model, and the dense network-based model. Additionally, Fig. 5 depicts the ROC curve for all individual classifiers, ensemble models, and the dense network-based model.

Furthermore, we conducted a comparison between our approach and five recently published sarcasm detection methods: Pandey *et al.* [17], Kumar *et al.* [13], Mandal *et al.* [21], Mehndiratta *et al.* [22], and Liu *et al.* [23]. These approaches all employed the Indian News Headline dataset for sarcasm detection and mostly employed hybrid models combining handcrafted and deep features. Comprehensive comparative outcomes are presented in Table 7. Among these methodologies, Pandey *et al.* [17] achieved the highest F1-Score of 0.88 by employing a complex hybrid attention network. In contrast, our relatively simple ensemble model attained an F1-Score of 0.9375, marking an improvement of 5.75%.

Subsequently, we conducted a comparison between our proposed approach and five other deep learning-based sarcasm detection methodologies: LSTM, Bi-LSTM, LSTM-A, DNN, and CNN. The comparisons, in terms of Precision, Recall, and F1-Score, are illustrated in Fig. 6. From this figure, it's evident that our relatively straightforward ensemble model outperforms the complex deep learning models. These numerical and visual findings validate that the set of 13 linguistic features we introduced effectively capture the semantic and syntactic cues intrinsic to sarcasm, thereby significantly improving the efficacy of sarcasm detection.

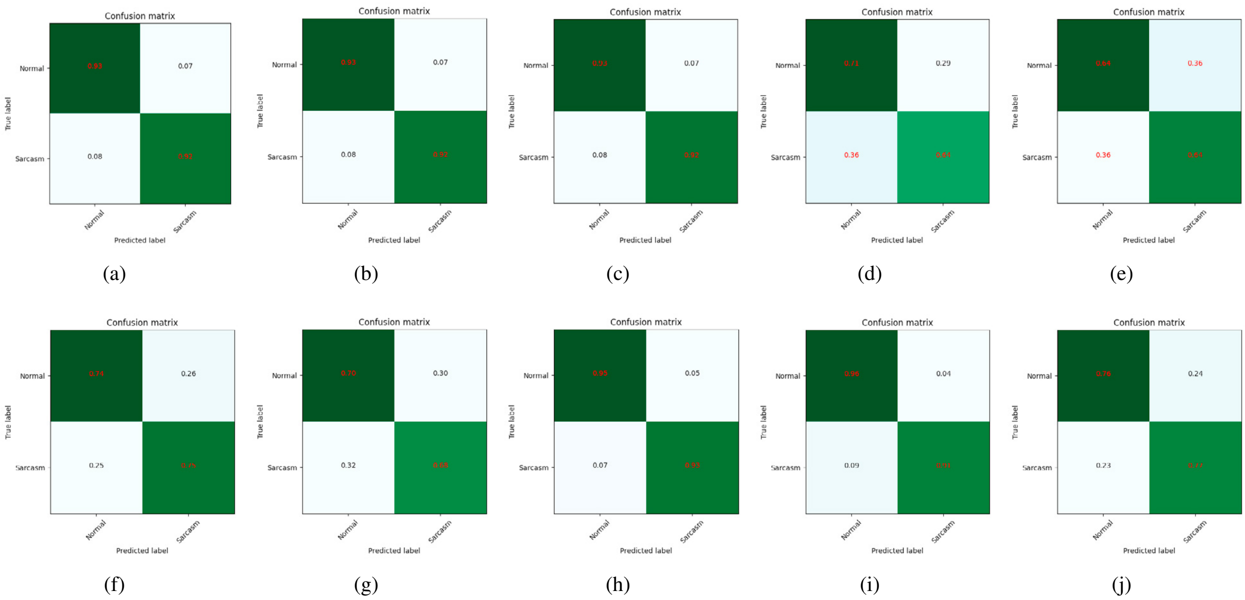

Fig. 4: Confusion matrix of (a) Gradient Boosting, (b) Decision Tree, (c) Random Forest, (d) Support Vector Machine, (e) Nave Bayes, (f) K-Nearest Neighbor, (g) Logistic Regression, (h) Ensemble + Majority Voting, (i) Ensemble + Averaging, and (i) Dense Network.

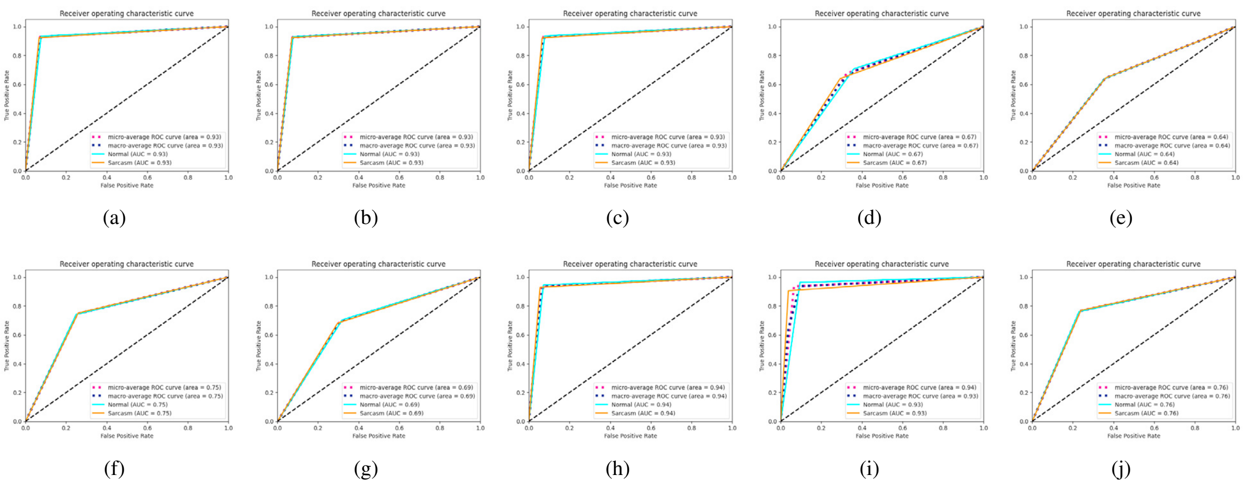

Fig. 5: ROC curve of (a) Gradient Boosting, (b) Decision Tree, (c) Random Forest, (d) Support Vector Machine, (e) Nave Bayes, (f) K-Nearest Neighbor, (g) Logistic Regression, (h) Ensemble + Majority Voting, (i) Ensemble + Averaging, and (i) Dense Network.

5. Conclusion and Future Works

Identifying sarcasm in text poses a significant challenge in the domain of natural language processing. Sarcastic statements can profoundly distort the sentiment extraction from social media text, often reversing the intended polarity of sentences. However, the application of deep learning models may fall short in capturing the intricate nuances of sarcasm due to the absence of cues like facial expressions and tone. To address this, our paper proposed a solution: the extraction of 13 linguistic features designed to effectively encapsulate sarcasm's semantic and syntactic cues. This approach substantially enhances sarcasm detection accuracy. Furthermore, we employed an ensemble learning,

Table 7: Performance comparison between proposed and other sarcasm detection schemes.

Author	Approach	Dataset	Performance/F1-Score
Pandey *et al.* [17]	LSTM + Lingustic Features	India News Headlines Dataset	$\mathcal{F}_s = 0.88$
Kumar *et al.* [13]	Bi-LSTM + GloVE		$\mathcal{F}_s = 0.83$
Mandal *et al.* [21]	LSTM + CNN + Word Embedding		$\mathcal{F}_s = 0.86$
Mehndiratta *et al.* [22]	RNN + CNN + GloVE + Word2Vec		$\mathcal{F}_s = 0.82$
Liu *et al.* [23]	DNN + POS		$\mathcal{F}_s = 0.86$
Proposed	**Lingustic Features + Ensemble Model**		$\mathcal{F}_s$ **= 0.9375**

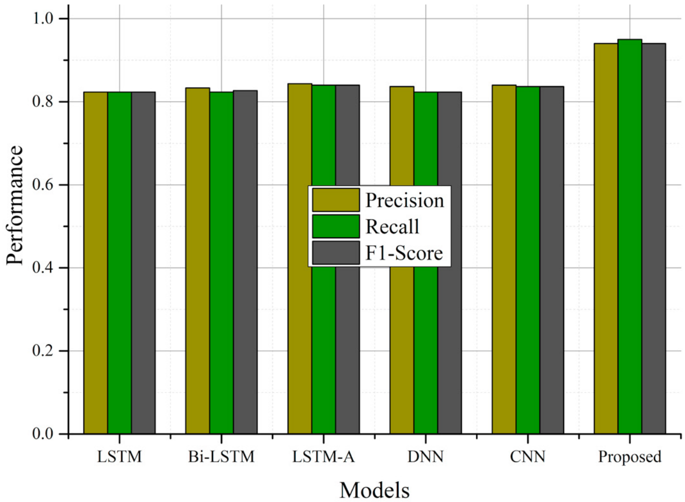

Fig. 6: Performance comparison with other deep learning-based models.

merging Gradient Boosting, Decision Tree, and Random Forest classifiers through a majority voting scheme. As a result, we achieved an improved accuracy of 0.9375 compared to existing methods. Our results underscore that the introduced features adeptly capture emotions like negation, contrast, and exaggeration. By incorporating these handcrafted features, the sarcasm detection system gained the ability to grasp the intricate and subtle nature of sarcasm in textual content, significantly elevating its detection performance.

However, in future, our goal is to incorporate these linguistic features into the attention mechanism of deep networks, boosting the effectiveness of sarcasm detection. Additionally, we plan to develop a multi-modal architecture that combines textual linguistic features with deep features extracted from associated images, further optimizing the efficiency of our sarcasm detection system.

References

[1] J. P. Singh, S. Irani, N. P. Rana, Y. K. Dwivedi, S. Saumya, P. K. Roy, Predicting the helpfulness of online consumer reviews, Journal of Business Research 70 (2017) 346–355.

[2] D. K. Sharma, B. Singh, S. Agarwal, N. Pachauri, A. A. Alhussan, H. A. Abdallah, Sarcasm detection over social media platforms using hybrid ensemble model with fuzzy logic, Electronics 12 (4) (2023) 937.

[3] S. Xiong, G. Zhang, V. Batra, L. Xi, L. Shi, L. Liu, Trimoon: Two-round inconsistency-based multi-modal fusion network for fake news detection, Information fusion 93 (2023) 150–158.

[4] B.-N. Xu, Y.-B. Cao, J. Meng, Z.-J. He, L. Wang, A transformer based multimodal fine-fusion model for false information detection, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2023, pp. 271–277.

[5] S. Suryavardan, S. Mishra, P. Patwa, M. Chakraborty, A. Rani, A. Reganti, A. Chadha, A. Das, A. Sheth, M. Chinnakotla, et al., Factify 2: A multimodal fake news and satire news dataset, arXiv preprint arXiv:2304.03897.

[6] P. K. Roy, A. K. Tripathy, T. K. Das, X.-Z. Gao, A framework for hate speech detection using deep convolutional neural network, IEEE Access 8 (2020) 204951–204962.

[7] K. Buschmeier, P. Cimiano, R. Klinger, An impact analysis of features in a classification approach to irony detection in product reviews, in: Proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis, 2014, pp. 42–49.

[8] P. Liu, W. Chen, G. Ou, T. Wang, D. Yang, K. Lei, Sarcasm detection in social media based on imbalanced classification, in: Web-Age Information Management: 15th International Conference, WAIM 2014, Macau, China, June 16-18, 2014. Proceedings 15, Springer, 2014, pp. 459–471.

[9] U. Shrawankar, C. Chandankhede, Sarcasm detection for workplace stress management, International Journal of Synthetic Emotions (IJSE) 10 (2) (2019) 1–17.

[10] M. Khodak, N. Saunshi, K. Vodrahalli, A large self-annotated corpus for sarcasm. arxiv, arXiv preprint arXiv:1704.05579.

[11] A. Ghosh, T. Veale, Fracking sarcasm using neural network, in: Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis, 2016, pp. 161–169.

[12] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, R. Mihalcea, Cascade: Contextual sarcasm detection in online discussion forums, arXiv preprint arXiv:1805.06413.

[13] A. Kumar, G. Garg, Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets, Journal of ambient intelligence and humanized computing (2019) 1–16.

[14] S. Hiai, K. Shimada, Sarcasm detection using rnn with relation vector, International Journal of Data Warehousing and Mining (IJDWM) 15 (4) (2019) 66–78.

[15] D. Jain, A. Kumar, G. Garg, Sarcasm detection in mash-up language using soft-attention based bi-directional lstm and feature-rich cnn, Applied Soft Computing 91 (2020) 106198.

[16] L. Ren, B. Xu, H. Lin, X. Liu, L. Yang, Sarcasm detection with sentiment semantics enhanced multi-level memory network, Neurocomputing 401 (2020) 320–326.

[17] R. Pandey, A. Kumar, J. P. Singh, S. Tripathi, Hybrid attention-based long short-term memory network for sarcasm identification, Applied Soft Computing 106 (2021) 107348.

[18] Z. Wen, L. Gui, Q. Wang, M. Guo, X. Yu, J. Du, R. Xu, Sememe knowledge and auxiliary information enhanced approach for sarcasm detection, Information Processing & Management 59 (3) (2022) 102883.

[19] Y. Zhang, D. Ma, P. Tiwari, C. Zhang, M. Masud, M. Shorfuzzaman, D. Song, Stance-level sarcasm detection with bert and stance-centered graph attention networks, ACM Transactions on Internet Technology 23 (2) (2023) 1–21.

[20] V. Dwivedy, P. K. Roy, Deep feature fusion for hate speech detection: a transfer learning approach, Multimedia Tools and Applications (2023) 1–23.

[21] P. K. Mandal, R. Mahto, Deep cnn-lstm with word embeddings for news headline sarcasm detection, in: 16th International Conference on Information Technology-New Generations (ITNG 2019), Springer, 2019, pp. 495–498.

[22] P. Mehndiratta, D. Soni, Identification of sarcasm using word embeddings and hyperparameters tuning, Journal of Discrete Mathematical Sciences and Cryptography 22 (4) (2019) 465–489.

[23] L. Liu, J. L. Priestley, Y. Zhou, H. E. Ray, M. Han, A2text-net: A novel deep neural network for sarcasm detection, in: 2019 IEEE first international conference on cognitive machine intelligence (CogMI), IEEE, 2019, pp. 118–126.