**SARCASM DETECTION AND CLASSIFICATION ON REGIONAL LANGUAGE USING MACHINE LEARNING TECHNIQUES**

**Santosh Chinchali1. Ruchita Atharga1. Sachinkumar Madarakhandi1.**

**Shreya Dixit1. Shyamala Wathare1**

## ABSTRACT

Sarcasm is a term used in language where the statement's intended meaning is contradicted. It is often used in everyday speech, particularly on social media platforms, and is often difficult to define exactly. Accurately identifying sarcasm can be crucial for a variety of applications, such as attitude analysis, opinion mining, and social media monitoring. Sarcasm detection in regional languages, such as Kannada, is particularly challenging due to the lack of trustworthy natural language processing techniques and resources. Sarcasm detection is an important area of research in natural language processing (NLP) that aims to automatically identify sardonic comments in text. A premium dataset of 7,000 lines was selected for this study, encompassing a variety of subjects and situations that are pertinent to sarcasm. This dataset is much larger than the datasets utilized in earlier studies, which had only about 2,500 sentences. Although previous research reached up to 85% accuracy, it was on smaller datasets, raising questions about robustness and generalizability. On the other hand, our model demonstrated its capacity to handle more vast and diversified data by achieving a competitive accuracy of 67% on the larger dataset. A number of text mining techniques, such as tokenization, stop word removal, stemming, and parts of speech (POS) labeling, are used to pre-process the gathered dataset. Deep learning models and machine learning algorithms are then trained using the processed data. SGD Classifier (66.78%), SVC (65.68%), Multinomial NB (67.19%), Random Forest Classifier (64.04%), Linear SVC (66.71%), Logistic Regression (67.19%), XGBoost (63.21%), and BERT (64.93%) are the performance metrics for the various algorithms. These findings show the difficulties in detecting sarcasm in Kannada, but they also point to the possibility of advancement through more study. More accurate and dependable models that can capture the subtleties and complexities of Kannada sarcasm may result from expanding and diversifying annotated datasets and using cutting-edge deep learning methods like transformer models or recurrent neural networks (RNNs).

**1 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. One significant issue in the study of natural language processing (NLP) is the identification of sarcasm in text. There are several possible uses for sarcasm detection, such as opinion mining, sentiment analysis, and social media monitoring. The detection of sarcasm is a challenging endeavor that necessitates knowledge of the text's language and social context. Irony, exaggeration, understatement, and rhetorical inquiries are some of the ways that sarcasm can be communicated. Additionally, it can be expressed through the text's tone, which can be challenging to communicate in writing.

Sarcasm identification has advanced significantly as a result of recent developments in machine learning and natural language processing. A lot of research has gone into creating machine learning models that can recognize sarcastic patterns and characteristics on their own. These models frequently employ supervised learning approaches, in which a dataset of labeled sarcastic and non-sarcastic text is used to train the machine learning algorithm. The model can be used to determine if fresh text is sarcastic or not after it has been trained.   
For sarcasm detection, researchers have employed rule-based approaches in addition to machine learning techniques. Using a collection of heuristics or rules, rule-based approaches identify sardonic material by analyzing its linguistic characteristics. For instance, a rule-based approach might classify as sarcastic sentences that exhibit a high degree of sentiment polarity or that contain terms with opposing meanings. Even with these developments, sarcasm detection is still difficult, especially when dealing with non-English languages.

Sarcasm can be hard to spot in many languages due to their distinctive linguistic traits and cultural variances. Furthermore, sarcasm can be quite context-dependent, which means that depending on the circumstance, the same text may be regarded as either sarcastic or non-sarcastic. Furthermore, geographical and cultural differences can have an impact on Kannada sarcasm. Urban and rural settings may employ sarcasm differently, and the cultural context of sarcasm might change based on the social standing, age, and gender of the speaker. It is difficult to create a thorough sarcasm detection system that can reliably detect sarcasm across various geographies and demographic groups because of these variances.

### Related Work

Hande et.al. [1] has created 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, is very lacking. Therefore, we propose to create **the KanHope English-Kannada** Hope speech dataset and compare its findings with those of other studies. 6,176 user-generated comments in a combination of Kannada and code were extracted from YouTube and meticulously categorized as either hope speech or not. Furthermore, they introduced DC-BERT4HOPE, a dual-channel model that uses the English version of KanHope to further train to support hope speech identification with a weighted F1-score of 0.756, the approach performs better than alternative models. KanHope promoted Kannada study while urging all academics to handle internet research pragmatically. They have achieved higher accuracy because they have used Kannada-English combination whereas we have used purely Kannada sentences , hence accuracy achieved by our work is lesser but efficient.

Ranjitha P and Bhanu K N [2] In order to ascertain whether the author of a piece of Kannada-language writing has 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 sentiment analysis** is used to achieve this. Websites such as Prajwani, One India News, and Wedunia are also part of the training data set. 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 , therefore achieving 67.12% accuracy for dataset of 7000 sentences is efficient than achieving 85% accuracy in sentiment analysis for lesser dataset i.e.,1000 sentences.

Manohar R1 , Suma Swamy2 [3] In order to detect sarcasm in Kannada, this study uses a hybrid methodology that blends more traditional machine learning techniques like Support Vector Machines (SVM) and Random Forests with deep learning techniques like Bidirectional Long Short-Term Memory (BiLSTM). Several preprocessing methods were employed in the analysis of Kannada textual data, including tokenization, stemming, and stop-word removal. 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 recommend including audio to enhance the model's sarcasm recognition abilities. Our work's emphasis on sarcasm in Kannada aligns closely with the problem scope, avoiding the distractions of generic linguistic tasks. Incorporating modern NLP techniques and focusing on data-driven approaches might yield better performance and more scalable results, even if the initial accuracy is lower (67.12%).

B.R. Shambhavi2, Rajani Shree M1, and [4] Two distinct methods for Part-of-Speech (POS) tagging in Kannada text are investigated in this work. The first strategy makes use of Conditional Random Field (CRF++), a supervised machine learning technique, while the second strategy blends deep learning methods with word embedding. Tokenization, cleaning, and tagging according to the BIS (Bureau of Indian Standards) tagset were preprocessing processes. 1200 Kannada texts from the agriculture domain made up the dataset; 1100 were used for training, and 100 were used for testing. While the deep learning method using Word2Vec achieved 71% accuracy, the CRF++ model achieved 76.45%. The tiny dataset size and the intricacy of Kannada's agglutinative nature were among the constraints noted by the authors. For increased accuracy, future research recommends using Keras and growing the dataset. Used a dataset of only **1,200 samples** (1,100 for training and 100 for testing), which is significantly smaller than your **7,000-line dataset**. The highest accuracy achieved in this work was **76.45%** using CRF++, and **71%** with deep learning. While our accuracy (67.12%) is lower, it is for a more complex task (sarcasm detection) and across a larger, more varied dataset, indicating a higher challenge and potential for improvement.

Manohar R1, Suma Swamy2 [5] With an emphasis 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 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 combine deep learning architectures like Long Short-Term Memory networks (LSTMs) with machine learning techniques like Support Vector Machines (SVM) are all part of the methodology. The suggested model's initial accuracy of 57.2% is regarded as a starting point for additional study. This work reports an initial accuracy of **57.2%**, which is significantly lower than our achieved accuracy of **67.19%** for text-based sarcasm detection. This indicates that their approach is still in early development and less reliable.

Bharti et.al. [6] When 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. For sarcastic sentiment identification, the existing systems support only English-scripted text. Scholars have become more interested in low-resource languages including Hindi, Telugu, Tamil, Arabic, Chinese, Dutch, Indonesian, and others in recent years. 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. In this post, They have collected and annotated a corpus of Telugu talk. Sarcasm can be identified by phrases that begin with an inquiry and conclude with an answer. Along with a set of algorithms, the investigation of sarcasm in a corpus of Telugu conversation phrases is also suggested. Four hyperbolic features—interjection, intensifier, question mark, and exclamation point—form the foundation of the proposed algorithms. 94% accuracy was attained. While this report **94% accuracy**, 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. Our work, with a more realistic **67.12% accuracy**, reflects the complexities of sarcasm detection across varied contexts.

Akula And Garibay [7] have created a linguistic concept called sarcasm, which is commonly used to express the opposite 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 text messages from different online media and social networking sites. They accomplish this by developing an interpretable deep learning model using gated recurrent units and multi-head self-attention. They show the effectiveness of their approach by achieving state-of-the-art results on multiple datasets from online media and social networks. The models created with their proposed approach are easy to comprehend and enable the identification of sarcastic cues in the input text that influence the classification outcome. They use a few representative input texts to illustrate the learned attention weights in order to show the effectiveness and interpretability of their method.

Akula and Garibay [8] To illustrate the effectiveness and interpretability of there approach, They obtain state-of-the-art results on datasets from online discussion forums, social networking sites, and political discussions. Used when having online conversations. The need for manually generated characteristics has been eliminated by recent studies that have employed deep learning to use neural networks to learn both lexical and contextual features (Ghosh and Veale, 2017; Ilic et al., 2018; Ghosh et al., 2018; Xiong et al., 2019; Liu et al., 2019). Deep convolutional, recurrent, or attention-based neural networks are trained using word embeddings in these works to generate state-of-the-art results. Deep learning-based methods are quite effective, however they are not interpretable. An important aspect of this study is interpretability, which was in addition to the model's excellent performance. The main contributions of their work are as follows: a) They suggest a model that can be comprehended for self-attention sarcasm detection. b) Through extensive testing and ablation trials, achieve state-of-the-art results on multiple datasets and show the effectiveness of our methodology. c) Show how interpretable the model is using the learned attention maps.

Moores and Mago [9] The topic of automatic sarcasm detection is expanding as computer science advances. Short text messages are increasingly being used for communication, especially on social media platforms like 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 a number of contemporary methods for sarcasm detection, including as 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.

Madan A, Ghose U [10] The special difficulties of processing sentiment in Hindi, a morphologically complex and resource-constrained language, are highlighted in this paper on sentiment analysis for Hindi-language Twitter data. Tokenization, stop-word elimination, and stemming are some of the preprocessing stages used in the methodology. In order to classify attitudes into positive, negative, and neutral categories, the authors used machine learning classifiers, such as Support Vector Machines (SVM) and Naïve Bayes. For better sentiment extraction, they made use of Hindi-specific linguistic elements. The suggested method demonstrated its efficacy in assessing feelings in Hindi tweets by achieving a noteworthy level of accuracy. The study emphasizes how sentiment analysis in regional languages can be achieved by fusing machine learning methods with linguistic resources. For improved performance, future research recommends enlarging datasets and investigating deep learning models.

Kulkarni et.al. [11] CNN, LSTM, UMFiT, and BERT base deep learning models were used in this work to create the author's current baseline classification findings. They also introduce L3CubeMahaSent, the first publicly available sizable dataset for Marathi sentiment analysis. Three groups of 16,000 unique tweets are included in the data collection. CNN achieved an accuracy of 83.24%, LSTM achieved 82.89%, UMFiT achieved 80.80%, and BERT achieved 84.13%.

Ankita Sharma, Udayan Ghose [12] In order to ascertain public opinion during the 2019 Indian general elections, this study does sentiment analysis on Twitter data. Using Twitter APIs, tweets about two candidates for prime minister, Candidate 1 and Candidate 2, were gathered between January and March 2019. Cleaning the data by eliminating stop words, URLs, and punctuation was one of the preprocessing processes. Next, R programming tools were used to create the corpus. The study further examined emotional characteristics like joy, sadness, wrath, and trust in addition to classifying tweets into positive, negative, and neutral attitudes using lexicon-based and NRC dictionary-based methodologies. The findings showed that Candidate-1 was preferred above Candidate-2, which was consistent with the May 2019 election outcomes. Incorporating multimedia data, such as music and images, and using geolocation data to obtain deeper sentiment insights are two suggested future areas.

Scola E, Segura-Bedmar I [13] Using BERT (Bidirectional Encoder Representations from Transformers), this work examines sarcasm recognition in textual data, including news headlines. In contrast to 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 sarcastic and non-sarcastic 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. In order to further improve accuracy, the study highlights the significance of contextual embeddings and recommends investigating hybrid approaches and bigger datasets in subsequent research.

Drawbacks of Existing Works:

* **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**: Sarcasm often depends on context and tone, which traditional machine learning models and even BERT may fail to fully capture, especially in ambiguous cases.
* **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 substantial computational resources, making them less feasible for real-time or low-resource 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.

**2. METHODOLOGY**

Data Collection

Logistic Regression

Classification result from all Models

MultinomialNB

Random Forest Classifier

XGBoost

Linear SVC

SVC

SGD Classifier

BERT

Removal of Stopwords

Stemming

Data Cleaning

Tokenization

**Fig. 1 Steps involved in Sarcasm detection.**

The steps involved in conducting a sentiment analysis of Kannada text taken from native speakers or everyday conversations are depicted in Fig. 1. The sentences will be taken out of this dataset. Sentences will be tokenized—that is, broken up into smaller words—after they have been retrieved. Data cleaning is a method of removing unnecessary elements from phrases that don't add any meaning to the assertion, including commas and grammar. Words like the, which, and, etc. that don't contribute sense to a statement are known as stop phrases in languages. 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 sentence is analyzed to identify whether a given sentence is sarcastic or non-sarcastic.

**2.1 DATA COLLECTION**

Data collection is the process of locating, computing, and gathering information about specific data. 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 collecting is a valuable component. To preserve the integrity of the inquiry, it is crucial to properly characterize and compile the data throughout data collecting.

**2.2 TOKENIZATION**

The process of splitting a sentence, paragraph, or text document into discrete units known as tokens is known as tokenization. Tokens might be words, phrases, keywords, or characters. We can take the example "It is a pen" as an example. The fundamental method of tokenization is to create a token by taking the space into consideration. 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 are able to tokenize sentences and documents.

**2.3 DATA CLEANING**

One crucial phase in NLP is data cleaning. Without data cleansing, the dataset is like to 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. The process of data cleaning in natural language processing (NLP) entails eliminating a variety of punctuation marks, such as commas, colons, exclamation points, hyphens, question marks, apostrophes, dashes, brackets, semicolons, brackets, brackets, brackets, ellipses, and quote marks.

**2.4 REMOVING STOP WORDS**

In any language, stop words are words or phrases that don't provide any feeling of weight to the sentence. Eliminating these stop words won't change the sentence's real meaning. 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 list of stop words to help exclude them from text processing. Stopwords common words with minimal meaning—were found and removed in order 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]. ​

**2.5 STEMMING**

The process of breaking a locution to expose the core word is known as stemming. Take the example of how the stemming algorithm reduces the words ನಡೆದ (nadeda - walked), ನಡೆಯಿತು (nadeyitu - happened), ನಡೆಯುತ್ತಿರುವ (nadeyuttiruva - happening) → ನಡ (nada - walk) The words were reduced to their basic, or root, form using a linguistic normalizing approach known as stemming. This required handling variations in verbal conjugations, nominal forms, and derivative morphemes in Kannada. Given the nuances of the language, a unique Kannada stemming algorithm was used to accurately execute stemming while maintaining the text's semantic structure [3].

**2.6 CLASSIFICATION**

As part of the categorization process, data are grouped into various groups 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, such as 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 predicts the probability of a word belonging to a specific class. ​ 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 embeddings, especially for identifying subtle patterns in Kannada sentences.

**2.6.1 TF-IDF**

The **TF-IDF (Term Frequency-Inverse Document Frequency)** method is used for feature extraction. ​ It has two components:

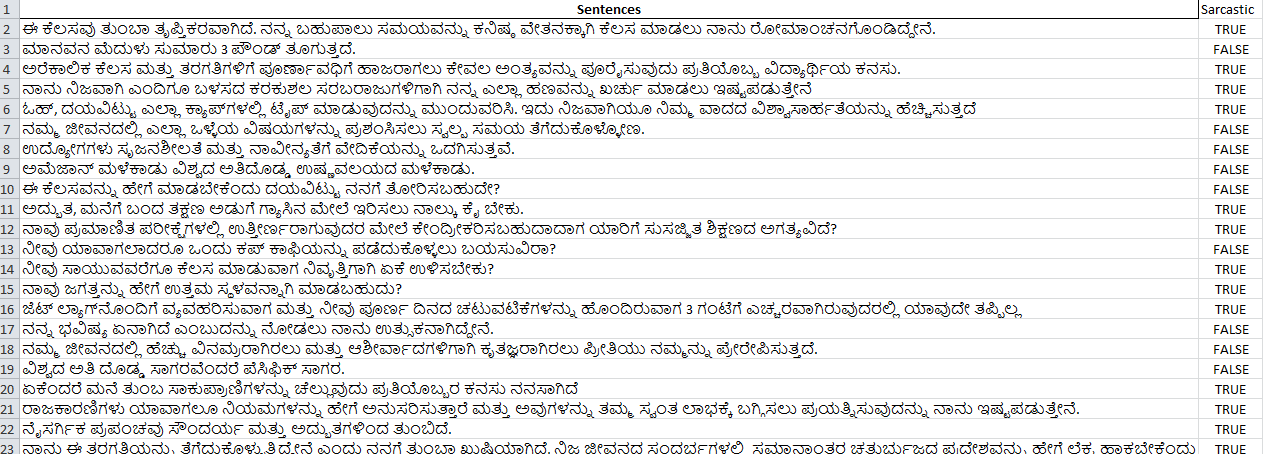
**Term Frequency (TF):** Measures how often a word appears in a document relative to the total number of words. ​

**Inverse Document Frequency (IDF):** Compares the total number of documents to the number of documents containing a specific word, reducing the importance of common words. ​

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. The technique relies on using the language that needs to be examined to train the classifiers, which necessitates having sufficient language resources. 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.

**3. RESULTS AND DISCUSSIONS**

The sarcasm detection system for Kannada language was evaluated using a separate test dataset, consisting of 7000+ instances of sarcastic and non-sarcastic sentences. The system's accuracy, precision, recall, and F1-score were calculated using standard evaluation metrics. These sentences were classified as True or False, respectively, into two classes: sardonic and non-sarcastic. These are kept in Excel format as data.



**Fig. 2 Few Kannada sentences from data collected.**

**English Translation of Kannada sentences from dataset**

2. This job is very satisfying. I am thrilled to spend most of my time working for minimum wage.

3. The human brain weighs about 3 pounds.

4.Working part-time and attending classes full-time is every student's dream to make ends meet.

5. I love spending all my money on craft supplies that I never actually use

6. Oh, please continue typing in all caps. This really increases the credibility of your argument

7. Let's take a moment to appreciate all the good things in our lives.

8. Jobs provide a platform for creativity and innovation.

9. The Amazon Rainforest is the largest tropical rainforest in the world,

10. Can you please show me how to do this task?

11. It takes four hands to put on the cooking gas as soon as you get home.

12. Who needs a well-rounded education when we can focus on passing standardized tests?

13. Do you ever want to grab a cup of coffee?

14. Why save for retirement when you can work yourself to death?

15. How can we make the world a better place?

16. There's nothing wrong with staying up at 3 a.m. when you're dealing with jet lag and

you've had a full day of activities

17. I am excited to see what my future holds.

18. Love motivates us to be more humble and grateful for blessings in our lives.

19.The largest ocean in the world is the Pacific Ocean.

20. Because having a house full of pets is everyone's dream come true

21. I love how politicians always follow the rules and try to bend them to their own advantage.

22. The natural world is full of beauty and wonder.

23. I am so glad I took this class. How to calculate the area of a parallelogram in real life situations

The results of each intermediate categorization stage are shown in the following figures.

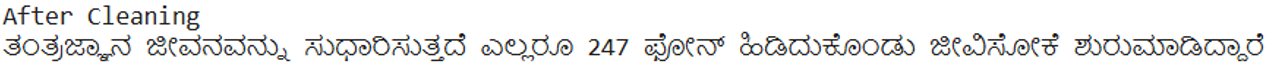
The dataset comprises 7000 Kannada sentences, each labeled with sarcasm indicators in the Sarcastic column. The inputs (X) are the processed Kannada sentences after cleaning and stemming, while the outputs (y) are their corresponding sarcasm labels. For training and evaluation, the dataset is split into 80% training data and 20% testing data 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.

Sentences are divided into specific tokens or words during tokenization. The data cleaning method uses these tokens as input. For instance, "ಮನೀಶ್ ತಮ್ಮ ಪಾತ್ರವನ್ನು ಚೆನ್ನಾಗಿ ನಿರ್ವಹಿಸದ್ದಾರೆ” (Manish did not perform his role well.). Tokens such as "ಮನೀಶ್," "ತಮ್ಮ," "ಪಾತ್ರವನ್ನು," "ಚೆನ್ನಾಗಿ," "ನಿರ್ವಹಸಸಿದ್ದಾಐೇದರೆ," "ಧಿರಯವಹಿಸಿದ್ದರೇ," ". The method of data cleaning includes eliminating extraneous information, like punctuation, that isn't useful for sentiment analysis. Punctuation such as commas, full stops, and dollars are 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 in order to reduce it to its basic term is known as stemming. The term "ಪಾತ್ರವನ್ನು" is shortened to "ಪಾತ್ರ" in the first sentence of the example above.



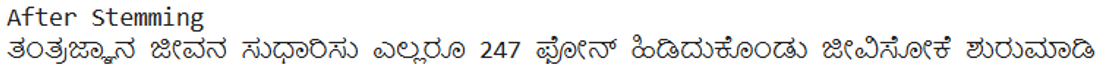
(Technology has improved our lives, so everyone has started living with a phone 24/7)

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



(technology has improved our lives so everyone has started living with a phone 24 7)

**Fig . 4 Screenshot after cleaning of the sentence.**



(technolog ha improv our live so everyon ha start live with a phone 24 7)

**Fig. 5 Screenshot after stemming of the sentence.**

An example of custom input is shown 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 Figures 4 and 5 following the cleaning and stemming procedures, which are crucial pipeline pretreatment stages. The classification outcomes following the use of different machine learning classifiers are displayed in Fig.6. Along with the BERT model for sequence classification, we experimented with a number of classifiers in our model, such as Linear SVC, Logistic Regression, SGD Classifier, SVC, Multinomial NB, Random Forest Classifier, 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 Kannada sentences, identified the sarcastic statements with an accuracy of 64.95%.

**3.1 COMPARITIVE ANALYSIS OF ALGORITHMS**

In order to classify Kannada sentences from a large dataset, we have implemented the BERT algorithm in conjunction with traditional classifiers such as Logistic Regression, SGD Classifier, SVC, Multinomial NB, Gaussian NB, and Random Forest Classifier. The Kannada sentences will undergo preprocessing before categorization, as was indicated in earlier parts. Sentiment analysis relies heavily on classification, which divides texts into two groups: sarcastic and non-sarcastic.

**3.1.1 Performance Measures**

The suggested Multinominal Naïve Bayes approach outperformed the other machine learning algorithms with an accuracy of 67% on a big dataset, based on the effectiveness indicators 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. It is expressed as the total of true and false positive (FP) values and the percentage of true positive (TP) outcomes.

Precision is equal to TP/(TP + FP).

**Recall**

percentage of cases with positive results. The positive instances are displayed by the algorithm's recall parameter. It is expressed as the total of true and false negative (FN) values and the proportion of true positive outcomes.

TP/(TP+FN) = Recall

**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 predictions average ratio. High data will receive a 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.

**Support**

Support is the parameter that determines how many of each type there are in the dataset. Stratified sampling or rebalancing may be necessary if there is uneven support in the training data, this suggests a structural flaw in the classifiers' output values. Rather, it interprets the evaluation process; the support for the different models remains constant.

**Accuracy**

True positive values are added to true negative values to calculate accuracy, which is then divided by the total number of samples taken from a document. When the model is balanced, this computation is appropriate; when there is a class imbalance, it is not.

.

**Macro Average**

The macro average is the mean of class precisions without accounting for proportion.   
Accuracy0=X   
Accuracy1=Y   
(X+Y)/2 is the macro average precision.

**Average Weighted**

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

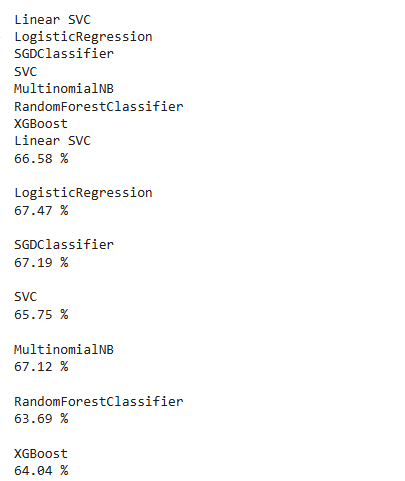
Weighted Average Precision = (0.68×0.514) + (0.66×0.486) = 0.6703

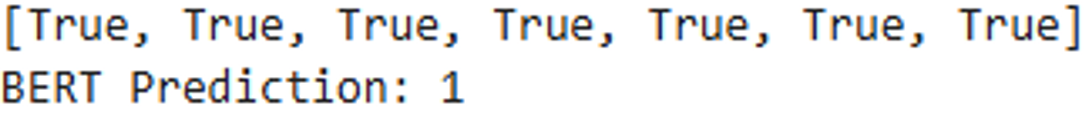
**3.1.2 Performance analysis of algorithms**

**Table 1. The level of accuracy of classifiers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL.NO |  | **Classifier** |  | **Accuracy** |
| 1 |  | Linear SVC |  | 66.71% |
| 2 |  | Logistic Regression |  | 67.19% |
| 3 |  | SGD Classifier |  | 66.78% |
| 4 |  | SVC |  | 65.68% |
| 5 |  | Multinominal NB |  | 67.19% |
| 6 |  | RandomForest Classifier |  | 64.04% |
| 7 |  | XGBoost |  | 63.21% |
| 8 |  | BERT |  | 64.93% |

Table 1 shows the calculated accuracy of several strategies across the dataset, and Fig. 6 analyzes the accuracy of all the algorithms used in our model. Seven thousand Kannada sentences were used to train our model. Training and testing sets were separated out of the dataset, with testing data making about 20% of the total. With an accuracy of 67.19%, Multinomial NB outperformed all other classifiers, according to the comparison. Accuracy was 67.19% for Logistic Regression, 66.78% for SGD Classifier, and 66.71% for Linear SVC. The Random Forest Classifier reached 64.04% accuracy, whereas SVC reached 65.68%. Bert's accuracy rate was 64.95%. With an accuracy of 63.21%, the XGBoost Classifier outperformed all other classifiers.





**Fig. 6 Boolean result of every ML model and result of Bert algorithm**

**Fig 8. Evaluation of machine learning algorithms' performance using a dataset of Kannada texts. The following methods are taken into consideration: Random Forest Classifier, Multinominal NB, Gaussian NB, K Neighbors Classifier, SGD Classifier, SVC, and Logistic Regression.**

Fig 8 and Table 2 compares various algorithms for the Kannada sarcasm detection model. The following instances anticipate the results, and parameters like precision, recall, and f1-score are computed based on true and false positives as well as true and false negatives.

It is regarded as 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) occurs when the input Kannada sentence is deemed sarcastic and the anticipated outcome is likewise sarcastic.
* A False Positive (FP) occurs 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 measures aid in evaluating how well the sarcasm detection system performs in correctly distinguishing between sarcastic and non-sarcastic Kannada sentences.

**Table 2 Classification Report of Classifiers**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Classifiers** | **Classification Report** | | | | | | | |
| 1 | **Linear Support Vector Classifier** | **Linear Support Vector Classifier** | | | | | | | |
|  | Precision | | recall | F1-score | | Support | |
| 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** | **Logistic Regression Classifier** | | | | | | | |
|  | Precision | | recall | F1-score | | Support | |
| 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** | **Stochastic Gradient Descent Classifier** | | | | | | | |
|  | | Precision | recall | F1-score | Support | | |
| 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** | **Support Vector Classifier** | | | | | | | |
|  | Precision | | recall | F1-score | | Support | |
| 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** | **Multinomial Naive Bayes Classifier** | | | | | | | |
|  | Precision | | recall | F1-score | | Support | |
| 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** | **Random Forest Classifier** | | | | | | | |
|  | Precision | | recall | F1-score | | Support | |
| 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** | **XGBoost** | | | | | | | |
|  | Precision | | recall | F1-score | | | Support |
| 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** | **BERT** | | | | | | | |
|  | 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 | |

* + 1. **Models used**

1. **Linear SVC**

One 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.

1. **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. Because of its ease of use and effectiveness, it is frequently utilized for binary classification jobs. One of the best algorithms for this assignment was Logistic Regression, which in our tests had an accuracy of 67.19%.

1. **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.

1. **SVC**

One machine learning approach for classification tasks is called Support Vector Classification (SVC). Finding the hyperplane that best divides the data points of several classes is how SVC operates. The hyperplane is selected to maximize the margin, or the separation between the hyperplane and the nearest data points for each class. Non-linearly separable data can be handled by SVC by employing a kernel function to map the data into a higher-dimensional space.

1. **Multinomial Naïve Bayes**

The Multinomial Naive Bayes algorithm is one of the NLP algorithms that is based on the 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 67.19%.

1. **Random Forest Classifier**

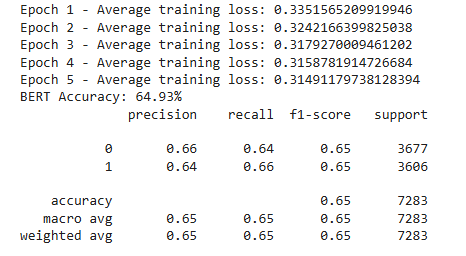
The Random Forest Classifier was used in our model to determine if Kannada sentences were sarcastic or not. During training, several decision trees are built using this ensemble learning technique, which then outputs the class that represents the average of the individual trees' predictions. The algorithm 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.

1. **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. In order to avoid overfitting, it optimizes performance via regularization approaches and makes use of decision tree-based learners. With an accuracy of 65.45%, XGBoost showed competitive performance in our tests, indicating its potential to handle the complex sarcastic patterns seen in Kannada language datasets.

1. **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 it uses a transformer architecture to comprehend the contextual links between words in a phrase. 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.



**Fig. 7 BERT model showing average training loss over five epochs and accuracy**

When training the BERT model, an epoch is defined as a single pass through the entire training dataset, during which time the model processes all of the training data to learn patterns and update its weights using back propagation. In this work, the BERT model was trained for five epochs:

**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 decreased to 0.32421, indicating improved learning and weight adjustment; and

**Epoch 3:** The training loss further decreased to 0.31792, indicating improved learning and weight adjustment; and

**Epoch 4:** The training loss further decreased 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 in Kannada sentences, attaining an accuracy of 64.95%. Its capacity to capture subtle contextual cues greatly improves classification performance when compared to traditional solution.

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.

**3.1.4 Comparison with Existing Work**

**Table 3 Comparing the suggested work with the current body of work**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Dataset used and Size | Language considered for processing | Approach | Algorithm used | Accuracy | Remarks |
| [1] | KanHope: A manually annotated Kannada-Englishcode-mixed dataset with 6,176 user-generated comments collected from YouTube. | Kannada-English (Code-mixed). | Hope speech detection using machine learning and deep learning models. Introduced DC-BERT4HOPE, a dual-channel BERT-based model leveraging translations. | Machine learning models (e.g., Logistic Regression, KNN, Decision Tree, Naive Bayes) and fine-tuned transformer-based models (e.g.,BERT,RoBERTa,DC-BERT4HOPE). | DC-BERT4HOPE achieved the highest weighted F1-score of 0.756, outperforming other models. | They have achieved higher accuracy because they have used Kannada-English combination whereas we have used purely Kannada sentences , hence accuracy achieved by our work is lesser but efficient. |
| [2] | A dataset of Kannada reviews collected from websites like Prajavani, Wedunia, and OneIndia News. Data dictionary includes over 500 positive and 500 negative words. | Kannada | |  | | --- | |  |   Sentiment analysis using a data dictionary for word-level sentiment polarity, tokenization, and tagging. Classification using a decision tree algorithm | Decision Tree Algorithm | Achieved 85% accuracy with the decision tree approach. | Detecting sarcasm in Kannada is challenging due to ambiguous machine translations. Achieving 67.12% accuracy on a 7000-sentence sarcasm dataset is more efficient than 85% accuracy on a smaller 1000-sentence sentiment analysis dataset. |
| [3] | Exact size not specified, but described as suitably large and demographically diverse. | Kannada | Hybrid approach combining traditional machine learning and deep learning. | Support Vector Machine (SVM), Random Forest, Bidirectional LSTM (BiLSTM), and Convolutional Neural Network (CNN); trained using Keras and pretrained word embeddings. | Achieved 83.57% accuracy after 5 training epochs (CNN); slightly decreased to 76.10% after 40 epochs with more stable loss. Shows strong potential for sarcasm detection in Kannada. | Our work's emphasis on sarcasm in Kannada aligns closely with the problem scope, avoiding the distractions of generic linguistic tasks. Incorporating modern NLP techniques and focusing on data-driven approaches might yield better performance and more scalable results, even if the initial accuracy is lower (67.12%). |
| [4] | 1200 tagged Kannada sentences (13,600 words for training, 1,053 words for testing), sourced from TDIL (Technology Development for Indian Languages). | Kannada. | CRF++ 0.50 (Conditional Random Fields) – Supervised Machine Learning. Deep Learning with Word2Vec + TensorFlow (Linear Chain CRF). | CRF++0.50 (Conditional Random Fields) – Supervised Machine Learning. Deep Learning with Word2Vec + TensorFlow (Linear Chain CRF). | CRF++: 76.45%; Deep Learning + Word2Vec: 71%. | Used a dataset of only **1,200 samples** (1,100 for training and 100 for testing), which is significantly smaller than your **7,000-line dataset**. The highest accuracy achieved in this work was **76.45%** using CRF++, and **71%** with deep learning. While our accuracy (67.12%) is lower, it is for a more complex task (sarcasm detection) and across a larger, more varied dataset, indicating a higher challenge and potential for improvement |
| [5] | Custom-built dataset with Kannada voice samples collected from diverse speakers, mostly reviewing mobile phones. Data was processed into Mel-spectrograms using NFFT. | Kannada | Multimodal sarcasm detection using both audio and textual data; includes speech preprocessing, feature extraction (MFCCs and linguistic cues), and deep learning. | Hybrid model using SVM, Random Forest, RNN with LSTM units, and CNNs; voice features processed with MFCCs and text with n-grams and sentiment features. | Achieved ~78% accuracy, ~75% recall, and 76.5% F1-score with the hybrid model; initial data-augmented CNN achieved 57.2%. | This work reports an initial accuracy of **57.2%**, which is significantly lower than our achieved accuracy of **67.19%** for text-based sarcasm detection. This indicates that their approach is still in early development and less reliable |
| Proposed  Model | 7000 sentences | Kannada | Evaluation of deep learning and different machine learning techniques for Kannada texts. | XGBoost, BERT, Multinominal Naïve Bayes, Gaussian Naïve Bayes, Random Forest Classifier, Support Vector Classifier, Logistic Regression, and Stochastic Gradient Descent Classifier | On large datasets, logistic regression and the multinomial Naive Bayes classifier have outperformed, with an accuracy of 67%. | Our Kannada sarcasm detection model uses a **large dataset of 7,000 texts**, surpassing previous works in size and diversity. It achieves **67.19% accuracy** using Logistic Regression and Multinomial Naive Bayes, focusing on **text-based sarcasm** across varied contexts. This makes it highly relevant for **social media monitoring and sentiment analysis**, advancing Kannada NLP research. |

Table 3 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, [3] and [5] concentrate on sarcasm detection in Kannada, with [3] employing hybrid techniques that combine deep learning and machine learning, and [5] investigating spoken Kannada using textual and prosodic data. Other studies use Indian languages, such as Hindi in [10], which uses a range of classifiers to perform sentiment analysis, and Telugu in [6], which uses hyperbolic characteristics to detect sarcasm with great accuracy. Mixed-language datasets are also common; for example, [9] combines Kannada and English for sentiment analysis, while [1] combines Kannada and English for hope speech detection. Furthermore, [4] discusses Kannada part-of-speech tagging, and [11] investigates Marathi sentiment analysis.

While models like [1], [3] and [4] obtained respectable accuracies when compared to current models specifically for Kannada, their performance was constrained by simpler approaches and smaller datasets. On the other hand, our suggested model was tested using eight distinct algorithms, including Linear SVC, BERT, XGBoost, Random Forest Classifier, Multinomial Naive Bayes, Support Vector Classifier, Logistic Regression, and Stochastic Gradient Descent Classifier, on a larger dataset of 7000 Kannada sentences. Multinomial Naive Bayes was the most successful of these algorithms, obtaining a 67% accuracy rate, which makes it a dependable option for sarcasm detection in textual Kannada data. However, albeit significantly lower, the accuracy of 64.95% attained by the BERT-based classifier shows the promise of using pre-trained models for language-specific tasks. Sarcasm identification is greatly enhanced by BERT's capacity to recognize subtle linguistic patterns and semantic correlations, even in languages with limited resources like Kannada. These findings demonstrate how successful our method is and stress the value of big datasets and cutting-edge machine learning methods in raising the precision and generalizability of Kannada sarcasm detection models.

**4. Conclusion**

Our study uses Natural Language Processing (NLP) techniques to offer a machine learning-based classification solution for sarcasm detection 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 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 a number of 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 in order 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. Our 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 purification, 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, such as Stochastic Gradient Descent Classifier (SGD), Multinomial Naive Bayes, Gaussian Naive Bayes, Random Forest Classifier, BERT, K-Neighbors Classifier, Linear SVC, Support Vector Classifier (SVC), and Logistic Regression, were used to train our models. Among them, Multinomial Naive Bayes and Logistic Regression both had an accuracy of 67.19%, demonstrating the promise of conventional methods when used with a carefully chosen dataset.

Our study outperforms the cited studies in a number of ways. Our text-based method covers a wider range of sarcasm than Bharti et al.'s Telugu and hyperbolic features, which makes it more suitable for real-world tasks like social media monitoring. With a substantially bigger dataset, our model directly addresses the subtleties of sarcasm recognition, in contrast to B.R. Shambhavi and Rajani Shree's work, which concentrated on POS tagging in Kannada using smaller datasets. Furthermore, our emphasis on textual data offers broader relevance in fields like sentiment analysis and opinion mining than Manohar R and Suma Swamy's work, which focused on audio data. Our study fills gaps in Kannada NLP research and lays the groundwork for future developments in regional language processing by utilizing a variety of datasets and numerous classifiers.

Overall, sarcasm detection for Kannada language is a promising area of research that has the potential to enhance our understanding of the language and its social and cultural contexts. With the continued development of advanced machine learning algorithms and the availability of more annotated data, we can expect significant progress in this field in the years to come.

# REFERENCES

1. Hande, A., Priyadharshini, R., Sampath, A., Thamburaj, K. P., Chandran, P., & Chakravarthi, B. R. (2021). Hope Speech detection in under-resourced Kannada language. *ArXiv*, abs/2108.04616.
2. Ranjitha, P., & Bhanu, K. N. (2021). Improved sentiment analysis for Dravidian language-Kannada using decision tree algorithm with efficient data dictionary. *IOP Conference Series: Materials Science and Engineering*, 1123(1), 012039. <https://doi.org/10.1088/1757-899x/1123/1/012039>
3. Manohar, R., & Swamy, S. (2023). Textual data analysis for identifying sarcasm in Kannada. *Journal of Intelligent & Fuzzy Systems*, 44(6), 1001–4055.
4. Shree, R. M., & Shambhavi, B. R. (2020). POS tagger model for Kannada text with CRF++ and deep learning approaches. *Journal of Discrete Mathematical Sciences and Cryptography*, 23, 485–493.
5. R., M., & Swamy, S. (2024). Voice-based sarcasm detection in Kannada language. *International Journal of Intelligent Systems and Applications in Engineering*, 12(14s), 356–367.
6. Bharti, S. K., Naidu, R., & Babu, K. S. (2021). Hyperbolic feature-based sarcasm detection in Telugu conversation sentences. *Journal of Intelligent Systems*, 30(1), 73–89. <https://doi.org/10.1515/jisys-2018-0475>
7. Akula, R., & Garibay, I. (2021). Interpretable multi-head self-attention architecture for sarcasm detection in social media. *Entropy*, 23, 394. <https://doi.org/10.3390/e23040394>
8. Akula, R., & Garibay, I. (2021). Explainable detection of sarcasm in social media. In Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 34–39). Association for Computational Linguistics.
9. Moores, B., & Mago, V. K. (2022). A survey on automated sarcasm detection on Twitter. *ArXiv*, abs/2202.02516.
10. Madan, A., & Ghose, U. (2021). Sentiment analysis for Twitter data in the Hindi language. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 784–789). IEEE. https://doi.org/10.1109/Confluence51648.2021.9377142
11. Kulkarni, A., Mandhane, M., Likhitkar, M., Kshirsagar, G. V., & Joshi, R. (2021). L3CubeMahaSent: A Marathi Tweet-based Sentiment Analysis Dataset. *Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.
12. Sharma, A., & Ghose, U. (2020). Sentimental analysis of Twitter data with respect to general elections in India. *Procedia Computer Science*, 173, 325–334.
13. Scola, E., & Segura-Bedmar, I. (2021). Sarcasm detection with BERT. *Procesamiento Del Lenguaje Natural*, 67, 13–25. https://doi.org/10.26342/2021-67-1