SN Computer Science (2026) 7:36
https://doi.org/10.1007/s42979-025-04557-2

ORIGINAL RESEARCH

Sarcasm Detection and Classification in Kannada Language Using Machine Learning Techniques on a Manually Annotated Dataset

Santosh Chinchali[1] · **Pushpa. B. Patil**[2]

Received: 30 June 2025 / Accepted: 17 November 2025

Abstract

Sarcasm, a form of expression where the intended meaning contradicts the literal wording, presents significant challenges for computational interpretation, especially in regional languages like Kannada. Accurate sarcasm detection is vital for applications such as sentiment analysis, opinion mining, and social media monitoring. This study tackles the challenges of detecting sarcasm in Kannada by utilizing advanced techniques. A comparatively large and diverse dataset comprising 7000 annotated sentences is larger compared to previous studies. Standard text preprocessing steps such as data cleaning, tokenization, stop word removal, stemming, and part-of-speech tagging were applied to the dataset. Several machine learning and deep learning models were then trained and evaluated, including SVC, SGD Classifier, Multinomial Naive Bayes, Random Forest, Linear SVC, Logistic Regression, XGBoost, and BERT. Compared to all these techniques, the Logistic Regression model achieved the highest accuracy of 67.4%, highlighting the challenges in capturing the nuances of sarcasm in Kannada. Although the performance is modest compared to smaller-scale studies, the results underscore the importance of dataset size and diversity in model robustness. Future research should focus on exploring more advanced architectures, such as transformer-based models and recurrent neural networks, along with further expansion of annotated datasets to improve accuracy and generalizability in regional sarcasm detection.

Keywords Sarcasm detection · Machine learning · NLP

Introduction

A common communication tool in both spoken and written language is sarcasm. It entails writing or expressing the exact opposite of what is intended, frequently with a comic or sardonic tone. Although sarcasm can be an effective communication strategy, it can sometimes be difficult to understand, especially when used in text-based communication like emails, texts, and postings on social media. Detecting sarcasm in Kannada text is one of the major challenges in Natural Language Processing (NLP). Understanding when someone is being sarcastic is essential for tasks such as opinion mining and monitoring content on media platforms. Sarcasm is particularly difficult to identify because it often depends on both linguistic and social context. It can be conveyed through irony, exaggeration, understatement, rhetorical questions, or conflicting sentiments within the same sentence. Moreover, sarcasm often relies on tone, which is inherently difficult to capture in written text.

Recent developments in machine learning and Natural Language Processing (NLP) have greatly enhanced the ability to detect sarcasm in text. Researchers have been continuously working on building models that can automatically identify patterns and features linked to sarcastic language. These models often rely on supervised learning methods, wherein algorithms are trained using labeled datasets that include both sarcastic and non-sarcastic examples [7]. Alongside machine learning methods, rule-based

✉ Santosh Chinchali
cse.santoshchinchali@bldeacet.ac.in

Pushpa. B. Patil
pushpapatil2008@gmail.com

[1] Department of Computer Science and Engineering, BLDEA's VP Dr. PG Halakatti College of Engineering and Technology, (Affiliated to Visvesvaraya Technological University, Belagavi 590018), Vijayapura 586103, Karanataka, India

[2] Department of Computer Science and Engineering (Data Science), BLDEA's VP Dr. PG Halakatti College of Engineering and Technology, (Affiliated to Visvesvaraya Technological University, Belagavi 590018), Vijayapura 586103, Karanataka, India

Published online: 26 December 2025

approaches have also been explored. These use predefined linguistic rules or heuristics to detect sarcasm. For example, a rule-based method might classify a sentence as sarcastic if it exhibits strong sentiment polarity or contains words with contrasting meanings.

Despite these advancements, sarcasm detection remains a complex problem, especially for non-English languages [8]. The unique linguistic features and cultural nuances of each language make it difficult to develop universally effective models [9]. In the case of Kannada, sarcasm detection is further complicated by regional and social factors. Communication in Kannada will vary between urban and rural communities, and the interpretation of sarcasm can depend on the speaker's age, gender, or social status. These cultural and geographical variations pose a major challenge in building robust sarcasm detection systems that generalize well across different user groups and contexts.

The contributions of proposed approach are:

- In contrast to most existing models that predominantly utilize Kannada-English code-mixed datasets, the proposed model is specifically developed to operate on monolingual Kannada texts, thereby addressing the underexplored challenge of sarcasm detection in pure regional language content.
- The suggested model was trained on a significantly bigger dataset than previous studies, which generally used small amounts of data. This made sarcasm detection more accurate and dependable.
- By combining BERT with an innovative machine learning approach, the proposed work demonstrates improved classification efficiency, addressing the limitations of prior Kannada language models, which largely employ CNN-based methods with average performance.

The further paper is organized as follows. Section "Related Work" Covers the Literature Survey. Section "Proposed Methodology" Methodology provides a detailed explanation of the proposed methodology for sarcasm detection, giving a full description of the machine learning algorithms used for classification jobs. Section "Results and Discussions" Results and Discussion demonstrates the analysis of the performance of the model. Section "Conclusion" Conclusion includes conclusion of the work.

Related Work

In 2021, Hande et al. [1] proposed a Model for contemporary methods to stop the propagation of negativity, such as clearing out offensive, insulting, and poisonous comments from social networking sites. Research on fostering optimism and promoting reassuring and supportive content in online forums. However. Therefore, researchers used the KanHope English-Kannada Hope speech dataset and compared its findings with other studies. Wherein, they used 6176 user-generated comments in a combination of Kannada and code-mixed sentences. The comments were retrieved from YouTube and meticulously categorized as either a hope speech or not. Furthermore, they developed DC-BERT4HOPE, a dual-channel model trained on the English version of KanHope to identify hope speech. With a weighted F1-score of 0.756, their approach outperformed other models. KanHope helped advance research in Kannada by encouraging practical approaches to online data. The model achieved higher accuracy largely because it used a mix of Kannada and English.

In 2023, Sharma et al. [2] proposed a hybrid ensemble model uses fuzzy logic for the detection of sarcasm in social media, improving its accuracy by 90% by combining multiple models in one framework, handling the complexity of informal data environments. The fuzzy logic technique was applied to classify tweets or statements as sarcastic or not using fuzzy rules.

In 2023, Misra and Arora [3] adopted a dataset of news headlines for sarcasm detection, and different machine learning techniques were employed with a focus on tiny language cues for sarcastic tones in formal news text. The proposed model uses manually labeled dataset because the understanding of sarcasm changes between individuals states of mind. The datasets in the proposed model used to train the deep learning model aren't looked into very much, which means it's hard to find real sarcastic expressions. This study successfully trained a fairly complex neural network model using the News Headlines Dataset and achieved an accuracy of up to 90%.

In 2022, Savini et al. [4] worked on BERT for sarcasm detection, the author's experiments on different datasets of different volumes to cover characteristics from social platforms. Adapt intermediate-task transfer learning for fine-tuning to improve the model's performance in detecting sarcasm using enhanced contextual understanding ability. This study achieved accuracy with an F1-score 93.5.

In 2021, Scola E and Segura-Bedmar [5] proposed a work to focused on BERT (Bidirectional Encoder Representations from Transformers). This work examines sarcasm recognition in textual data, including news headlines. In comparison with previous models such as BiLSTM and conventional machine learning techniques, BERT uses contextual embeddings to identify minor sarcastic linguistic distinctions. The process entails optimizing BERT using a dataset of positive and negative headlines for sarcasm detection. When pretreatment processes like tokenization and embedding preparation are implemented consistently across

models, the performance of BERT and BiLSTM is compared. The robustness of the BERT-based model in sarcasm detection was demonstrated by its improved performance over BiLSTM. To enhance accuracy, the study focuses on the significance of contextual embeddings and recommends investigating hybrid approaches and bigger datasets in subsequent research.

In 2021, Ranjitha and Bhanu [6] proposed a model for analyzing a piece of Kannada-language writing as having a good, negative, or neutral attitude toward a certain topic or product. The authors of this work created a method that computationally recognizes and classifies thoughts contained in the writing. The decision tree approach for Kannada emotion analysis is used to achieve this. Dataset prepared using different newspapers such as Websites such as Prajwani, One India News. The results showed 85% accuracy, 0.78 precision, and 0.79 recall. This study's drawback is that certain Kannada phrases cause machine translation to give ambiguous messages, leading to erroneous results. Sentiment analysis would be easier than analyzing and detecting sarcasm in pure Kannada language.

In 2023, Manohar and Swamy [7] introduced an approach to detect sarcasm in Kannada. They used a hybrid methodology that makes the use of traditional machine learning methods like Support Vector Machines (SVM), Random Forests with deep learning techniques and Bidirectional Long Short-Term Memory (BiLSTM). Various preprocessing steps were used in the analysis of Kannada textual data including tokenization, stemming, and stop-word removal techniques. The next step was feature extraction, which concentrated on the language's linguistic, syntactic, and semantic components. The model had a promising accuracy of 83.57% after five training epochs. The accuracy decreased slightly to 76.10% across 40 training epochs, but the loss curve remained constant, indicating the need for further optimization. The authors recommended including audio to enhance the model's sarcasm recognition abilities.

In 2024, Manohar and Swamy [8] proposed a model to work on using audio data to detect sarcasm. This work investigates sarcasm detection in spoken Kannada. It highlights the difficulties caused by Kannada's distinct linguistic and prosodic characteristics, such as tone, pitch, and accent. In contrast to textual methods, this study emphasizes the usefulness of voice characteristics such as prosody, rhythm, and Mel-Frequency Cepstral Coefficients (MFCCs) in identifying sarcastic statements. Data preprocessing (such as noise reduction and speech-to-text conversion), feature extraction (textual and prosodic) and training hybrid models that ensemble deep learning models such as Support Vector Machines (SVM) and Long Short-Term Memory networks (LSTMs) with machine learning techniques are all part of the methodology. The suggested model's initial accuracy of 57.2% is regarded as a starting point for additional study.

In 2021, Bharti et al. [9] proposed a model that works on utilizing sentiment analysis; it has proven challenging to recognize sarcastic statements. Sarcastically, a phrase expresses an unfavorable attitude using only positive terms. It was so challenging for any automated program to ascertain the precise sentiment of the text due to sarcasm. The current tools can only read English-scripted text to figure out if a message is sarcastic. In recent years, researchers have shown growing interest in low-resource languages such as Marathi, Telugu, Tamil, Arabic, Chinese, Dutch, Indonesian, and others. In the case of Indian languages, the lack of resources is the main barrier to analyzing these low-resource languages. Automated robots have a harder time understanding Indian languages because of their incredibly complex morphology. Telugu is one of the most spoken languages in India, second only to Hindi. Authors collected and annotated a corpus of Telugu talk. Sarcasm can be identified by phrases that begin with an inquiry and conclude with an answer. In addition to a set of algorithms, the study also proposes analyzing sarcasm in a collection of Telugu conversational phrases. The suggested algorithms are based on four key hyperbolic features: interjections, intensifiers, question marks, and exclamation marks. 94% accuracy was attained. This high result may be attributed to the use of hyperbolic features (e.g., interjection, intensifier) in a controlled dataset, which might not generalize to less-structured or diverse datasets.

In 2021, Akula and Garibay [10] presented a work to concentrate on a linguistic concept called sarcasm. The Term sarcasm is commonly used to express the contrast of what is being said, typically something very disagreeable, with the intention of offending or mocking. Because sarcastic statements are by their very nature vague, sarcasm detection is especially difficult. Their primary goal in this study is to detect sarcasm in Kannada text messages from different online media and social networking sites. To do this, they created a deep learning model that could be understood by using controlled recurrent units and multi-head self-attention. They achieved promising results on diverse datasets, highlighting the effectiveness of their method from online media and social networks, with an accuracy of 0.76 with a smaller volume English dataset. The models developed using their proposed approach are interpretable and allow for the identification of sarcastic elements in the input text that affect classification decisions. To showcase both the effectiveness and interpretability of their method, they visualized learned attention weights using selected example texts.

In 2021, Akula and Garibay [11] proposed a work to illustrate the effectiveness and interpretability, reaching

leading performance benchmarks on datasets drawn from online discussions, social networking sites, and political discussions during the online conversations. The need for manually generated characteristics has been eliminated by recent studies. Several studies have applied deep learning methods, employing neural networks to extract lexical and contextual information. These models are commonly trained using word embeddings and include architectures such as convolutional, recurrent, and attention-based neural networks.

In 2022, Moores and Mago [12] proposed a topic on automatic sarcasm detection is expanding with advancements in computer science, and short text sentences are frequently used for communication, particularly on Twitter. Unidentified sarcasm in these messages may cause misunderstanding and communication breakdowns by inadvertently reversing the meaning of a statement due to inadequate or absent context. This article explores several contemporary methods for sarcasm detection, including machine learning models, posting history, and context-based detection. Furthermore, there is a discernible shift in favor of deep learning methods, which is most likely due to the benefits of using models with induced features as opposed to discrete ones and the development of transformers. The model tends to achieve an F-1 score of 0.97.

In 2018,Ghosh et al. [13] Analyzed the conversation context to detect sarcasm, showing how multi-turn dialogue could be better in understanding and the detection of sarcasm in conversational settings, such as human annotator can identify a sentence in a sarcastic post, and Sentence-level attention weights can highlight relevant information in the text. The proposed study is designed to interpret the attention mechanism within LSTM architectures and achieves an F1-score of 69.88.

The proposed model is evaluated using a larger dataset compared to previous studies. This model demonstrates improved classification efficiency by addressing the limitations of earlier Kannada language models, which were constrained by limited accuracy due to the use of low-volume Kannada text data.

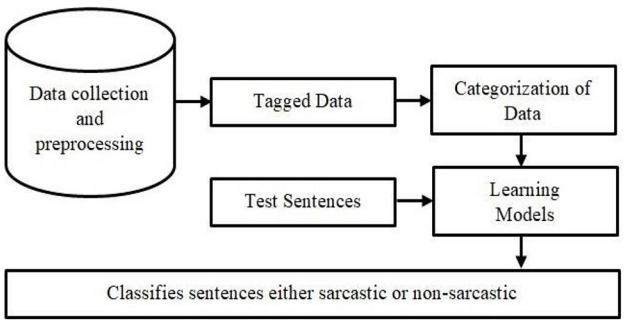

Fig. 1 Steps involved in sarcasm detection and classification in Kannada text

Challenges

- **Dependence on Data Quality**: The model's effectiveness is limited by the quality and diversity of the training data, potentially leading to poor performance on unseen or diverse datasets.
- **Contextual Understanding Challenges**: Since sarcasm is heavily influenced by contextual cues and tonal subtleties, traditional machine learning models, and even advanced models like BERT, may struggle to accurately detect it in cases where the meaning is ambiguous.
- **Language-Specific Limitations**: The nuances of Kannada, including slang, idiomatic expressions, and code-mixed text, pose significant challenges for the model's preprocessing techniques, like stemming and TF-IDF.
- **High Computational Requirements**: Advanced models like BERT demand significant computational resources and reducing their viability for low-resource or real-time applications.
- **Subjectivity in Labels**: Sarcasm annotation often varies across annotators due to its subjective nature, leading to inconsistencies in the training data and impacting model accuracy.

Proposed Methodology

This work examines sarcasm recognition in the Kannada Language textual data. The process entails optimizing Machine learning techniques using a dataset of sarcastic and non-sarcastic sentences for sarcasm detection. When pretreatment processes like tokenization and embedding preparation are implemented consistently across models, Different machine learning techniques compared.

The steps involved in the collection of Kannada text taken from native speakers or everyday conversations are depicted in Fig. 1. The sentences will be removed from this dataset. After being retrieved, sentences will be tokenized, or divided into smaller tokens. Data cleaning is the process of removing unusual content from the sentences that does not add any significance to the statement, such as punctuation, commas, etc [14]. In languages, stop words are terms such as the, which, and, etc. that add no meaning to a phrase. Stemming is the process of splitting a word and determining its fundamental word. Text tagging, the process of classifying groupings of texts into their respective categories, is another name for classification. The last step is calculation, where the sentence is analyzed to identify whether a given sentence is sarcastic or non-sarcastic.

Preprocessing Techniques

The subsequent section outlines the text preprocessing procedure implemented on the input data.

Data Collection

Data collection means finding, organizing, and gathering information that belongs to a specific topic or purpose. It makes it possible to compute approximate results and provide helpful answers to questions. In many disciplines, including the social and physical sciences, as well as in humankind and employment, data collection is a valuable component. It is essential to accurately characterize and compile the data during data collection to maintain the integrity of the investigation.

Tokenization

The act of breaking a sentence, paragraph, or text document into discrete units known as tokens is known as tokenization. Tokens might be words, phrases, keywords, or characters [10]. We can take the example "It is a pen" as an example. The fundamental method of tokenization is to create a token by considering the space. The aforementioned text is reduced to the tokens "It," "is," "a," and "pen" after passing through the tokenization procedure. Each reduced token in this case is a word. We can tokenize sentences and documents.

Data Cleaning

One crucial phase in NLP is data cleaning. Without data cleansing, the dataset is like a collection of unintelligible words that the computer cannot comprehend. This stage entails locating redundant, inaccurate, or ancillary data, after which the undesirable data is changed, replaced, or removed. Data cleaning is the procedure for dividing a sentence in natural language processing (NLP), data cleaning comprises removing various punctuation marks, such as commas, colons, exclamation points, hyphens, question marks, apostrophes, dashes, brackets, semicolons, brackets, brackets, ellipses, and quote marks.

Removing Stop Words

Stop words are common words or phrases that don't add much meaning to a sentence in any language. Eliminating these stop words won't change the sentence's real meaning [15]. Eliminating these stop words will improve performance and accuracy while reducing the amount of data and training time. The NLTK library provides various modules for NLP, including a corpus module that contains a collection of stopwords to help exclude them from text processing. Stopwords, common words with minimal meaning, were found and removed to reduce the noise in the dataset. Articles, prepositions, and conjunctions that are frequently used in Kannada were compiled into a list of stopwords. Following tokenization and stemming, stopword elimination was carried out to preserve linguistic coherence [3].

Categorization

As part of the categorization process, data are grouped into various partitions according to their attributes. The TF-IDF approach is used to extract features prior to classification. Several classification algorithms have been tested in this work, namely Multinomial Naive Bayes, which uses a selective learning method, the SGD Classifier uses a simple Stochastic Gradient Descent routine for classification, supporting different loss functions and penalties. Logistic Regression is a supervised algorithm that classifies the probability of a word belonging to a specific category. The Gaussian Naïve Bayes Classifier assumes a normal distribution and is effective for handling continuous data. Furthermore, the BERT model was put into practice, which improved the classification process by utilizing its deep contextual embedding, especially for identifying subtle patterns in Kannada sentences [12].

Feature Extraction

The TF-IDF method is used to pick out important words from text. It has two components:

Term Frequency (TF) shows how often a word appears in a document compared to the total number of words. Inverse Document Frequency (IDF) looks at how common or rare a word is across many documents — giving less importance to words that appear in lots of them.

A sparse matrix is created following the extraction of TF-IDF features. Classification is done using this matrix. To train the machine, we employed in-language classification. This technique works by training classifiers in the specific language being analyzed, which means there must be enough language resources available for it to work effectively. As a result, all testing and training data are in the form of Kannada text. To train and test the data, we employed a range of classifiers, including the SGD Classifier, Multinomial NB, Logistic Regression, Random Forest Classifier, Linear SVC, XGBoost, BERT and SVC [11].

Models Used

Several machine learning methods have been applied to perform classification. Kannada sentences as either sarcastic or non-sarcastic as follows.

(i) **Linear SVC**
To identify sarcasm in Kannada words, our model used Linear SVC (Support Vector Classifier) as one of the classification algorithms. A supervised machine learning approach called Linear SVC divides data into discrete groups by locating a hyperplane. It performs well in text classification tasks and is effective with high-dimensional datasets. With an accuracy of 66.71% in our tests, Linear SVC demonstrated its capacity to successfully identify patterns in the dataset.

(ii) **Logistic Regression**
One of the classifiers used in our model to identify sarcasm in Kannada words was logistic regression. A probabilistic approach called logistic regression uses a sentence's properties to forecast the likelihood that it belongs to a specific class. It is widely used for binary classification tasks due to its effectiveness and ease of use. One of the best algorithms for this assignment was Logistic Regression, which in our tests had an accuracy of 67.19%.

(iii) **SGD Classifier**
Our model used the Stochastic Gradient Descent (SGD) Classifier to detect sarcasm in Kannada sentences. The SGD Classifier optimizes the model iteratively by updating weights based on a stochastic approximation of the gradient of the loss function, which works especially well with sparse features and large-scale data. The SGD Classifier's accuracy of 66.78% in our experiments showed that it could handle the complexity of sarcasm detection.

(iv) **SVC**
Support Vector Classification (SVC) is one machine learning method for classification tasks. Finding the best hyperplane to divide data points from various classes is how Support Vector Classification (SVC) operates. The margin, or the separation between the hyperplane and the nearest data points from each class, is maximized by selecting this hyperplane. SVC employs a kernel function to convert data into a higher-dimensional space in situations where the data isn't linearly separable, which facilitates the drawing of distinct class boundaries [4].

(v) **Multinomial Naïve Bayes**
Multinomial Naive Bayes is a commonly used NLP algorithm that follows a probabilistic learning approach. The method predicts the label of the considered textual input and is impacted by the Bayes' theorem. Each label's likelihood is determined, and the resultant label with the greatest value is provided. This method is a group of several algorithms that all adhere to the same general principle: the feature being classed is independent of other features, therefore, its presence or absence has no bearing on the presence or absence of other features. Multinomial Naïve Bayes was one of the best algorithms for this problem in our tests, with an accuracy of 67%.

(vi) **Random Forest Classifier**
The Random Forest Classifier was used in our model to determine if Kannada sentences were sarcastic or not. During training, several decision trees are built using this ensemble learning technique, which then outputs the class that represents the average of the individual trees' predictions. The algorithm lowers the chance of overfitting and manages non-linear interactions well. In our tests, the Random Forest Classifier obtained an accuracy of 64.04%, despite its robustness, suggesting that it may be improved upon for the complex task of sarcasm detection.

(vii) **XGBoost**
Extreme Gradient Boosting, or XGBoost, was used in our model as a classifier to identify sarcasm in Kannada texts. Based on gradient boosting, XGBoost is a potent ensemble machine learning method renowned for its effectiveness, scalability, and capacity to manage intricate data relationships. To avoid overfitting, it optimizes performance via regularization approaches and makes use of decision tree-based learners. With an accuracy of 63.85%, XGBoost showed competitive performance in our tests, indicating its potential to handle the complex sarcastic patterns seen in Kannada language datasets [2].

(viii) **Bert Algorithm**
BERT (Bidirectional Encoder Representations from Transformers) [13], a sophisticated algorithm in Natural Language Processing (NLP), is founded on deep contextual learning techniques. BERT is very successful at text categorization problems because To comprehend the contextual relationships between words in a sentence, it employs a transformer architecture. The approach ensures a thorough comprehension of the text by processing input sentences in both directions. In this study, BERT was optimized to identify sarcasm in Kannada words with a 64.95% accuracy rate. When compared to conventional methods, its capacity to pick up on small contextual clues greatly improves classification results.

When training the BERT model, A complete iteration across the training dataset is called an epoch, during which the

model learns from every data point once to learn patterns and update its weights using back propagation [3]. In this work, the BERT model was trained for five epochs as shown in Fig. 2.

Epoch 1: The model began with a training loss of 0.35515, which gradually decreased as it learned the data patterns.
Epoch 2: The training loss further reduced to 0.32421, indicating improved learning and weight adjustment; and.
Epoch 3: The training loss further reduced to 0.31792, indicating improved learning and weight adjustment; and.
Epoch 4: The training loss further reduced to 0.31587, indicating improved learning and weight adjustment; and
Epoch 5: By the final epoch, the training loss further decreased to 0.314911, indicating improved learning and weight adjustment, indicating the model's convergence toward an optimal solution in Kannada sentences, attaining an accuracy of 64.93%. Its capacity to capture subtle contextual cues greatly improves classification performance when compared to traditional solutions.

The gradual reduction in training loss across epochs demonstrates how the model becomes better at predicting labels by refining its understanding of the dataset over multiple iterations [2].

Results and Discussions

The sarcasm detection system for the Kannada language was evaluated using a separate test dataset, consisting of 7000 + samples containing sarcastic as well as non-sarcastic sentences. The system's accuracy, precision, recall, and F1-score were calculated using standard evaluation metrics. The sentences were classified as True or False, falling into two categories: sarcastic and non-sarcastic. The dataset used in this research is organized in CSV format, with two main columns: sentences and sarcastic. The sentences column contains data collected from various sources, while the sarcastic column serves as the target label. A sample of the dataset is shown in Table 1, with its English translation provided in Table 2 for reference. This dataset includes a wide range of topics, such as politics, sports, and education, which ensures variety in content and helps the model generalize across different subjects [15]. To support training and evaluation, the dataset is split into two parts, with 80% used for training and 20% for testing. Samples of Kannada sentences appear in Table 1.

```
Epoch 1 - Average training loss: 0.3351565209919946
Epoch 2 - Average training loss: 0.3242166399825038
Epoch 3 - Average training loss: 0.3179270009461202
Epoch 4 - Average training loss: 0.3158781914726684
Epoch 5 - Average training loss: 0.31491179738128394
BERT Accuracy: 64.93%
              precision    recall  f1-score   support

           0       0.66      0.64      0.65      3677
           1       0.64      0.66      0.65      3606

    accuracy                           0.65      7283
   macro avg       0.65      0.65      0.65      7283
weighted avg       0.65      0.65      0.65      7283
```

Fig. 2 BERT model showing average training loss over five epochs and accuracy

Table 1 Few Kannada sentences from dataset

Sl. no	Sentences	Sarcastic
1	ಈ ಕೆಲಸ ತುಂಬಾ ತೃಪ್ತಿಕರವಾಗಿದೆ. ನನ್ನ ಹೆಚ್ಚಿನ ಸಮಯವನ್ನು ಕನಿಷ್ಠ ವೇತನಕ್ಕೆ ಕೆಲಸ ಮಾಡುವುದರಲ್ಲಿ ಕಳೆಯಲು ನಾನು ರೋಮಾಂಚನಗೊಂಡಿದ್ದೇನೆ.	TRUE
2	ಮಾನವನ ಮೆದುಳು ಸುಮಾರು 3 ಪೌಂಡ ತೂಗುತ್ತದೆ	FALSE
3	ಅರೆಕಾಲಿಕ ಕೆಲಸ ಮಾಡುವುದು ಮತ್ತು ಪೂರ್ಣಾವಧಿ ತರಗತಿಗಳಿಗೆ ಹಾಜರಾಗುವುದು ಪ್ರತಿಯೊಬ್ಬ ವಿದ್ಯಾರ್ಥಿಯ ಜೀವನೋಪಾಯವನ್ನು ಪೂರೈಸುವ ಕನಸಾಗಿದೆ.	TRUE
4	ನಾನು ಎಂದಿಗೂ ಬಳಸದ ಕರಕುಶಲ ಸಾಮಗ್ರಿಗಳಿಗೆ ನನ್ನ ಎಲ್ಲಾ ಹಣವನ್ನು ಖರ್ಚು ಮಾಡಲು ಇಷ್ಟಪಡುತ್ತೇನೆ.	TRUE
5	ಓಹ್, ದಯವಿಟ್ಟು ಎಲ್ಲಾ ದೊಡ್ಡಕ್ಷರಗಳನ್ನು ಟೈಪ್ ಮಾಡುವುದನ್ನು ಮುಂದುವರಿಸಿ. ಇದು ನಿಜವಾಗಿಯೂ ನಿಮ್ಮ ವಾದದ ವಿಶ್ವಾಸಾರ್ಹತೆಯನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ.	TRUE
6	ನಮ್ಮ ಜೀವನದ ಎಲ್ಲಾ ಒಳ್ಳೆಯ ವಿಷಯಗಳನ್ನು ಪರಿಶಂಸಿಸಲು ಸ್ವಲ್ಪ ಸಮಯ ತೆಗೆದುಕೊಳ್ಳಿ	FALSE
7	ಉದಯೋಗಗಳು ಸೃಜನಶೀಲತೆ ಮತ್ತು ನಾವೀನ್ಯತೆಗೆ ವೇದಿಕೆಯನ್ನು ಒದಗಿಸುತ್ತವೆ	FALSE
8	ಅಮೆಜಾನ್ ಮಳೆಕಾಡು ವಿಶ್ವದ ಅತಿದೊಡ್ಡ ಉಷ್ಣವಲಯದ ಮಳೆಕಾಡು.	FALSE
9	ಈ ಕೆಲಸವನ್ನು ಹೇಗೆ ಮಾಡಬೇಕೆಂದು ದಯವಿಟ್ಟು ನನಗೆ ತೋರಿಸಬಲ್ಲಿರಾ?	FALSE
10	ಮನೆಗೆ ಬಂದ ತಕ್ಷಣ ಅಡುಗೆ ಅನಿಲ ಹಚ್ಚಲು ನಾಲ್ಕು ಕೈಗಳು ಬೇಕಾಗುತ್ತವೆ	TRUE
11	ಪರಮಾಣೀಕೃತ ಪರೀಕ್ಷೆಗಳಲ್ಲಿ ಉತ್ತೀರ್ಣರಾಗುವತ್ತ ನಾವು ಗಮನಹರಿಸಬಹುದಾದಾಗ, ಯಾರಿಗೆ ಉತ್ತಮ ಶಿಕ್ಷಣ ಬೇಕು?	TRUE
12	ನೀವು ಎಂದಾದರೂ ಒಂದು ಕಪ್ ಕಾಫಿ ಕುಡಿಯಲು ಬಯಸುತ್ತೀರಾ?	FALSE

The results of each intermediate categorization stage are describes as follows.

The dataset comprises 7000 Kannada sentences, each labeled with sarcasm indicators in the Sarcastic column. The inputs (x) are the processed Kannada sentences after

Table 2 English translation of Kannada sentences of Table 1

Sl. no	Sentences	Sarcastic
1	This job is very satisfying. I am thrilled to spend most of my time working for minimum wage.	TRUE
2	The human brain weighs about 3 pounds.	FALSE
3	Working part time and attending classes fullt ime is every student's dream to make ends meet.	TRUE
4	I love spending all my money on craft supplies that I never actually use	TRUE
5	Oh, please continue typing in all caps. This really increases the credibility of your argument	TRUE
6	Let's take a moment to appreciate all the good things in our lives.	FALSE
7	Jobs provide a platform for creativity and innovation.	FALSE
8	The Amazon Rainforest is the largest tropical rainforest in the world.	FALSE
9	Can you please show me how to do this task?	FALSE
10	It takes four hands to put on the cooking gas as soon as you get home.	TRUE
11	Who needs a well rounded education when we can focus on passing standardized tests?	TRUE
12	Do you ever want to grab a cup of coffee?	FALSE

cleaning and stemming, while the outputs (y) are their corresponding sarcasm labels. For training and evaluation, the dataset is divided into 80% for training and 20% for testing using the train_test_split function. This results in a training set containing 5600 sentences and a testing set comprising 1400 sentences. The training set is utilized to train multiple machine learning and deep learning models, while the testing set is reserved for evaluating the performance and generalization capability of these models on unseen data. This split ensures a balanced approach to model training and evaluation, maintaining the integrity of the results [13].

Sentences are divided into specific tokens or words during tokenization. The data cleaning method uses these tokens as input. For instance, "ಮನೀಶ ತಮ್ಮ ಪಾತರವನನು ಚೆನನಾಗಿ ನೆರವಹಿಸದದಾರೆ" (Manish performed his role well.). Tokens such as " ಮನೀಶ "ತಮ್ಮ" " ಪಾತರವನನು" " ಚೆನನಾಗಿ" "ನೆರವಹಿಸದದಾರೆ ". The method of data cleaning includes eliminating extraneousinformation, like punctuation, that isn't useful for sentiment analysis. Punctuation such as commas, full stops, and dollars is eliminated in the example above. Words classified as stop words are those that add no meaning to the phrase. Eliminating stop words reduces the amount of the dataset, which shortens the training time without compromising system accuracy. The process of eliminating a word's suffix to reduce it to its basic term is known as stemming [3]. The term "ಪಾತರವನನು" is shortened to "ಪಾತರ" in the first sentence of the example above [1].

An illustration of a custom input is presented in Fig. 3, where users can manually enter Kannada sentences and have them classified as either sarcastic or non-sarcastic. The text is shown in Figs. 4 and 5 following the cleaning and stemming procedures, which are crucial pipeline pretreatment stages. The classification outcomes by incorporating different machine learning classifiers are displayed in Fig. 6. Along with the BERT model for sequence classification, we experimented with several classifiers in our model, including models such as Linear SVC, Logistic Regression, SGD Classifier, SVC, Multinomial Naive Bayes, Random Forest, and XGBoost. Each classifier used its trained features to estimate whether the input Kannada line was sarcastic or not. For instance, the BERT model, which was trained on

```
Linear SVC
LogisticRegression
SGDClassifier
SVC
MultinomialNB
RandomForestClassifier
XGBoost
[True, True, True, True, True, True, True]
BERT Prediction: 1
```

Fig. 6 Boolean result of every ML model and result of BERT algorithm

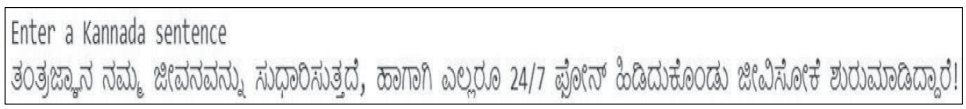

Fig. 3 Screenshot of a Kannada sentence taken as input from user

(English Translation : Technology has improved our lives, so everyone has started living with a phone 24/7)

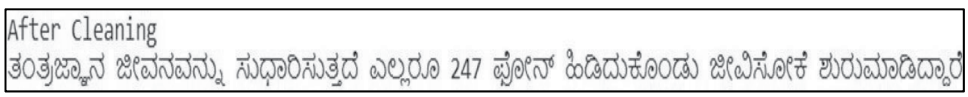

Fig. 4 Screenshot after cleaning of the sentence

(English Translation : Technology has improved our lives so everyone has started living with a phone 24/7)

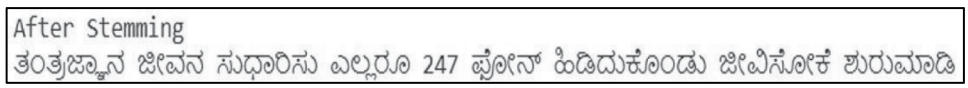

Fig. 5 Screenshot after cleaning of the sentence

(English Translation :Technology has improved our live so everyone has started live with a phone 24/ 7)

Kannada sentences, identified the sarcastic statements with an accuracy of 64.93%.

Comparative Analysis of Algorithms

To classify Kannada sentences from a large dataset, we employed the BERT algorithm alongside machine learning classifiers such as Logistic Regression, SGD Classifier, SVC, Multinomial Naive Bayes, Gaussian Naive Bayes, and Random Forest. As discussed in earlier sections, the Kannada sentences undergo preprocessing before classification. This classification step plays an important role in sentiment analysis, where each sentence is categorized as either sarcastic or non-sarcastic.

Performance Measures

Among the evaluated machine learning algorithms, the proposed Multinomial Naive Bayes method demonstrated superior performance, achieving an accuracy of 67% on a big dataset, based on the performance metrics we employed to evaluate the various algorithms' efficacy.

Precision

Proportion of accurate forecasts. The number of accurate predictions for actual data is shown by this algorithmic parameter. As per the Eq. (1). It is expressed as the percentage of true positive (TP) results out of the total of true positives and false positives (TP+FP).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

Recall

Proportion of positive cases. The algorithm's recall parameter shows the positive examples. As per the Eq. (2). The percentage of genuine positive results is expressed using the total of true negatives (TN) and false negatives (FN) [5].

$$\frac{TP}{TP + FN} = \text{Recall} \tag{2}$$

Table 3 Accuracy of classifiers

Sl. no.	Classifier	Accuracy
1	Linear SVC	66.45%
2	**Logistic Regression**	**67.4%**
3	SGD Classifier	66.78%
4	SVC	65.68%
5	Multinomial NB	67%
6	Random Forest Classifier	64.04%
7	XGBoost	63.85%
8	BERT	64.93%

F1 score

It shows the percentage of accurate positive predictions. In mathematical terms, the F1 score is a valued harmonic mean that is used to determine the prediction's average ratio. High data will receive the best score of 1.0 in comparison to poor data. The F1 score can't be used to quantify accuracy; instead, it's typically used to distinguish amongst classifier models [6].

Support

Support refers to the number of instances of each class present in the dataset. Stratified sampling or rebalancing may be necessary if the training data is imbalanced, which can lead to a structural weakness in the classifiers' output scores. However, during evaluation, this doesn't affect the results, as the level of support remains consistent across different models.

Accuracy is determined by adding true positive values to true negative values, then dividing the result by the number of samples from a document. This calculation is acceptable when the model is balanced and inappropriate when there is a class imbalance.

Macro Average.

The average of class precisions without taking proportion into account is called the macro average.

Accuracy 0 = X.
Accuracy 1 = Y.

$$\frac{(X + Y)}{2} = \text{MacroAveragePresision} \tag{3}$$

The weighted average takes the proportion into account.
Precision for class 0: 0.68
Precision for class 1: 0.66
Support for class 0: 749
Support for class 1: 708
Total samples: 1457

$$\text{WeightedAveragePresision} = (0.68 \times 0.514) + (0.66 \times 0.486) = 0.6703 \tag{4}$$

Performance Analysis of Algorithms

The accuracy comparison of all the algorithms utilized in our model is shown in Fig. 6, and Table 3 displays the calculated accuracy of various techniques across the dataset. Seven thousand Kannada sentences were used to train our model. Training and testing sets were separated from the dataset, with testing data making up about 20% of the total. With an accuracy of 67.4%, Logistic Regression outperformed all other classifiers, according to the comparison. Accuracy was 67% for Multinomial NB, 66.78% for SGD

Classifier, and 66.45% for Linear SVC. The Random Forest Classifier reached 64.04% accuracy, whereas the SVC reached 65.68%. Bert's accuracy rate was 64.93%.

Figure 7; Table 4 compare various algorithms for the Kannada sarcasm detection model. The values of true positives, false positives, true negatives, and false negatives are used to compute metrics such as precision, recall, and F1-score. The following scenarios are then used to evaluate the results.

- It is regarded as a True Negative (TN) if the input Kannada sentence is deemed non-sarcastic and the anticipated outcome is likewise non-sarcastic. Negatives.
- A True Positive (TP) represents when the input Kannada sentence is deemed sarcastic and the anticipated outcome is likewise sarcastic.
- A False Positive (FP) represents when the intended outcome is non-sarcastic, but the input Kannada sentence is categorized as sarcastic.
- When the projected outcome is sardonic but the input Kannada language is categorized as non-sarcastic, this is known as a False Negative (FN).

These metrics help measure the performance of the sarcasm detection model in accurately identifying sarcastic and non-sarcastic sentences in Kannada. The bold values shown in Tables 3 and 4 indicate the best-performing classifier based on accuracy.

Comparison with Existing Work

Table 5 illustrates the comparison of the proposed Kannada sarcasm detection model with existing models. A number of the examined papers include models specifically designed for Kannada. For example [7] and [8], concentrate on sarcasm detection in Kannada, with [2, 3, 5] and [10] employing hybrid techniques. Models that use a combination of deep learning and machine learning to classify sarcasm in English sentences have shown varying results.

While models like [7, 8] obtained respectable accuracies when compared to current models specifically for Kannada, their performance was constrained by simpler approaches and smaller datasets In contrast, the proposed model was evaluated on a larger dataset of 7,000 Kannada sentences using eight different algorithms, including Linear SVC, Logistic Regression, Stochastic Gradient Descent Classifier, Support Vector Classifier, Multinomial Naive Bayes, Random Forest, XGBoost, and BERT.

Confusion Matrix Using Logistic Regression

For the task of sarcasm detection in regional languages, fine-tuned Logistic Regression models were employed. After model inference, the confusion matrix was generated by comparing the predicted labels with the true labels. The binary classification results are illustrated through the confusion matrix presented in Fig. 8. Although the model achieved reasonable overall performance, misclassifications were still present, suggesting limitations in capturing subtle contextual nuances. On the given dataset, the Logistic Regression classifier attained an overall accuracy of 67.4%.

Conclusion

This study uses Natural Language Processing (NLP) techniques to develop a machine learning-based approach for detecting sarcasm in Kannada texts. Every day, enormous volumes of text data are produced due to the exponential rise of social media and internet usage. Gaining corporate

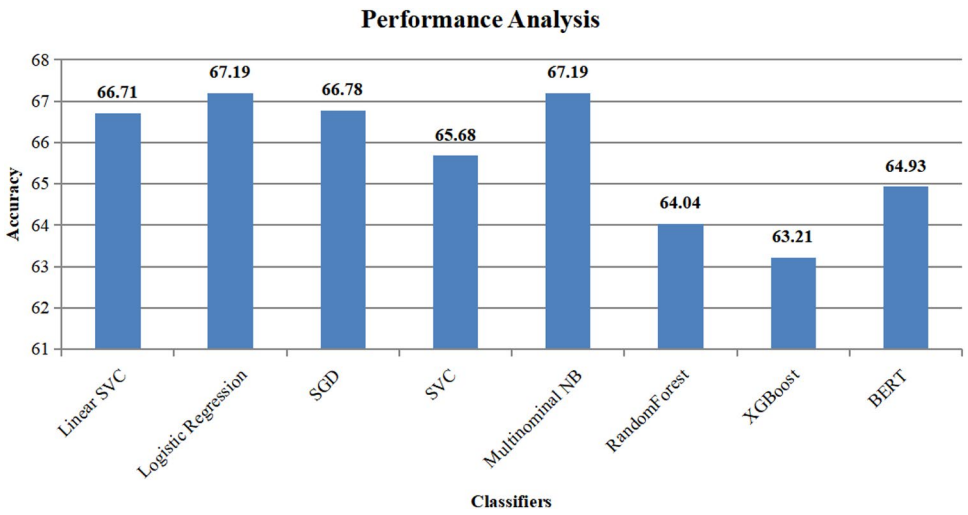

Fig. 7 Performance evaluation of machine learning algorithms applied to a dataset comprising Kannada text

Table 4 Classification report of classifiers

	Classifiers	Performance evaluation report				
			Precision	Recall	F1-score	Support
1	Linear support vector classifier	0	0.68	0.65	0.67	749
		1	0.65	0.68	0.66	708
		Accuracy	–	–	0.67	1457
		Macro avg	0.67	0.67	0.67	1457
		Weighted avg	0.67	0.67	0.67	1457
2	Logistic regression classifier	**0**	**0.70**	**0.64**	**0.67**	**749**
		1	**0.65**	**0.71**	**0.68**	**708**
		Accuracy	–	**--**	**0.67**	**1457**
		Macro avg	**0.68**	**0.68**	**0.67**	**1457**
		Weighted avg	**0.68**	**0.67**	**0.67**	**1457**
3	Stochastic gradient descent classifier	0	0.69	0.66	0.68	749
		1	0.66	0.68	0.67	708
		Accuracy	–	–	0.67	1457
		Macro avg	0.67	0.67	0.67	1457
		Weighted avg	0.67	0.67	0.67	1457
4	Support vector classifier	0	0.68	0.63	0.65	749
		1	0.64	0.69	0.66	708
		Accuracy	–	–	0.66	1457
		Macro avg	0.66	0.66	0.66	1457
		Weighted avg	0.66	0.66	0.66	1457
5	Multinomial naive bayes classifier	0	0.68	0.67	0.68	749
		1	0.66	0.67	0.66	708
		Accuracy	–	–	0.67	1457
		Macro avg	0.67	0.67	0.67	1457
		Weighted avg	0.67	0.67	0.67	1457
6	Random forest classifier	0	0.65	0.63	0.64	749
		1	0.62	0.64	0.63	708
		Accuracy	–	–	0.64	1457
		Macro avg	0.64	0.64	0.64	1457
		Weighted avg	0.64	0.64	0.64	1457
7	XGBoost	0	0.67	0.58	0.63	749
		1	0.61	0.70	0.65	708
		Accuracy	–	–	0.64	1457
		Macro avg	0.64	0.64	0.64	1457
		Weightedavg	0.64	0.64	0.64	1457
8	BERT	0	0.66	0.64	0.65	3677
		1	0.64	0.66	0.65	3606
		Accuracy	–	–	0.65	7283
		Macro avg	0.65	0.65	0.65	7283
		Weighted avg	0.65	0.65	0.65	7283

insights, understanding user perspectives, and speeding up decision-making processes all depend on enhancing sentiment analysis accuracy, which requires an understanding of sarcasm. There are several sentiment analysis models for English, but there aren't many tools or models made especially for Kannada sarcasm detection. We suggested an effective Kannada sarcasm detection model that makes use of several classification techniques to close this gap. We selected and manually annotated a more extensive dataset of 7,000 Kannada texts, classifying them into sardonic and non-sarcastic labels, in contrast to earlier research that usually depended on datasets of 2,500–3,000 words. Proposed research offers a more solid basis for real-world sarcasm detection applications, even if larger datasets brought more diversity and complexity, which decreased accuracy when compared to smaller datasets. Tokenization, data cleaning, stop word removal, and stemming are examples of preprocessing procedures that were essential for enhancing model performance. We used TF-IDF to quantitatively represent the texts for feature extraction. A variety of classification techniques, A diverse set of algorithms such as Logistic Regression, Linear SVC, Support Vector Classifier (SVC),

Table 5 A comparative analysis of the results from the proposed model with those of other existing models

Year	Language	Dataset	Method	Accuracy
2024 [8]	Kannada	Voicesamples and text data	Random Forest, RNN with LSTM units, and CNNs; voice features processed with MFCCs and text with n-grams and sentiment features.	57.2%
2023 [7]	Kannada	Limited dataset	CNN	76%
2023 [2]	English	Multiple dataset	Word2Vec, GloVe, and BERT	90%
2023 [3]	English	–	LSTM, Word2Vec	84.88%
2021 [5]	English	28,503 Headlines	LSTM, BERT	F1-score 0.90
2021 [10]	English	Limted dataset	Deep Learning model	F1-score 0.76
Proposed model	Kannada	7000 samples	Logistic Regression and Multinomial Naive Bayes Classifier has performed better with large dataset	F1-score of 0.653

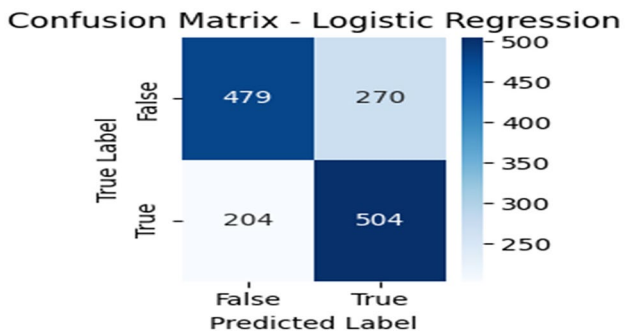

Fig. 8 Confusion matrix using the Logistic Regression classifier

K-Neighbors Classifier, Multinomial Naive Bayes, Gaussian Naive Bayes, Random Forest Classifier, BERT, and Stochastic Gradient Descent Classifier (SGD) were employed to train the models. Among them, Logistic Regression had an accuracy of 67.4%, demonstrating the promise of conventional methods when used with a carefully chosen dataset. The proposed work addresses critical gaps in Kannada NLP by employing diverse datasets and multiple classifiers, thereby establishing a foundation for future research in regional language processing.

One of the main challenges in this study is how difficult it is to detect sarcasm in the Kannada language. Sarcasm often depends on small hints in the way something is said, the cultural background, or the tone all of which are hard to understand from written text alone. Since Kannada is a low-resource language, there aren't many labeled datasets or advanced tools available to help with this task. This makes it even harder to accurately identify sarcasm. Even though we tried to maintain good annotation quality by using multiple people and majority voting, sarcasm is still very subjective different people may understand the same sentence in different ways. This can affect how well the model performs.

Authors Contributions Both the authors play a vital role in the research of proposed work. Santosh Chinchali: This author prepared data set and implemented the Sarcasm detection and classification on Kannada language. Prepared initial draft of manuscript. Pushpa B. Patil: This author performed the validation and analysis of results. Reviewed and edited the manuscript. Both the authors read and approved the final version of the manuscript.

Funding Authors acknowledges Vision Group on Science and Technology (VGST) Govt. of Karnataka (Grant No.ECRA/GRD NO.1243/2023-24).

Data Availability The data supporting this study are available from the corresponding author upon reasonable request.

Declarations

Competing Interests The authors declare that they have no conflict of interest.

Research Involving Human and/or Animals Not applicable.

Informed Consent Not applicable.

References

1. Hande A, Priyadharshini R, Sampath A, Thamburaj KP, Chandran P, Chakravarthi BR. Hope speech detection in under-resourced Kannada Language. ArXiv Preprint arXiv:2108.04616 (2021).
2. Sharma DK, Singh B, Agarwal S, Pachauri N, Alhussan AA, Abdallah HA. Sarcasm detection over social media platforms using a hybrid ensemble model with fuzzy logic. Electronics. 2023;12(4):937. https://doi.org/10.3390/electronics12040937.
3. Misra R, Arora P. Sarcasm detection using a news headlines dataset. AI Open. 2023;4:13–8.
4. Savini E, Caragea C. Intermediate-task transfer learning with BERT for sarcasm detection. Mathematics. 2022;10(5):844. https://doi.org/10.3390/math10050844.
5. Scola E, Segura-Bedmar I. Sarcasm detection with BERT. Procesamiento Del Lenguaje Nat. 2021;67:13–25. https://doi.org/10.26342/2021-67-1.
6. Ranjitha P, Bhanu KN. Improved sentiment analysis for Dravidian language-Kannada using decision tree algorithm with efficient data dictionary. IOP Conf Ser Mater Sci Eng. 2021;1123(1):012039. https://doi.org/10.1088/1757-899X/1123/1/012039.
7. Manohar R, Swamy S. Textual data analysis for identifying sarcasm in Kannada. J Intell Fuzzy Syst. 2023;44(6):1001–4055.
8. Swamy RM. Voice-based sarcasm detection in Kannada Language. Int J Intell Syst Appl Eng. 2024;12(14s):356–67.
9. Bharti SK, Naidu R, Babu KS. Hyperbolic feature-based sarcasm detection in Telugu conversation sentences. J Intell Syst. 2021;30(1):73–89. https://doi.org/10.1515/jisys-2018-0475.

10. Akula R, Garibay I. Interpretable multi-head self-attention architecture for sarcasm detection in social media. Entropy. 2021;23:394. https://doi.org/10.3390/e23040394.
11. Akula R, Garibay I. Explainable detection of sarcasm in social media. In: Proc. 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 34–39. Association for Computational Linguistics (2021).
12. Moores B, Mago VK. A survey on automated sarcasm detection on Twitter. ArXiv Preprint arXiv:220202516 (2022).
13. Ghosh D, Fabbri AR, Muresan S. Sarcasm analysis using conversation context. Comput Linguist. 2018;44(4):755–92. https://doi.org/10.1162/coli_a_00336.
14. Patil PB, Ijeri D, Kulkarni SA, et al. Comparative study of machine learning algorithms for Kannada Twitter sentimental analysis. Multimed Tools Appl. 2024;83:45693–713.
15. Ijeri D, Patil PB. Leveraging sequence-to-sequence models for Kannada abstractive summarization. SN Comput Sci. 2025;6:527.