# Automated Sarcasm Identification in Code-Mixed Social Media Text using Machine Learning Techniques

Kogilavani Shanmugavadivel[1], Palani Murugan V[2], Pooja Sree M[3] and Pavul Chinnappan D[4,*]

*Department of AI , Kongu Enginnering College, Perundurai, Erode*

**Abstract**

Sarcasm detection in natural language processing is particularly challenging when dealing with code-mixed languages, often seen on social media platforms. Code mixing, where users mix multiple languages within a single statement, complicates the task for traditional NLP models. This study applies various machine learning techniques to detect sarcasm in code-mixed text. The dataset, comprising labeled sarcastic and non-sarcastic samples, is preprocessed using text normalization and TF-IDF vectorization. In this work, sarcasm in code-mixed text is detected using four models: Naive Bayes, Random Forest, Support Vector Machine, and Logistic Regression. The dataset is assessed using F1-score, recall, accuracy, and precision. The results highlight the advantages and disadvantages of each model for sarcasm recognition in multilingual settings. In contrast to the majority of earlier studies, which focus on monolingual data, this investigation delves into the little-studied field of code-mixed sarcasm detection. In addition, it draws attention to the difficulties in managing casual social media text and makes recommendations for enhanced deep learning methods in the future.

**Keywords**

Code-Mixed Languages, Natural Language Processing (NLP), Text Normalization, Support Vector Classifier, Term Frequency-Inverse Document Frequency (TF-IDF)

## 1. Introduction

On social media, sarcasm is frequently employed to convey irony, humor, or criticism. However, because the meaning of the words is frequently different from their literal meaning, it can be challenging to identify sarcasm in texts [1]. Identifying sarcasm is crucial for activities like social media language analysis, content moderation, and opinion comprehension.

Dealing with code-mixed languages makes the task considerably more difficult. When two or more languages are used in the same speech or discourse, this is known as code-mixing [2]. Users frequently combine Tamil and English in informal messages on social media sites like Facebook and Twitter. These mixes create additional challenges, such as grammatical and culturally specific word meanings, which make sarcasm more difficult to spot [3].

The purpose of this project is to create and evaluate machine learning models for sarcasm detection in text that is code-mixed between Tamil and English. The objective is to assess the effectiveness of these models and address the difficulties associated with managing multilingual data. [4] [5]. The results will aid in the development of natural language processing systems that handle datasets with many languages.

## 2. Literature Survey

The significance of sarcasm in security and commercial applications has made it an essential component of sentiment analysis and opinion mining. The study conducted by Swami et al. [6] examined a dataset of tweets that had been classified as ironic and sarcastic. They achieved an average F1-score of 0.78%, highlighting the importance of sentiment analysis. In [7], Khandagale et al. presented a method for identifying sarcasm in tweets that combine Hindi and English codes. Their method demonstrated its

*Forum for Information Retrieval Evaluation, December 12-15, 2024, India*

*Corresponding author.

kogilavani.sv@gmail.com (K. Shanmugavadivel); palanimuruganv.22aid@kongu.edu (P. M. V); poojasreem.22aid@kongu.edu (P. S. M); pavulchinnappand.22aid@kongu.edu (P. C. D)

potential for use in market research, social media monitoring, and customer service by achieving a noteworthy F1-score of 0.96% using Random Forest and Logistic Regression classifiers. In a study by Ratnavel et al. [8], a transformer-based model designed for text with mixed Tamil codes was presented. With the use of feed-forward neural networks, normalization, dropout layers, and multi-head self-attention, the model achieved a weighted F1-score of 0.77%. Compared to earlier state-of-the-art models, this improvement performed better.

Shetty et al. [9] investigated the use of different embeddings and models for sarcasm detection in social media messages written in Tamil and Malayalam. The model's effectiveness was demonstrated by its top-performing method, the TF-IDF Vectorizer, which obtained F1-scores of 0.79% for Tamil and 0.78% for Malayalam. Several machine learning models, such as SVM, Logistic Regression, K-Nearest Neighbors, Decision Tree, and a new CNN-based model, were tested by Chakravarthi et al. [10]. Their CNN model demonstrated its capacity to handle multilingual sarcasm detection by achieving the highest macro F1-scores: 0.75% for English, 0.62% for Tamil, and 0.67% for Malayalam. Kumar et al. [11] created a transformer-based model especially for Tamil code-mixed text. This method successfully addressed the difficulties presented by Tamil code-mixed text, achieving a weighted F1-score of 0.77% while simultaneously integrating feed-forward neural networks with multi-head self-attention.

In their study, Shanmugavadivel et al. [12] compared a number of machine learning and deep learning methods, such as Random Forest, Multinomial Naive Bayes, Logistic Regression, and Linear SVC, as well as deep learning models like CNN, LSTM, BiLSTM, BiGRU, and IndicBERT-based transfer learning. With a 0.66% accuracy rate on preprocessed Tamil code-mixed data, their hybrid CNN+BiLSTM model displayed the best performance. Another study used machine learning techniques such as K-Nearest Neighbors, Naive Bayes, SVC, and Random Forest to classify 1,500 citation sentences (Shanmugavadivel et al. [13]). They demonstrated the efficacy of these models by evaluating them based on criteria like accuracy and F1-score. A comparative study of deep learning algorithms, including transformer-based methods, hybrid models, and uni- and bi-directional models, was carried out by Thara et al. [14]. Their models produced outstanding F1-scores of 0.76 on the FIRE 2020 dataset and 0.99 on the EACL 2021 dataset by using selective translation, transliteration, and hyperparameter optimization, underscoring the significance of data pretreatment and fine-tuning.code mixed

Although real-time adaptability is limited, Orosoo et al. (2024) present a Federated Bi-LSTM Model for code-mixed text analysis that outperforms conventional techniques and achieves 0.99% accuracy [15]. By employing fuzzy logic with Word2Vec, GloVe, and BERT embeddings, Sharma et al. (2023) improve sarcasm detection on social media to an accuracy of up to 0.90% [16]. With F1 scores of 0.68% and 0.63%, Chakravarthi et al. (2023) placed seventh and fifth in the FIRE-2023 competition for their work on sarcasm identification in Tamil and Malayalam literature [17].

## 3. Dataset Description

We trained machine learning models to identify sarcasm in code-mixed social media chats using a dataset of 29,571 entries. A example of text marked as sarcastic or non-sarcastic is included with every entry. These samples include elements like emoticons, colloquial expressions, and informal spellings that make it difficult to identify sarcasm.

We used a different test dataset with 6,338 entries for evaluation. There are three different forms of sarcasm in this dataset: implicit, explicit, and indirect. The models had to manage intricate language and cultural subtleties in order to detect these types. The dataset additionally include slang, hashtags, and acronyms to guarantee that it represents authentic code-mixed social media conversations [18].

| TEXT | CATEGORY |
|---|---|
| Avara Control Pannunga Pls... Vera level expression... Thala... | Sarcastic |
| Thenavattu movie ku appuram ippodhan jiiva anna mass acting panni irrukkangga movie blockbuster dhaan | Non-sarcastic |

**Table 1**
Sample Texts from the Dataset

# 4. Methodology

## 4.1. Dataset Preprocessing

Preparing and cleaning code-mixed data is essential, particularly for jobs involving natural language processing such as sarcasm identification in social media text [19]. Code-mixed content usually combines several languages, including Tamil and English, and frequently include informal aspects like slang, symbols, and typographical errors. This emphasizes how crucial it is to prepare and clean data thoroughly in order to guarantee efficient analysis and model performance.

### 4.1.1. Text Cleaning

Lowercasing: All text was converted to lowercase in order to create uniformity. This improves overall performance by preventing confusion brought on by varied letter cases and making it simpler for the model to match text.

Noise reduction: Extraneous elements such as special characters, hashtags, mentions, URLs, and punctuation were eliminated. This phase is crucial because it keeps the analysis concentrated on the text's main elements, making it clearer and simpler for the model to comprehend.

Tokenization: Using a unique technique, we divided the text into smaller chunks, such as words or phrases. This helps the model capture the subtle variations in language use and is particularly helpful for assessing code-mixed material, which is text that contains various languages or dialects.

Handling Emojis and Emoticons: We turned emojis into text rather than eliminating them. Emojis frequently convey emotional meaning, which helps identify sarcasm. Maintaining these symbols aids the model's comprehension of the text because, depending on the context, a smiley face may indicate sarcasm.

Extraction of Features: To identify key terms, we employed a technique known as TF-IDF [20]. In order for the model to concentrate on the most relevant terms that may indicate sarcasm or other significant meanings, this strategy assists in identifying words that are common in one text but rare in another.

### 4.1.2. Label Encoding

For supervised learning, encode the target label (sarcastic or non-sarcastic). Sarcasm is usually binary, with 1 denoting sarcasm and 0 denoting non-sarcasm.

## 4.2. Models

### 4.2.1. Logistic Regression

For binary classification tasks, logistic regression is a straightforward yet powerful classification model. TF-IDF characteristics are frequently combined with logistic regression in sarcasm detection to forecast sarcasm. Because TF-IDF effectively captures the significance of words in a text, it is useful for differentiating between sarcastic and non-sarcastic information, which is why we utilized it in this work. Table 2 displays the logistic regression model's performance metrics for sarcasm prediction.

| Metrics | Value |
|---|---|
| Accuracy | 79.42 |
| Precision | 70 |
| Recall | 79 |
| F1-score | 68 |

**Table 2**
Classification Report for Logistic Regression

#### 4.2.2. Support Vector Machine

As seen in Table 3, Support Vector Machines (SVMs) with linear kernels are well recognized for their dependable performance in text classification and their efficacy in managing high-dimensional feature spaces, including text data.

| Metrics | Value |
|---|---|
| Accuracy | 79.11 |
| Precision | 62 |
| Recall | 68.6 |
| F1-score | 65.2 |

**Table 3**
Classification Report for SVM

#### 4.2.3. Random Forest

Table 4 demonstrates how the Random Forest model can capture non-linear correlations between features, which enables it to handle the complexity of sarcasm. The model performed well and could detect subtle patterns of sarcasm. It is more dependable and efficient since it uses numerous decision trees to lessen overfitting.

| **Metrics** | **Value** |
|---|---|
| Accuracy | 77.73 |
| Precision | 1.0 |
| Recall | 1.0 |
| F1-score | 1.0 |

**Table 4**
Classification Report for Random Forest

#### 4.2.4. Naive Bayes

TF-IDF or n-gram features are frequently utilized with Naive Bayes, especially when dealing with sparse text data. Its probabilistic methodology aids in locating crucial textual expressions that suggest sarcasm. It successfully captures significant words that are essential for sarcasm recognition when paired with TF-IDF. When more than one language, such as Tamil and English, are utilized together, this concept works well. In text classification problems, Naive Bayes performs well even if it assumes feature independence. It uses word probabilities to identify sarcastic patterns. Because of this, it's a good and efficient option for social media content analysis. For sarcasm detection, Naive Bayes works consistently, as Table 5 demonstrates.

| Metrics | Value |
|---|---|
| Accuracy | 77.68 |
| Precision | 68.9 |
| Recall | 70.5 |
| F1-score | 68 |

**Table 5**
Classification Report for Naive Bayes

## 5. Workflow

This Figure 1 displays the procedures required to develop and assess a machine learning model for text data. Data preprocessing and loading are the first steps in the process. Stop words, unnecessary letters, and punctuation are among the noises that are eliminated from the raw data at this first stage. For uniformity, text normalization methods like stemming and lowercasing are also used. Inconsistencies are fixed and the data is prepared for analysis.

Converting the text to a numerical representation comes next after preprocessing. Textual input is transformed into numerical vectors that machine learning models may handle using methods like TF-IDF or word embeddings (e.g., Word2Vec, GloVe). In order to enhance model performance, feature extraction is also carried out to determine which words or phrases are most relevant to the prediction task.

The data is separated into training and test sets after transformation. The machine learning model is trained using the training set, and the test set is kept apart for assessment. To train the model, standard techniques like Support Vector Machines, Random Forests, Naive Bayes, and Logistic Regression can be used. The model learns to find patterns and connections between the input features and the target labels during training.

Predictions are made on the test set following training, and the model's performance is assessed using a number of measures, including accuracy, precision, recall, and F1-score. The model's ability to identify sarcasm or other desired results is indicated by these metrics. Should the performance be unsatisfactory, the model's hyperparameters are adjusted by methods such as grid search or random search to improve the model's precision and effectiveness.

The model can be used to generate predictions on fresh, unknown text data once it has been optimized. It could be necessary to regularly assess and retrain the model to make sure it keeps working well when fresh data is added.

## 6. Result and Discussion

Four machine learning models were used in this study to evaluate the ability to identify sarcasm in Tamil-English code-mixed text: Naive Bayes, Random Forest, Support Vector Classifier (SVC), and Logistic Regression. The most accurate of these was Logistic Regression, which had an accuracy of 79.42%. This demonstrates that when it comes to sarcasm detection in mixed-language social media content, Logistic Regression performs better than other models like Random Forest and SVC in terms of precision, recall, and F1-score.

According to the findings, it can be difficult to identify sarcasm in mixed-language texts due to language mixing and informal writing. Despite their effectiveness, the models failed to pick up on a few tiny sarcastic cues. To increase accuracy in future studies, deep learning may be used. Table 6 displays the models' performance and shows that Logistic Regression was the most accurate and therefore more effective than the others.

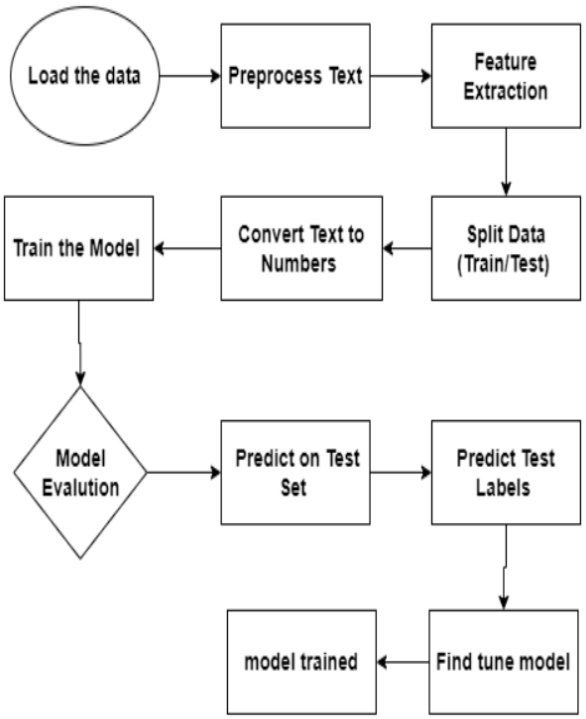


**Figure 1:** Processed workflow

| Models | Accuracy |
|---|---|
| Logistic Regression | 79.42 |
| Random Forest | 77.73 |
| Naive Bayes | 77.68 |
| SVM | 79.11 |

**Table 6**
Report For Models

## 7. Conclusion

This study achieved an outstanding 79.42% accuracy rate using the Logistic Regression model. The classification report focuses on important performance indicators including precision, recall, and F1-score to demonstrate how well the model separates sardonic from non-sarcastic text. Future research on code-mixed language will benefit from the well-structured and annotated dataset this study provides, which is a significant contribution to sentiment analysis. But there are still issues like overfitting and restrictions on using the model in certain situations. This work represents a significant breakthrough in sarcasm recognition despite these obstacles, providing a strong basis for future research and the creation of more sophisticated methods.

## Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: drafting content, grammar and spelling check, etc. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## References


[1] A. Perera, A. Caldera, Sentiment analysis of code-mixed text: A comprehensive review., Journal of Universal Computer Science (JUCS) 30 (2024).

[2] M. Bedi, S. Kumar, M. S. Akhtar, T. Chakraborty, Multi-modal sarcasm detection and humor classification in code-mixed conversations, IEEE Transactions on Affective Computing 14 (2021) 1363–1375.

[3] N. Sripriya, T. Durairaj, K. Nandhini, B. Bharathi, K. K. Ponnusamy, C. Rajkumar, P. K. Kumaresan, R. Ponnusamy, C. Subalalitha, B. R. Chakravarthi, Findings of shared task on sarcasm identification in code-mixed dravidian languages, FIRE 2023 16 (2023) 22.

[4] B. R. Chakravarthi, A. Hande, R. Ponnusamy, P. K. Kumaresan, R. Priyadharshini, How can we detect homophobia and transphobia? experiments in a multilingual code-mixed setting for social media governance, International Journal of Information Management Data Insights 2 (2022) 100119.

[5] B. R. Chakravarthi, N. Sripriya, B. Bharathi, K. Nandhini, S. C. Navaneethakrishnan, T. Durairaj, R. Ponnusamy, P. K. Kumaresan, K. K. Ponnusamy, C. Rajkumar, Overview of the shared task on sarcasm identification of dravidian languages (malayalam and tamil) in dravidiancodemix, in: Forum of Information Retrieval and Evaluation FIRE-2023, 2023.

[6] S. Swami, A. Khandelwal, V. Singh, S. S. Akhtar, M. Shrivastava, A corpus of english-hindi code-mixed tweets for sarcasm detection, arXiv preprint arXiv:1805.11869 (2018).

[7] K. Khandagale, H. Gandhi, Sarcasm detection in hindi-english code-mixed tweets using machine learning algorithms, in: International Conference on Computing in Engineering & Technology, Springer, 2022, pp. 221–229.

[8] R. Ratnavel, R. G. Joshua, S. Varsini, M. A. Kumar, Sarcasm detection in tamil code-mixed data using transformers, in: International Conference on Speech and Language Technologies for Low-resource Languages, Springer, 2023, pp. 430–442.

[9] P. Shetty, Sarcasm identification in dravidian languages tamil and malayalam., in: FIRE (Working Notes), 2023, pp. 240–248.

[10] B. R. Chakravarthi, Hope speech detection in youtube comments, Social Network Analysis and Mining 12 (2022) 75.

[11] M. A. Kumar, Sarcasm detection in tamil code-mixed data using transformers, Speech and Language Technologies for Low-Resource Languages (????) 430.

[12] K. Shanmugavadivel, S. H. Sampath, P. Nandhakumar, P. Mahalingam, M. Subramanian, P. K. Kumaresan, R. Priyadharshini, An analysis of machine learning models for sentiment analysis of tamil code-mixed data, Computer Speech & Language 76 (2022) 101407.

[13] K. Shanmugavadivel, M. Subramanian, V. Palanimurugan, et al., Innovationengineers@ dravidianlangtech-eacl 2024: Sentimental analysis of youtube comments in tamil by using machine learning, in: Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, 2024, pp. 262–265.

[14] S. Thara, P. Poornachandran, Social media text analytics of malayalam–english code-mixed using deep learning, Journal of big Data 9 (2022) 45.

[15] M. Orosoo, J. C. Sekhar, M. Rengarajan, N. Tsendsuren, A. Gopi, Y. A. B. El-Ebiary, A. I. Taloba, et al., Analysing code-mixed text in programming instruction through machine learning for feature extraction, International Journal of Advanced Computer Science & Applications 15 (2024).

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

[17] B. R. Chakravarthi, N. Sripriya, B. Bharathi, K. Nandhini, S. Chinnaudayar Navaneethakrishnan, T. Durairaj, R. Ponnusamy, P. K. Kumaresan, K. K. Ponnusamy, C. Rajkumar, Overview of the shared task on sarcasm identification of Dravidian languages (Malayalam and Tamil) in DravidianCodeMix, in: Forum of Information Retrieval and Evaluation FIRE - 2023, 2023.

[18] B. R. Chakravarthi, S. N, B. B, N. K, T. Durairaj, R. Ponnusamy, P. K. Kumaresan, K. K. Ponnusamy,

C. Rajkumar, Overview of sarcasm identification of dravidian languages in dravidiancodemix@fire-2024, in: Forum of Information Retrieval and Evaluation FIRE - 2024, DAIICT , Gandhinagar, 2024.

[19] R. Kanakam, R. K. Nayak, Sarcasm detection on social networks using machine learning algorithms: A systematic review, in: 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), IEEE, 2021, pp. 1130–1137.

[20] R. Singh, R. Srivastava, A novel balancing technique with tf-idf matrix for short text classification to detect sarcasm, Int. J. Mech. Eng 7 (2022) 602–607.