

Telugu Sarcasm Detection: Performance Evaluation of Machine Learning Approaches

Bhargavi Vemala , M. Humera Khanam 

Department of Computer Science and Engineering, Sri Venkateswara University, Tirupati 517502, India

Corresponding Author Email: humera.svec@gmail.com

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.120427>

ABSTRACT

Received: 19 January 2025

Revised: 4 March 2025

Accepted: 10 March 2025

Available online: 30 April 2025

Keywords:

sarcasm detection, Telugu language, natural language processing, machine learning, term frequency-inverse document frequency

Sarcasm detection in Telugu, a Dravidian language widely used across the Indian regions of Andhra Pradesh and Telangana, presents distinct challenges due to its rich morphology and complex syntactic structures. The word-building nature of the language, where multiple morphemes combine to form words, complicates sarcasm identification because subtle changes in word forms and structure can convey ironic meanings. Additionally, the intricate syntactic patterns of Telugu further complicate the task of recognizing sarcastic expressions. Despite the extensive research on sarcasm detection in widely spoken languages like English, there is a noticeable lack of resources and research dedicated to sarcasm detection for Telugu. This research explores the application of diverse machine learning techniques for the effective detection of sarcasm in Telugu text (TSD-PEMLA). The dataset collected from social media platforms, with labels for sarcasm and sentiment polarity. The proposed TSD-PEMLA method using the Voting Classifier with CountVectorizer and TF-IDF, demonstrated superior performance compared to other models. Further processing of the sarcasm dataset, a dual-label classification approach was implemented, combining sarcasm detection with sentiment analysis, using AdaBoost and Voting Classifiers. The findings emphasize the effectiveness of the suggested approach for identifying sarcasm in Telugu text while showcasing the ability of machine learning methods.

1. INTRODUCTION

In natural language processing (NLP), sarcasm detection is a crucial but difficult problem that entails spotting tiny clues that imply the opposite of what is literally conveyed [1]. Although irony or mocking tones are commonly used to identify sarcasm, computationally recognizing it can be challenging, particularly in languages with intricate linguistic structures [2]. The complex morphological traits and syntactic structure of Telugu, a Dravidian language spoken mostly in Andhra Pradesh and Telangana, set it apart from more well studied languages like English [3-5]. Because of these linguistic features, sarcasm detection in Telugu is a very difficult task that calls for advanced techniques to capture the subtleties of the language [6].

Telugu's agglutinative nature, in which a single word can include many morphemes that express separate grammatical meanings, makes it difficult to discern sarcasm in the language [7-10]. Because sarcasm may be conveyed by minute changes in word forms, suffixes, or even word order, this morphological diversity can make text analysis tasks more difficult [11-15]. Furthermore, Telugu's intricate syntactic patterns provide a variety of phrase formations, which may make it more difficult to spot sarcasm [10-12]. The challenge of developing trustworthy sarcasm detection algorithms for Telugu text is further increased by the existence of dialectal variances and colloquial idioms [16-22].

This paper investigates the use of several machine learning

techniques for Telugu sarcasm detection to address these issues. To construct and assess our sarcasm detection models, we used a variety of methods, such as Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), Multilayer Perceptron (MLP), AdaBoost, CatBoost, XGBoost, and a Voting Classifier. From ensemble approaches that improve model performance by integrating many classifiers to tree-based methods that may capture non-linear correlations, each of these algorithms has certain advantages when processing text data. Since Telugu is a morphologically rich and syntactically complicated language, the variety of approaches aids in determining which machine learning algorithms are most suited for sarcasm detection.

The study demonstrates the capability of ensemble learning methods for sarcasm detection alongside providing an understanding of the efficiency of specific algorithms. Our goal is to determine the best methods for managing Telugu's distinct linguistic characteristics by comparing the outcomes of several models. Furthermore, this study contributes to the development of NLP methods that consider a broad range of linguistic structures and cultural settings, highlighting the wider importance of sarcasm identification in underrepresented languages. The main key contributions are:

- It assesses various models like DT, SVM, KNN, Logistic Regression, AdaBoost, CatBoost, XGBoost, and Voting Classifier for sarcasm detection in Telugu, offering a comprehensive analysis.

- It examines the performance of these techniques to address Telugu's rich morphology and complex syntax, identifying effective approaches for NLP tasks.

- Develops a sarcasm-labeled Telugu dataset from social media, filling the gap of annotated data and aiding future sentiment analysis research.

- Employs measures including accuracy, precision, recall, and F1-score for evaluating model performance to ensure reliable results and valuable insights for selecting models.

- Demonstrates the potential of applying machine learning to regional languages, expanding NLP capabilities in underrepresented linguistic communities.

- Offers a significant dataset and knowledge to support the advancement of tools for sarcasm detection and sentiment analysis in Telugu as well as other under-resourced languages.

2. RELATED WORK

This section examines prior studies on the implementation of machine learning approaches for detecting sarcasm. Some recent studies in this field are highlighted below.

In 2023, Kumar et al. [16] presented the pioneering work of creating two separate datasets for sarcasm detection in Telugu and Tamil by collecting tweets from Twitter. Sarcasm identification in Tamil and Telugu tweets was carried out using multiple approaches, incorporating machine learning algorithms like DT, NB, LR, SVM, and RF, as well as sophisticated neural network models such as CNN and LSTM. However, it has low precision value.

In 2021, Eket al. [17] developed a context-driven feature approach for sarcasm recognition, incorporating deep learning models, the BERT framework, and classical machine learning methods. Two widely recognized datasets, namely Twitter and Internet Argument Corpus version two (IAC-v2), were applied for classification with three separate learning models. The initial model employed an embedding-based technique using deep learning with Bi-LSTM, a type of RNN, combined with Global Vector (GloVe) for word embedding and contextual understanding. The second model employed the Transformer framework, integrating the pre-trained Bidirectional Encoder Representations from Transformers (BERT). The third model utilized feature fusion by combining BERT characteristics, sentiment-based attributes, syntactic information, and GloVe embeddings with traditional machine learning approaches. However, it has F1-score value.

In 2022, Goel et al. [18] examined the use of neural approaches, including LSTM, GRU, and Baseline CNN, within a combined model for detecting sarcasm online. To boost the model's performance, a dataset was developed incorporating different pre-trained word embedding models like fastText, Word2Vec, and GloVe, with a comparative analysis of their accuracy. The primary aim was to classify the writer's sentiment as positive, negative, sarcastic, or non-sarcastic, that ensures the message was effectively conveyed to the target audience. The incorporation of the ensemble model led to better consistency, precision, and forecasting ability of the system. However, it has high error rate.

In 2023, Ratnavel et al. [19] developed a model based on transformer architecture to detect sarcasm in Tamil code-mixed text. The architecture included two uniquely crafted layers such as an encoder and an embedding layer. It employed a multi-head self-attention approach, feed-forward networks, and featured both normalization and dropout layers. The

introduced model surpassed existing top models in sarcasm detection, attaining a weighted score of 0.77. This strategy successfully addressed the complexities of Tamil code-mixed text. How it has high error rate.

In 2022, Vinoth and Prabhavathy [20] developed a technique utilizing machine learning for identifying sarcasm in content shared on social media platforms. The IMLB-SDC approach involved a series of processes, including data cleaning, engineering of features, selection of relevant features (FS), categorization, and fine-tuning of model parameters. TF-IDF was utilized for feature extraction. To identify important features, the chi-square and information gain methods were applied. The classification was carried out using SVM, with the penalty factor adjusted via the PSO method. However, it has low sensitivity value.

In 2023, Kumar and Garg [21] explored the impact of context in sentiment analysis. The study employed Twitter data from the SemEval 2015 Task 11 benchmark alongside approximately 20,000 posts from Reddit, categorizing the data as sarcastic or non-sarcastic with the application of three predictive learning models. The first approach employed standard TF-IDF for feature weighting and was trained with three classifiers such as Multinomial NB, Gradient Boosting, and RF. The ultimate prediction was obtained via Ensemble Voting. The second model merged semantic (sentiment) and pragmatic (punctuation) attributes with the top-200 TF-IDF features, which were assessed using five foundational classifiers such as DT, SVM, RF, KNN, and MLP. The final approach employed deep learning techniques, specifically LSTM and Bi-directional LSTM, alongside GloVe to generate word embeddings and understand context. The Bi-directional LSTM model delivered the best results, reaching an accuracy of 86.32% on Twitter data and 82.91% on Reddit data. However, it has low accuracy value.

In 2024, Gedela et al. [22] developed a voting-driven ensemble technique for detecting sarcasm, leveraging deep learning models. The technique applied BERT to produce word embeddings that reflect context, which were subsequently fed into an ensemble of four deep learning models. The ensemble consisted of CNN, Bi-LSTM, and various parallel and sequential model combinations. The extracted features from each model were evaluated using four machine learning classifiers such as SVM, Least Squares SVM, multinomial NB, and RF, with the application of a Sigmoid activation function for classification. The most effective classifier was selected for every model, and a majority voting strategy was employed to aggregate the results of all four models for classifying the texts as either sarcastic or non-sarcastic. The method was assessed using two standard datasets, consisting of news headlines and a Reddit corpus that was self-annotated. However, it has low accuracy value.

Although sarcasm detection in English has been thoroughly investigated, there is a lack of knowledge regarding low-resource languages such as Telugu. While sentiment analysis has been utilized for identifying sarcasm in English, there has been limited research on its efficacy in detecting sarcasm in Telugu. Telugu has specific challenges for the identification of sarcasm.

Among these surveys, extensive study has been conducted on English with superior accuracy, as well as code-mixed data involving Telugu, Tamil, and English. My primary method involves focusing on certain languages and identifying appropriate models to address challenges and enhancements. The Telugu dataset is accessible; however, it lacks sentences

labelled for sarcasm. Training with this kind of dataset will get lack of precision and recall values. Despite employing back translations with English, there exists an issue of syntactic organization in low-resourced languages.

2.1 Problem statement and motivation

Sarcasm detection in Telugu, a Dravidian language characterized by rich morphology and complex syntax, presents significant challenges within the field of NLP. Telugu’s word-building nature, where a single word embody multiple morphemes, complicates the task of identifying sarcasm. The language’s diverse word forms, suffixes, and syntactic structures further contribute to difficulties in sarcasm detection, as minor modifications in structure convey ironic meanings. Additionally, the lack of sufficiently labeled datasets for sarcasm recognition in Telugu restricts the development of robust models. This research focuses on evaluating various machine learning algorithms for sarcasm detection in Telugu text, aiming to assess their strengths and weaknesses when handling the language’s unique linguistic features [16-22]. Recognizing sarcasm is crucial for enhancing the precision of sentiment analysis, particularly in informal communication and social media contexts. While sarcasm detection has been extensively studied in languages like English, the complexities inherent in Telugu such as its word-building morphology and intricate syntactic structure necessitate specialized methods. Telugu presents a unique challenge and an opportunity for contributing to the broader NLP field, especially for underrepresented languages. The evaluation of several machine learning algorithms in this study aims to address the specific challenges posed by Telugu’s linguistic features, with the goal of enhancing sentiment analysis and content moderation in regional languages. By expanding the reach of NLP technologies to include Telugu, this research promotes the development of more inclusive AI systems capable of interpreting sentiment in low-resource languages. This research explores the use of machine learning techniques to detect sarcasm in Telugu language text (TSD-PEMLA). The models developed and assessed encompass DT, RF, SVM, KNN, LR, MLP, AdaBoost, CatBoost, XGBoost, and a Voting Classifier.

3. PROPOSED METHODOLOGY

The suggested technique for identifying sarcasm in Telugu text consists of multiple steps, including gathering data, preprocessing, extracting features, training models, and assessing performance.

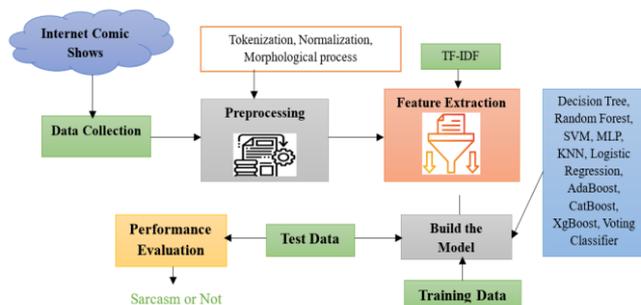


Figure 1. Structure of sarcasm detection model

Machine learning algorithms are employed to ensure precise detection of sarcasm. The diagram in Figure 1 demonstrates the structure of this framework.

3.1 Data collection

The real time data for this study was curated from social media platforms, where Telugu speakers commonly express opinions, share thoughts, and engage in conversations. A total of 6,200 sentences were collected, representing a diverse range of topics to ensure the dataset captured various contexts in which sarcasm occurs. The data selection process aimed to include sentences that exhibited both explicit and subtle forms of sarcasm, as well as non-sarcastic text, to provide a balanced representation for training and evaluating sarcasm detection models. To prepare the dataset for sarcasm detection, manual annotation was performed by native Telugu speakers who are familiar with the nuances of the language, including its dialectal variations. Each sentence was reviewed to determine whether it conveyed sarcasm, with labels assigned as "yes" for sarcastic content and "no" for non-sarcastic content. This manual annotation process ensured high-quality labeling, capturing the implicit meaning and contextual clues often associated with sarcasm. In addition to sarcasm labeling was integrated into the dataset using SentiWordNet to categorize each sentence's sentiment polarity. Annotators assigned labels of positive, negative, or neutral to reflect the underlying sentiment expressed in the text. This dual annotation approach facilitated a deeper analysis, allowing the models not only to detect sarcasm but also to understand the sentiment orientation in sarcastic and non-sarcastic content. The resulting annotated dataset serves as a valuable resource for developing machine learning models that can simultaneously perform sarcasm detection in Telugu, enhancing the advancement of NLP in languages with scarce resources.

3.2 Preprocessing

The preprocessing process encompassed multiple tasks aimed at cleaning and transforming the raw text data into a format compatible with machine learning models [23]. First, tokenization was applied, breaking down each sentence into individual words or tokens. This step enabled more granular text analysis and facilitated the extraction of meaningful features. Tokenization breaks down each sentence into individual words or tokens. If a sentence S is represented as a sequence of characters, it can be expressed in Eq. (1):

$$S = \{w_1, w_2, \dots, w_n\} \tag{1}$$

where, w_i represents the individual tokens (words) in the sentence. This step enables more granular analysis by treating each token as a feature for the model.

Following tokenization, normalization was performed to convert text into a consistent format by addressing variations in word forms, correcting spelling errors, and standardizing the use of special characters and punctuation. Let w_i be a token, the normalized form $N(w_i)$ can be defined as in Eq. (2):

$$N(w_i) = \text{standardize}(w_i) \tag{2}$$

where, the standardization function converts the token into a uniform representation, such as lowercase letters and corrected spellings.

The next step was stop-word removal, which involved filtering out common Telugu words such as conjunctions and prepositions that do not contribute significantly to the meaning of the text. This process helped reduce noise in the data and focused the analysis on more informative words that could aid in sarcasm detection. If the set of stop-words is represented as SW , then the cleaned token list T is obtained by in Eq. (3):

$$T = \{w_i \setminus w_i \notin SW\} \quad (3)$$

A crucial aspect of preprocessing for Telugu text was morphological processing. Given Telugu's rich morphology, where words can have multiple inflectional forms, morphological analysis was applied to standardize these variations. Morphological analysis involves techniques like stemming and lemmatization to convert inflected forms to a base form R . For a word w_i , the root form Rw_i is obtained as in Eq. (4):

$$R(w_i) = stem(w_i) \text{ or } lemma(w_i) \quad (4)$$

For example, different inflected forms of a verb or noun were converted to a common root, allowing the models to treat them as a single feature. This standardization improved the consistency of feature extraction and enabled the machine learning algorithms to better recognize patterns in the text, thereby enhancing sarcasm detection performance.

3.3 Feature extraction

Extracting features plays a vital role in converting the preprocessed text into a numerical representation that is compatible with machine learning techniques. In this research, we used a combination of n-gram models and TF-IDF weighting to extract meaningful features from the preprocessed text, along with additional features such as sentiment scores derived from SentiWordNet to enhance sarcasm detection and sentiment analysis [24].

N-grams capture sequences of words to provide context. We used Single words (unigrams), word pairs (bigrams), and three-word combinations (trigrams) were utilized to characterize text features. These n-grams help in identifying patterns in the data, as sarcastic expressions often involve specific word combinations. Character n-grams were employed to address Out-of-Vocabulary (OoV) terms. Combining all the data to perform TF-IDF, followed by subsequent splitting. The unknown term from the test data cannot pose any issues. TF-IDF was used to quantify the importance of terms. TF indicates how frequently a term occurs in a document, whereas IDF measures the relative rarity or prevalence of the term across all documents. The TF-IDF score for a term t in a document d is computed as shown in Eq. (5):

$$TF - IDF(t, d) = TF(t, d) \times IDF(t, d) \quad (5)$$

To enhance sarcasm detection, sentiment features derived from SentiWordNet were included. Words were assigned positive, negative, and neutral scores, which were aggregated to form overall sentiment features for each sentence. Combining n-gram features, TF-IDF scores, and sentiment information provided a rich set of features for the models, enabling better sarcasm detection.

3.4 Sarcasm detection using ensemble learning

In this study, we implemented a diverse array of machine learning algorithms to build and evaluate models for sarcasm detection in Telugu text. Each algorithm was chosen based on its unique strengths and capabilities, facilitating a comprehensive analysis of their performance in identifying sarcastic content.

3.4.1 DT

DTs are clear models that progressively partition the dataset according to feature values, with each terminal node providing a classification outcome. The criterion for splitting is commonly derived from metrics like Gini coefficient or entropy. The formula for Gini impurity G for a split is given in Eq. (6):

$$G = 1 - \sum_{i=1}^c p_i^2 \quad (6)$$

In this case, p_i refers to the proportion of class i in the dataset, while C stands for the total number of classes. Although DT are straightforward to interpret and visualize, they may suffer from overfitting, especially when handling complex datasets.

3.4.2 RF

RF enhances the DT approach by incorporating randomness in feature selection during splits. This improves robustness and reduces overfitting, which is particularly useful in high-dimensional datasets. The technique randomly picks a group of features F for every split, which helps to improve generalization.

3.4.3 SVM

SVMs are robust classification models that determine the ideal hyperplane to separate distinct classes within a high-dimensional space. The hyperplane can be defined as in Eq. (7):

$$w \cdot x + b = 0 \quad (7)$$

In this context, w is the vector of coefficients, x is the vector of input features, and b represents the bias value. SVMs are effective for sarcasm detection due to their ability to handle non-linear boundaries using kernel functions, such as for polynomial kernels is defined in Eq. (8).

$$K(x_i, x_j) = (\alpha x_i \cdot x_j + r)^d \quad (8)$$

3.4.4 KNN

KNN categorizes instances by determining the predominant class among the k nearest points in the feature space. The decision rule can be expressed in Eq. (9):

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^k I(y_i = y) \quad (9)$$

where, I is an indicator function, y_i is the class of the i^{th} neighbor, and Y is the set of classes. This method is useful for capturing local patterns, making it suitable for sarcasm detection, where context is crucial.

3.4.5 LR

LR predicts the probability of a binary result based on input variables. The probability $P(y = 1 | x)$ can be modeled in Eq. (10):

$$P(y = 1 | x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n)}} \quad (10)$$

where, θ represents the model coefficients [25]. This model is straightforward and interpretable, serving as a baseline for evaluating the impact of features on sarcasm detection.

3.4.6 MLP

MLPs are made up of several layers of interconnected neurons, that enables the capture of intricate patterns within the data. The output y for a neuron can be represented in Eq. (11):

$$y = f\left(\sum_{i=1}^n w_i x_i + b\right) \quad (11)$$

where, f is an activation function, w is weight, x are input, and b is the bias term. MLPs are well-suited for tasks requiring modeling of non-linear relationships.

3.4.7 AdaBoost

AdaBoost is a collective approach that integrates the predictions of multiple weak classifiers to generate a powerful classifier. The final prediction can be formulated in Eq. (12):

$$\hat{y} = \text{sign}\left(\sum_{m=1}^M \delta_m h_m(x)\right) \quad (12)$$

where, δ_m is the weight for the m^{th} weak classifier $h_m(x)$. By focusing on misclassified instances, AdaBoost effectively enhances model performance, especially in challenging tasks like sarcasm detection.

3.4.8 CatBoost

CatBoost is a gradient boosting algorithm that efficiently handles categorical features. It improves accuracy while requiring less tuning compared to other boosting methods, making it suitable for processing textual data.

3.4.9 XGBoost

XGBoost is an efficient implementation of gradient boosting that incorporates regularization, represented in Eq. (13):

$$\hat{y}_{t+1} = \hat{y}_t + \vartheta \cdot \text{argmax}_f \left(\sum_{i=1}^N L(y_i, f(x_i)) + \varphi(f) \right) \quad (13)$$

where, L is the loss function and $\varphi(f)$ is the regularization term. This allows for robust performance in classification tasks and is beneficial in detecting sarcasm.

3.4.10 Voting Classifier

The Voting Classifier consolidates predictions from various base models to increase overall accuracy. The final decision is based on the majority vote from the classifiers, as represented in Eq. (14):

$$\hat{y} = \text{argmax}_y \left(\sum_{i=1}^N \hat{y}_i = y \right) \quad (14)$$

where, I is an indicator function for the predicted \hat{y}_i . This combined strategy takes advantage of multiple algorithms to strengthen the accuracy of sarcasm detection.

To evaluate inter-annotator agreement (IAA), two coefficients were employed: Cohen's Kappa [26] and Fleiss Kappa [27]. Cohen's Kappa is utilized for two annotators, whereas Fleiss Kappa is suitable for more than two annotators. Cohen's Kappa coefficient, applicable due to the presence of two labels (Yes/No), is computed using the formula presented in Eq. (15). The IAA for this study is 0.83, signifying a substantial consensus among the annotators.

$$k = \frac{P_0 - P_e}{1 - P_e} \quad (15)$$

Each algorithm was trained and evaluated on the curated dataset of 6,200 annotated sentences. This comprehensive approach highlighted the strengths and weaknesses of individual algorithms and facilitated comparisons between their performances. Through this implementation of various machine learning techniques, our study aimed to establish effective methods for detecting sarcasm in Telugu text, enhancing progress in NLP for languages with few resources. By leveraging a diverse set of algorithms, we sought to enhance the understanding and application of sarcasm detection in complex linguistic environments.

4. EXPERIMENTAL RESULTS

Table 1. Example Telugu sarcasm labelled dataset

Telugu Sentence	English Meaning	Label
నీ కృషి బుడిదలో పోసిన పన్నీరు లాగా ఫలితం లేదు.	Your hard work is like cheese poured into the mud.	Sarcastic
గొప్పలు చెప్పుకుంటూ జగన్ శునకానందాన్ని పొందుతున్నాడని ఎద్దేవా చేశారు.	By boasting, Jagan was getting Shunakananda.	Sarcastic
మల్కాజ్ గిరిలో తన గెలుపుపై కేటీఆర్ చేసిన వ్యాఖ్యలు గురివింద నామెతను గుర్తు చేస్తున్నాయని రేవంత్ ఎద్దేవా చేశారు	Revanth complained that KTR's comments on his victory in Malkaj Giri reminded him of the Gurivinda proverb.	Sarcastic
హైదరాబాద్ లోని గాంధీ భవన్ లో సాయంత్రం 4 గంటలకు టీపీసీసీ చీఫ్ ఉత్తమ్ కుమార్ రెడ్డి అధ్యక్షతన డీసీసీ అధ్యక్షులు పార్లమెంటు నియోజకవర్గ అభ్యర్థులతో సమావేశం జరగనుంది.	A meeting will be held between DCC presidents and parliamentary constituency candidates under the chairmanship of TPCC Chief Uttam Kumar Reddy at Gandhi Bhavan in Hyderabad.	Non-Sarcastic
ప్రజలకు మూడు పంగనామాలు పెట్టి వెళ్లిపోయారు అని విమర్శించారు.	He criticized that they gave three names to the people and left.	Sarcastic



Figure 2. Output of the TSD-PEMLA model for sarcasm detection in Telugu

SENTIMENT METRIC	SCORE
Positive	0.0
Neutral	0.833
Negative	0.167

Figure 3. Output sentiment metric of proposed TSD-PEMLA using SentiWordNet method

This section details the assessment of the effectiveness of different machine learning models used for sarcasm detection in Telugu language processing (TSD-PEMLA). The experiments took place on a high-performance system with an Intel Core i7 processor and 16 GB of RAM, providing a stable environment for model training and evaluation. The dataset consists of 6,200 manually annotated Telugu sentences sourced from social media, with labels identifying sarcasm ("yes" or "no"). Among 3400 are labelled as 'no' and rest of 2800 are labelled as 'yes'. The dataset was partitioned into 80:20 for training and testing to ensure an equitable and complete evaluation. Python libraries were used to train the models, and performance criteria were applied to assess the effectiveness of each algorithm. The models employed in this study included DT, RF, SVM, KNN, LR, MLP, AdaBoost, CatBoost, XGBoost, and a Voting Classifier.

The sarcasm labeled dataset is provided in Table 1. The output of the proposed TSD-PEMLA model for sarcasm detection in Telugu language processing, including the determination of sarcasm presence ("yes" or "no") using SentiWordNet is shown in Figure 2. The output metrics of the proposed TSD-PEMLA method are presented in Figure 3.

4.1 Performance metrics

The next section describes the evaluation metrics:

Accuracy: It evaluates the fraction of correctly classified instances (sarcasm and non-sarcasm) compared to the total instances. The corresponding expression is represented in Eq. (16).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (16)$$

where, True Positive (TP) refers to correctly predicted sarcasm, while True Negative (TN) refers to correctly predicted non-sarcasm. False Positive (FP) indicates incorrectly predicted sarcasm, and False Negative (FN) refers to missed sarcasm.

Precision: It calculates the proportion of correctly predicted sarcastic instances out of all instances predicted as sarcastic. The corresponding expression is represented in Eq. (17).

$$Precision = \frac{TP}{(TP + FP)} \quad (17)$$

Recall (Sensitivity): It evaluates the ability of the model to correctly identify all relevant instances of sarcasm, that emphasizes the identification of positive cases. The corresponding expression is represented in Eq. (18).

$$Recall = \frac{TP}{(TP + FN)} \quad (18)$$

F1 Score: It is the harmonic mean of precision and recall. The corresponding expression is represented in Eq. (19).

$$F1\ Score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (19)$$

Computation time: It refers to the total time taken by the model to process the dataset and produce predictions. This includes training and classification stages and is measured in seconds (s).

4.2 Performance analysis

Figure 4 outlines the accuracy analysis of TSD-PEMLA for Sarcasm Detection in Telugu using different Word Embeddings. Here, DT and RF achieved accuracy scores of 0.747 and 0.780 respectively. These models demonstrated solid performance based on simple tree-like structures. SVM and KNN achieved accuracies of 0.748 and 0.621 respectively. The SVM performed reasonably well by separating classes in high-dimensional space while KNN's lower score indicated its sensitivity to feature quality. LR, with an accuracy of 0.730, proved effective for binary classification tasks. MLP and AdaBoost achieved accuracies of 0.771. AdaBoost benefited from its ability to focus on misclassified instances. CatBoost and XGBoost, both gradient boosting algorithms, achieved strong performances with accuracy scores of 0.791 and 0.801 respectively. These models showcased the effectiveness of iteratively correcting errors through boosting. Notably, the Voting Classifier achieved the highest performance with an accuracy of 0.923 using CountVectorizer and 0.918 using TF-IDF. By combining predictions from multiple models, the Voting Classifier outperformed individual classifiers can be seen in Table 2. These results suggest that ensemble approaches, particularly those with advanced word embedding techniques, offer a more robust and accurate solution for sarcasm detection in Telugu text.

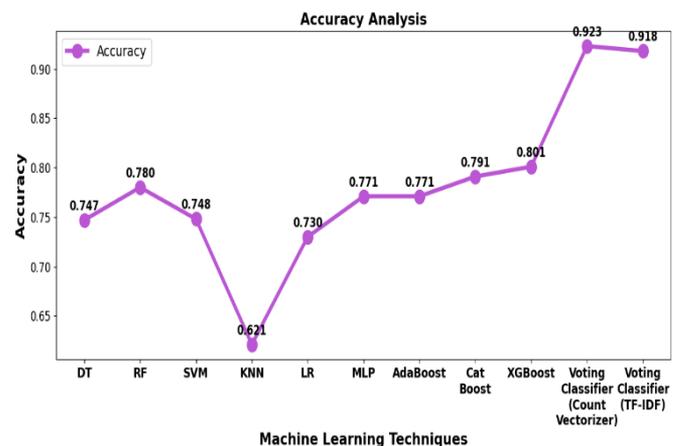


Figure 4. Accuracy analysis among ML algorithms

Table 2. Performance analysis among multiple classifiers

ML Model	Accuracy	Precision	Recall	F1-Score
KNN	0.621	0.681	0.627	0.648
LR	0.730	0.738	0.611	0.661
MLP	0.771	0.760	0.771	0.760
SVM	0.748	0.736	0.748	0.741
DT	0.747	0.740	0.747	0.740
RF	0.780	0.747	0.780	0.750
AdaBoost	0.771	0.704	0.771	0.691
CatBoost	0.791	0.767	0.791	0.738
XGBoost	0.801	0.775	0.797	0.756
Voting Classifier (TF-IDF)	0.918	0.947	0.918	0.925
Voting Classifier (CountVectorizer)	0.923	0.948	0.923	0.929

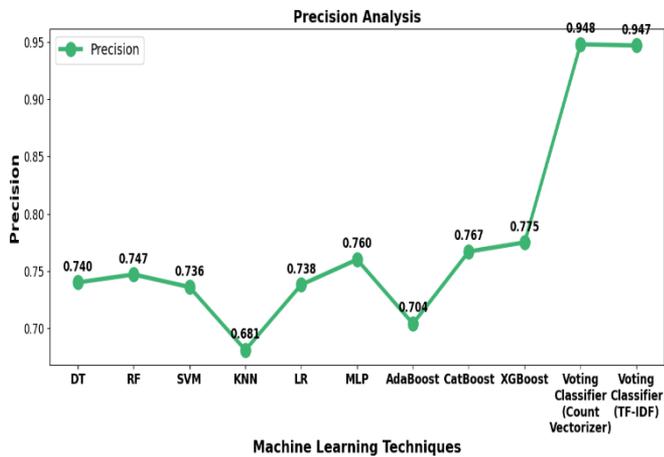


Figure 5. Precision analysis of TSD-PEMLA

The precision results presented in Figure 5. Among the models tested, the Voting Classifier using CountVectorizer achieved the highest precision score of 0.948, closely followed by the Voting Classifier with TF-IDF, which attained a precision of 0.947. These high precision scores indicate the ensemble approach’s ability to accurately identify sarcastic instances while minimizing false positives, thereby enhancing detection reliability. This superior performance of the Voting Classifier underscores the effectiveness of aggregating predictions from multiple classifiers, which improves overall detection accuracy. In comparison, individual classifiers such as XGBoost (0.775) and CatBoost (0.767) demonstrated relatively high precision but did not match the ensemble model's performance. These results suggest that while these models perform well, it still misclassifies some non-sarcastic instances as sarcastic, though to a lesser extent than other models like AdaBoost (0.704) and KNN (0.681), which exhibited noticeably lower precision. The relatively lower precision of models like AdaBoost and KNN further emphasizes the challenge of sarcasm detection. These models tend to generate more false positives, which reduces their precision in sarcasm detection tasks. Other models, such as DT (0.74), RF (0.747), SVM (0.736), LR (0.738), and MLP (0.76), demonstrated moderate precision scores. These models, despite being competent, struggle with distinguishing sarcasm from non-sarcasm and show lower precision than the Voting Classifier but higher precision than AdaBoost and KNN. The overall trend observed in this analysis indicates that combining advanced classifiers with effective feature extraction techniques, such as CountVectorizer and TF-IDF substantially

improve the precision of sarcasm detection in Telugu text.

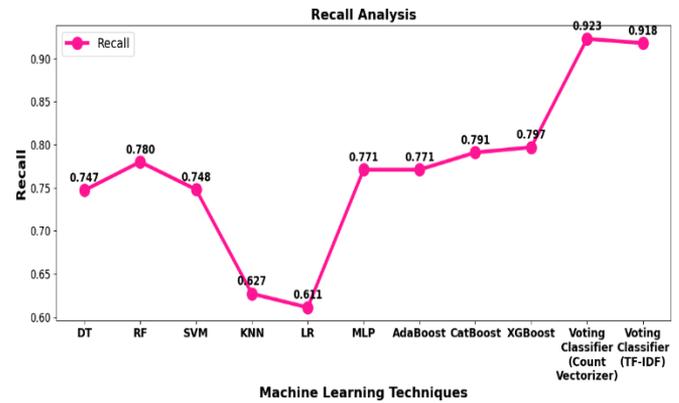


Figure 6. Recall analysis of TSD-PEMLA

The recall results in Figure 6 show that the Voting Classifier with CountVectorizer achieved the highest recall score of 0.923, closely followed by the Voting Classifier with TF-IDF at 0.918. RF and XGBoost had recall scores of 0.780 and 0.797, respectively, indicating strong performance in capturing sarcastic instances. CatBoost achieved a recall score of 0.791, which also demonstrated its effectiveness in sarcasm detection. Models such as MLP and AdaBoost both attained a recall of 0.771, indicating a balanced performance. On the other hand, DT and SVM scored 0.747 and 0.748, respectively, reflecting moderate effectiveness. Models like KNN with 0.627 and LR with 0.611 showed comparatively lower recall, suggesting that they missed a significant portion of sarcastic instances. These results suggest that the ensemble approach, particularly the Voting Classifier, significantly outperforms individual classifiers in terms of recall, providing a more comprehensive and accurate solution for sarcasm detection in Telugu language.

The F1-score values of 0.929 and 0.925, respectively for the Voting Classifier models which is Count Vectorizer and TF-IDF attained the higher F1-scores based on the demonstration of Figure 7. This suggests that as compared to individual models, classification performance is greatly enhanced by ensemble learning. While Logistic Regression (0.661) and KNN (0.648) had the lowest F1-scores among the separate classifiers, MLP (0.760), Random Forest (0.750), and Decision Tree (0.740) all did reasonably well. The effectiveness of boosting approaches was demonstrated by the superior performance of XGBoost (0.756) and AdaBoost (0.738) over some conventional classifiers.

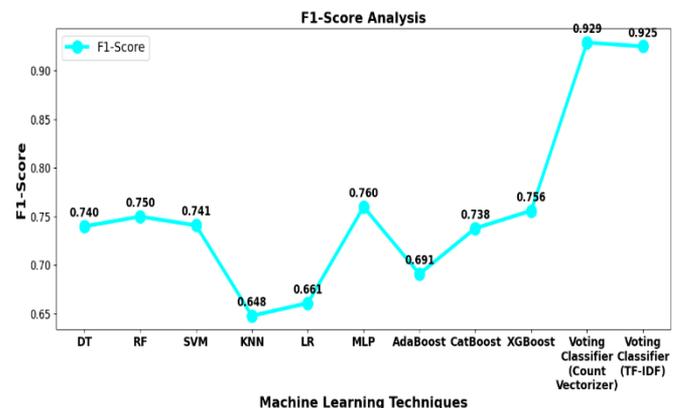


Figure 7. F1-score analysis of TSD-PEMLA

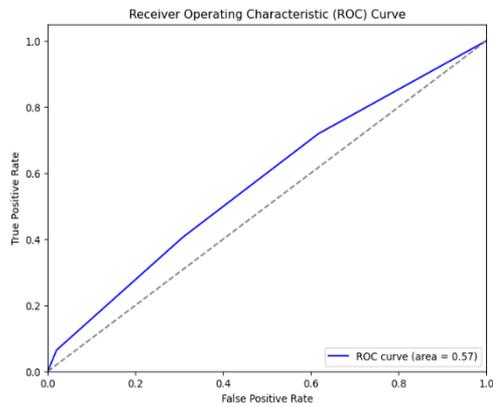


Figure 8. ROC for the KNN classifier

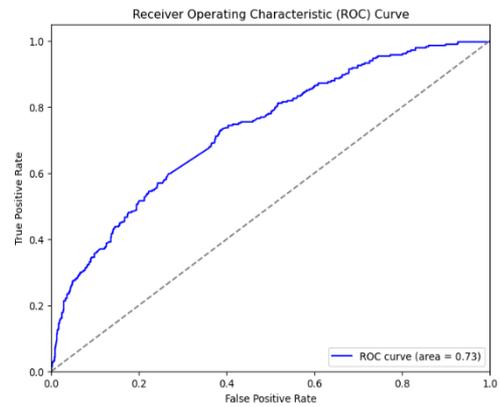


Figure 12. ROC for the XGBoost classifier

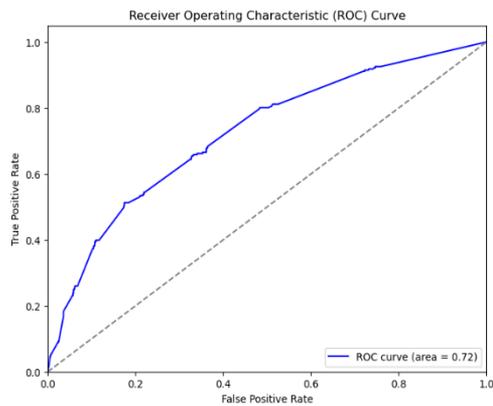


Figure 9. ROC for the DT classifier

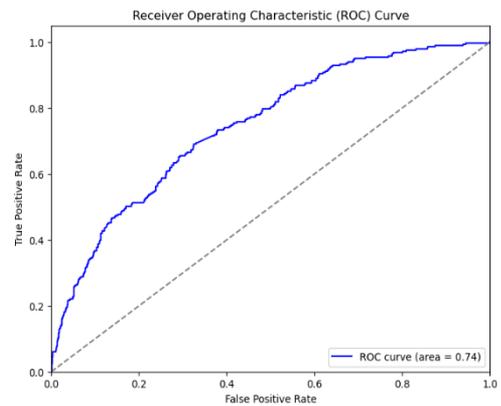


Figure 13. ROC for the CatBoost classifier

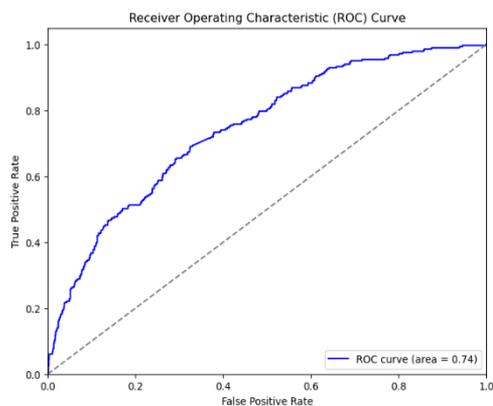


Figure 10. ROC for the RF classifier

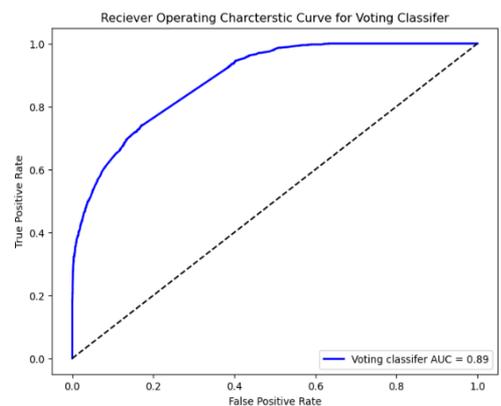


Figure 14. ROC for the Voting Classifier

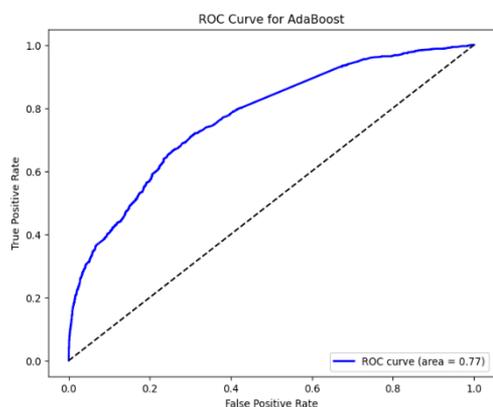


Figure 11. ROC for the AdaBoost classifier

The Receiver Operating Characteristic (ROC) curve for the machine learning algorithms is depicted in the Figures 8-14. The ROC curve depicts the relationship between the true positive rate and the false positive rate. The boosting algorithms acquired an Area Under the Curve (AUC) ranging from 0.71 to 0.74, surpassing the performance of random guessing. Values approaching 1 have major discriminating power. In social media sentiment analysis, a score of 0.71 may be deemed satisfactory as it yields reasonably accurate insights.

4.3 Sarcasm detection with sentiment analysis using SentiWordNet and multi-label classification

The sarcasm dataset was further processed to incorporate

sentiment polarity using SentiWordNet. This process aimed to assess the polarity of the text (positive, negative, or neutral) alongside sarcasm classification. The application of SentiWordNet on the dataset achieved an accuracy of 62.4%. To enhance the model's performance, a dual-label approach was implemented with one label for sarcasm and the other for sentiment value as positive, negative, or neutral.

Due to the association of more than two labels in the dataset based on sentiment polarity, the IAA was assessed using the Fleiss Kappa coefficient [27], calculated according to the formula shown in Eq. (20), yielding a score of 0.86.

$$k = \frac{P - P_e}{1 - P_e} \quad (20)$$

This refined dataset was utilized with AdaBoost and a Voting Classifier, where the Voting Classifier combined Decision Tree, Random Forest, and weighted combination of Gini and Entropy with AdaBoost models with word embeddings through CountVectorizer and TF-IDF. This approach demonstrated an improved balance in performance for detecting sarcasm along with analyzing sentiment by offering enhanced accuracy and reliability during classification tasks for Telugu text.

Table 3. Performance analysis of sarcasm detection combined with sentiment analysis using AdaBoost and Voting Classifier on Telugu text dataset

Metrics	AdaBoost	Voting Classifier
Accuracy	0.518	0.941
Precision	0.611	0.943
Recall	0.518	0.941
F1-score	0.535	0.941
Specificity	0.744	0.97

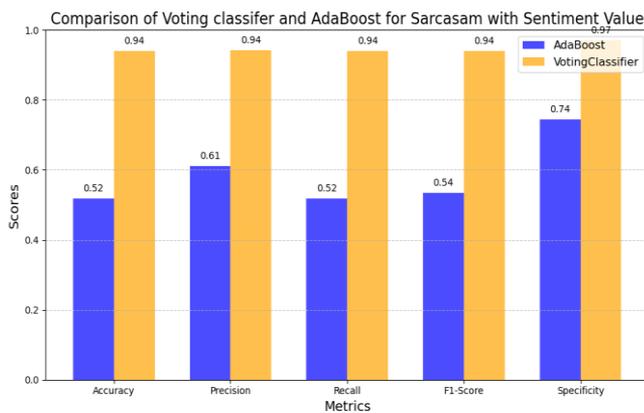


Figure 15. Comparison metrics among two classifiers for sarcasm detection along sentiment analysis

Table 3 analysis focuses on evaluating sarcasm detection combined with sentiment analysis using machine learning models. The CountVectorizer implemented with a Voting Classifier demonstrates superior performance in terms of accuracy, precision, recall, F1-score, and specificity when compared to AdaBoost. The Voting Classifier achieves 81.66% high accuracy by effectively integrating features for classification in comparison to AdaBoost with lower accuracy. The model demonstrates 54.33% high precision in different sentiment contexts in contrast to the precision levels of AdaBoost. With a recall rate of 81.66%, the Voting Classifier

ensures consistent detection of sarcastic elements throughout the dataset. Additionally, the model records a high F1-score of 75.88%, reflecting its balanced handling of false positives and false negatives. A specificity of 30.37% is achieved by the Voting Classifier with its ability to effectively distinguish non-sarcastic content in comparison to AdaBoost as shown in Figure 15.

These findings validate the robustness and reliability of the Voting Classifier in enhancing sarcasm detection and sentiment analysis for Telugu.

Table 4 illustrates the performance analysis of non-sarcasm detection combined with sentiment analysis using AdaBoost and Voting Classifier on a Telugu text dataset. The analysis highlights specific metrics for both models and showcases their effectiveness in various aspects. The Voting Classifier demonstrates a 6.45% increase in accuracy, a 0.82% decrease in precision, a 6.45% improvement in recall, a 2.51% enhancement in F1-Score, and a 31.92% rise in specificity when compared to AdaBoost. These results confirm the robustness of the Voting Classifier as an effective tool for non-sarcasm detection combined with sentiment analysis on Telugu text datasets which is shown in Figure 16.

Table 4. Performance analysis of non-sarcasm detection combined with sentiment analysis using AdaBoost and Voting Classifier on Telugu text dataset

Metrics	AdaBoost	Voting Classifier
Accuracy	0.867	0.923
Precision	0.97	0.962
Recall	0.867	0.923
F1-score	0.913	0.936
Specificity	0.755	0.996

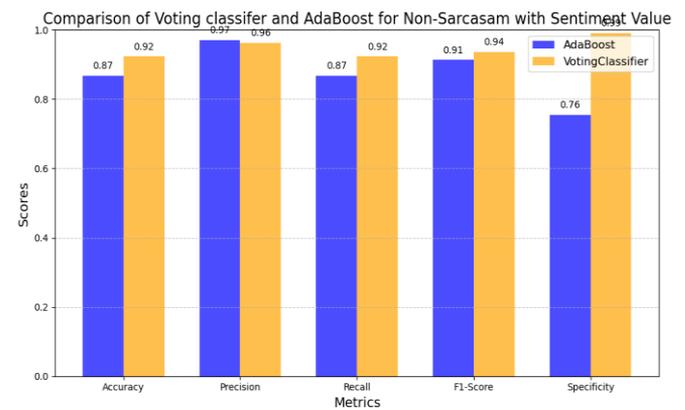


Figure 16. Comparison metrics among two classifiers for non-sarcasm detection along sentiment analysis

The Voting Classifier exhibits a computation time of 8164.15 ms, which is higher compared to AdaBoost, which completes the task in 150.60 ms. Despite the longer computation time, the Voting Classifier outperforms AdaBoost in terms of accuracy, precision, recall, and specificity. These findings highlight the trade-off between model performance and computational efficiency in the context of sarcasm detection with sentiment analysis.

The evaluation of several machine learning models for sarcasm detection in Telugu text offers key insights into the efficiency of various strategies and emphasizes the benefits of combining multiple classifiers with advanced word representations. Ensemble models outperformed individual

models in all metrics with effectiveness observed in the Voting Classifier. The Voting Classifier achieved high performance through the combination of multiple models, with a focus on leveraging their strengths for a robust and accurate solution to sarcasm detection. The Voting Classifier's integration of classifiers like Decision Tree, Random Forest, and AdaBoost enhanced its performance by compensating for the weaknesses of individual models. It demonstrated high precision and recall values, indicating its effectiveness in detecting sarcasm accurately while minimizing false positives and false negatives using Figures 4-7.

The introduction of sentiment analysis using SentiWordNet to the sarcasm dataset and the implementation of dual-label classification (sarcasm and sentiment) further improved the model's performance. While SentiWordNet alone achieved an accuracy of 62.4%, the combination of sarcasm detection with sentiment analysis provided a more comprehensive understanding of the text through the capture of both sarcastic tone and sentiment. The Voting Classifier's improved performance with this dual-label approach indicates that multi-label classification is effective for enhancing sarcasm detection by considering sentiment polarity simultaneously.

The comparison in Table 3 between AdaBoost and the Voting Classifier in sarcasm detection combined with sentiment analysis shows the superiority of the ensemble approach. The Voting Classifier achieved significant improvements in accuracy (0.941), precision (0.943), recall (0.941), and F1-score (0.941), highlighting the benefits of aggregating predictions from multiple models. In contrast to the Voting Classifier, AdaBoost showed lower accuracy and performance metrics, that indicates limitations in robustness through its focus on misclassified instances and its challenges with handling complex datasets like sarcasm detection.

In summary, the results emphasize the importance of ensemble methods through their combination with advanced word embeddings such as CountVectorizer and TF-IDF. In addition to this, the integration of sentiment analysis by means of multi-label classification demonstrates a promising approach for improving sarcasm detection accuracy and reliability in complex languages such as Telugu. For future research, emphasis could be placed on optimizing these models further along with exploring hybrid approaches through the combination of deep learning techniques with traditional machine learning models for addressing more complex natural language processing challenges.

5. CONCLUSION

The TSD-PEMLA model demonstrated significant improvements in sarcasm detection for Telugu text using data collected from social media platforms. The preprocessing steps such as tokenization, normalization, stop-word removal, and morphological processing, were applied to prepare the data. Effective feature extraction techniques including n-grams, TF-IDF, and sentiment scores derived from SentiWordNet, represented the data effectively. Training was conducted on various traditional machine learning models, including DT, RF, SVM, KNN, LR, MLP, AdaBoost, CatBoost, XGBoost, and an ensemble Voting Classifier. The Voting Classifier achieved superior performance by combining predictions from multiple classifiers to achieve better accuracy and overall metrics. The ensemble approach enhanced by word embeddings such as CountVectorizer and

TF-IDF provided a strong mechanism for sarcasm detection and outperformed individual models. Further experiments with SentiWordNet for polarity detection resulted in moderate outcomes, which improved through the training of classifiers with dual labels for sarcasm and sentiment classification. AdaBoost and Voting Classifiers showed noticeable improvements in accuracy when sarcasm detection was combined with sentiment analysis. The integration of ensemble models with feature extraction methods such as TF-IDF addressed the challenges of sarcasm detection in Telugu. Although there was a slight trade-off in computational time, the improved performance justified the approach. Future research will explore advanced hybrid deep learning architectures to capture nuanced patterns in Telugu language. Contextual embeddings from TeluguBERT and character-level convolutional neural networks (CNNs). This guaranteed that unfamiliar words like in Telugu to handle OOV words could still be accurately represented using subword patterns, morphemes. In Future increase the dataset size to handle effectively. This research emphasizes the potential of combining ensemble techniques with enriched feature extraction processes to develop effective solutions for linguistic challenges in resource-constrained environments.

ACKNOWLEDGMENT

I extend my gratitude to T. Muni Krishna Reddy, B.Ed., retired government teacher from Tirupati, and D. Madhan Mohan Reddy, M.Phil. in Telugu literature, for their contributions in annotating the Telugu dataset.

REFERENCES

- [1] Rahma, A., Azab, S.S., Mohammed, A. (2023). A comprehensive survey on Arabic sarcasm detection: Approaches, challenges and future trends. *IEEE Access*, 11: 18261-18280. <https://doi.org/10.1109/ACCESS.2023.3247427>
- [2] Du, Y., Li, T., Pathan, M.S., Teklehaimanot, H.K., Yang, Z. (2022). An effective sarcasm detection approach based on sentimental context and individual expression habits. *Cognitive Computation*, 14(1): 78-90. <https://doi.org/10.1007/s12559-021-09832-x>
- [3] Vitman, O., Kostiuk, Y., Sidorov, G., Gelbukh, A. (2023). Sarcasm detection framework using context, emotion and sentiment features. *Expert Systems with Applications*, 234: 121068. <https://doi.org/10.1016/j.eswa.2023.121068>
- [4] Ren, Y., Wang, Z., Peng, Q., Ji, D. (2023). A knowledge-augmented neural network model for sarcasm detection. *Information Processing & Management*, 60(6): 103521. <https://doi.org/10.1016/j.ipm.2023.103521>
- [5] Maaad, A.Y., Jayappa Davanagere, H., Benifa, J.B., Alabrah, A., et al. (2022). Artificial intelligence-based approach for misogyny and sarcasm detection from Arabic texts. *Computational Intelligence and Neuroscience*, 2022(1): 7937667. <https://doi.org/10.1155/2022/7937667>
- [6] Ding, N., Tian, S.W., Yu, L. (2022). A multimodal fusion method for sarcasm detection based on late fusion. *Multimedia Tools and Applications*, 81(6): 8597-8616. <https://doi.org/10.1007/s11042-022-12122-9>

- [7] Tiwari, P., Zhang, L., Qu, Z., Muhammad, G. (2024). Quantum fuzzy neural network for multimodal sentiment and sarcasm detection. *Information Fusion*, 103: 102085. <https://doi.org/10.1016/j.inffus.2023.102085>
- [8] Chen, W., Lin, F., Li, G., Liu, B. (2024). A survey of automatic sarcasm detection: Fundamental theories, formulation, datasets, detection methods, and opportunities. *Neurocomputing*, 578: 127428. <https://doi.org/10.1016/j.neucom.2024.127428>
- [9] Govindan, V., Balakrishnan, V. (2022). A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for sarcasm detection. *Journal of King Saud University-Computer and Information Sciences*, 34(8): 5110-5120. <https://doi.org/10.1016/j.jksuci.2022.01.008>
- [10] Kamal, A., Abulaish, M. (2022). Cat-BIGRU: Convolution and attention with bi-directional gated recurrent unit for self-deprecating sarcasm detection. *Cognitive Computation*, 14(1): 91-109. <https://doi.org/10.1007/s12559-021-09821-0>
- [11] Wen, Z., Gui, L., Wang, Q., Guo, M., Yu, X., Du, J., Xu, R. (2022). Sememe knowledge and auxiliary information enhanced approach for sarcasm detection. *Information Processing & Management*, 59(3): 102883. <https://doi.org/10.1016/j.ipm.2022.102883>
- [12] Vinoth, D., Prabhavathy, P. (2022). Automated sarcasm detection and classification using hyperparameter tuned deep learning model for social networks. *Expert Systems*, 39(10): e13107. <https://doi.org/10.1111/exsy.13107>
- [13] Lu, Q., Long, Y., Sun, X., Feng, J., Zhang, H. (2024). Fact-sentiment incongruity combination network for multimodal sarcasm detection. *Information Fusion*, 104: 102203. <https://doi.org/10.1016/j.inffus.2023.102203>
- [14] Galal, M.A., Yousef, A.H., Zayed, H.H., Medhat, W. (2024). Arabic sarcasm detection: An enhanced fine-tuned language model approach. *Ain Shams Engineering Journal*, 15(6): 102736. <https://doi.org/10.1016/j.asej.2024.102736>
- [15] Wang, J., Yang, Y., Jiang, Y., Ma, M., Xie, Z., Li, T. (2024). Cross-modal incongruity aligning and collaborating for multi-modal sarcasm detection. *Information Fusion*, 103: 102132. <https://doi.org/10.1016/j.inffus.2023.102132>
- [16] Kumar, R.P., Mohan, G.B., Kakarla, Y., Jayaprakash, S.L., et al. (2023). Sarcasm detection in Telugu and Tamil: An exploration of machine learning and deep neural networks. In 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, pp. 1-7. <https://doi.org/10.1109/ICCCNT56998.2023.10306775>
- [17] Eke, C.I., Norman, A.A., Shuib, L. (2021). Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and BERT model. *IEEE Access*, 9: 48501-48518. <https://doi.org/10.1109/ACCESS.2021.3068323>
- [18] Goel, P., Jain, R., Nayyar, A., Singhal, S., Srivastava, M. (2022). Sarcasm detection using deep learning and ensemble learning. *Multimedia Tools and Applications*, 81(30): 43229-43252. <https://doi.org/10.1007/s11042-022-12930-z>
- [19] Ratnavel, R., Joshua, R.G., Varsini, S.R., Kumar, M.A. (2023). Sarcasm detection in Tamil code-mixed data using transformers. In International Conference on Speech and Language Technologies for Low-resource Languages, pp. 430-442. https://doi.org/10.1007/978-3-031-58495-4_32
- [20] Vinoth, D., Prabhavathy, P. (2022). An intelligent machine learning-based sarcasm detection and classification model on social networks. *The Journal of Supercomputing*, 78(8): 10575-10594. <https://doi.org/10.1007/s11227-022-04312-x>
- [21] Kumar, A., Garg, G. (2023). Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. *Journal of Ambient Intelligence and Humanized Computing*, 14(5): 5327-5342. <https://doi.org/10.1007/s12652-019-01419-7>
- [22] Gedela, R.T., Baruah, U., Soni, B. (2024). Deep contextualised text representation and learning for sarcasm detection. *Arabian Journal for Science and Engineering*, 49(3): 3719-3734. <https://doi.org/10.1007/s13369-023-08170-4>
- [23] Elbarougy, R., Behery, G., El Khatib, A. (2019). A proposed natural language processing preprocessing procedures for enhancing Arabic text summarization. In Recent Advances in NLP: The Case of Arabic Language, pp. 39-57. https://doi.org/10.1007/978-3-030-34614-0_3
- [24] Qasem, A.E., Sajid, M. (2022). Exploring the effect of N-grams with BOW and TF-IDF representations on detecting fake news. In 2022 International Conference on Data Analytics for Business and Industry (ICDABI), Sakhir, Bahrain, pp. 741-746. <https://doi.org/10.1109/ICDABI56818.2022.10041537>
- [25] Mienye, I.D., Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10: 99129-99149. <https://doi.org/10.1109/ACCESS.2022.3207287>
- [26] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1): 37-46. <https://doi.org/10.1177/001316446002000104>
- [27] Fleiss, J.L., Cohen, J., Everitt, B.S. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, 72(5): 323. <https://doi.org/10.1037/h0028106>