

# COMPARATIVE ANALYSIS OF MACHINE LEARNING AND DEEP LEARNING TECHNIQUES FOR SARCASM DETECTION IN INDIAN POLITICAL HEADLINES

S. Sathish Kumar<sup>1</sup>, Arunkalyan Muddamsetty<sup>2\*</sup>, Yash Tyagi<sup>3</sup>, Archana Gaddam<sup>4</sup>

<sup>1</sup>Assistant Professor, <sup>2,3,4,5</sup>UG Student, <sup>1,2,3,4,5</sup>Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning)

<sup>1,2,3,4,5</sup>J B Institute of Engineering and Technology (JBIET), Bhaskar Nagar, Moinabad, Hyderabad, 500075, Telangana, India

\*Corresponding author: Arunkalyan Muddamsetty ([arunmuddamsetty@gmail.com](mailto:arunmuddamsetty@gmail.com))

**Abstract**—Sarcasm is a complex linguistic phenomenon in which the intended meaning diverges from the literal interpretation, often conveying implicit criticism or irony. In political discourse, particularly in news headlines, sarcasm plays a crucial role in shaping public perception, where misinterpretation can lead to misinformation and unreliable sentiment analysis. Detecting sarcasm remains a challenging task due to its context-dependent nature, reliance on cultural and domain-specific knowledge, and the presence of implicit semantic contradictions, especially in short text formats. Conventional machine learning approaches based on surface-level features such as TF-IDF lack the ability to capture contextual semantics, while deep learning models, although more expressive, often struggle with implicit relationships in limited-context data. Furthermore, most existing studies rely on general-purpose datasets, limiting their effectiveness in domain-specific settings such as Indian political headlines. To address this gap, this paper presents a comparative analysis of machine learning and deep learning techniques for sarcasm detection using a manually curated dataset of Indian political headlines, enabling evaluation in a culturally grounded context. Experimental results show that machine learning models achieve moderate performance (~65–68% accuracy), whereas deep learning approaches outperform them, with the best model exceeding 70% accuracy after hyperparameter optimization, highlighting the importance of contextual modeling for sarcasm detection.

**Keywords**—Sentiment Analysis, Sarcasm Detection, NLP, Machine Learning, Deep Learning, Transformers, Neural Networks

## 1) INTRODUCTION

Sarcasm detection in natural language processing (NLP) is a challenging task due to its implicit meaning, context dependence, and frequent reliance on world knowledge and cultural cues [1]. The Merriam-Webster dictionary defines sarcasm as “the use of words that mean the opposite of what you really want to say especially in order to insult someone, to show irritation, or to be funny” [2]. Unlike literal text, sarcastic expressions often convey sentiment opposite to their surface meaning, making them difficult for traditional rule-based and statistical models to interpret correctly. This challenge becomes

particularly significant in applications such as sentiment analysis, public opinion mining, and media analysis, where misinterpretation can lead to misleading insights. In political discourse, especially in news headlines, sarcasm is frequently used to express criticism or satire in a subtle manner, further increasing the complexity of accurate interpretation. These characteristics highlight the need for robust approaches capable of modeling contextual and implicit semantic relationships in text.

In India, mainstream news dissemination platforms continue to be widely trusted, particularly in the context of political reporting. The increasing use of sarcasm in political headlines introduces significant challenges for reliable sentiment analysis and opinion mining, as sarcastic expressions can distort the intended meaning of information. While sarcasm can be easily recognized in face-to-face communication through tone, facial expressions, and gestures, its detection in written text is considerably more difficult due to the absence of these cues [4]. This challenge is further amplified in short-form content such as headlines and social media text, where limited context makes it harder to interpret implicit meaning and underlying intent.

Several studies have applied machine learning techniques for sentiment and emotion analysis to identify sarcastic text; however, these approaches often struggle to capture grammatical ambiguity and contextual dependencies inherent in sarcasm [3]. To address these limitations, prior work has explored ensemble learning strategies that combine multiple models and handcrafted linguistic features related to lexical usage, readability, and semantic patterns, demonstrating improved predictive accuracy [4]. More recently, deep learning models such as CNNs and LSTMs have been employed to learn semantic and sequential representations directly from text, offering improvements over traditional approaches. Despite these advancements, deep learning models still face challenges in capturing implicit semantic relationships in short texts. Additionally, many existing methods rely heavily on feature engineering or are evaluated on general-purpose datasets, limiting their ability to generalize to domain-specific and culturally nuanced contexts such as Indian political headlines. These limiting their ability to generalize to domain-specific contexts.

Despite significant progress in sarcasm detection, limited research has focused on domain-specific and culturally grounded datasets, particularly in the context of Indian political

discourse. Most existing studies rely on general-purpose datasets such as social media posts or reviews, which fail to capture the linguistic nuances, cultural references, and contextual dependencies present in Indian political headlines [3], [4]. The absence of a dedicated dataset for this domain restricts the ability of models to generalize effectively and accurately interpret sarcasm in real-world political content. This gap highlights the need for a domain-specific approach that accounts for the unique characteristics of sarcasm in Indian news media.

To address these challenges, this paper presents a comparative study of machine learning and deep learning techniques for sarcasm detection using a manually curated dataset of Indian political headlines. The proposed framework evaluates traditional machine learning models based on TF-IDF representations alongside deep learning architectures capable of learning semantic and contextual patterns directly from text. By conducting a systematic comparison across these approaches, the study aims to analyze their effectiveness in capturing sarcastic expressions within a domain-specific setting. Experimental evaluation is performed using standard classification metrics, providing insights into the strengths and limitations of each modeling paradigm. The findings of this work contribute to improving sarcasm detection in politically nuanced text and highlight the importance of contextual modeling in domain-specific natural language processing tasks.

## 2) RELATED WORK

Traditional sarcasm detection approaches primarily rely on machine learning models combined with surface-level text representations such as TF-IDF and Bag-of-Words. These methods transform textual data into numerical feature spaces and apply classifiers such as Support Vector Machines (SVM), Random Forest, Naïve Bayes, and Decision Trees for prediction. Prior work using social media datasets, particularly Twitter, demonstrates that such models can achieve strong performance, with SVM-based approaches reaching up to 90% accuracy in controlled settings [5]. Similarly, studies on Indian-language datasets using TF-IDF features report competitive performance, with classifiers such as Naïve Bayes achieving accuracy above 85% in political tweet classification tasks [6]. These models are typically evaluated using standard metrics such as precision, recall, F1-score, and accuracy, highlighting their effectiveness in high-dimensional text classification. However, despite these results, traditional approaches remain limited by their reliance on handcrafted features and their inability to capture contextual meaning, grammatical ambiguity, and implicit semantic relationships inherent in sarcasm, leading to poor generalization across domains.

Deep learning approaches have gained increasing attention in sarcasm detection due to their ability to learn representations directly from unstructured text and automatically extract semantic features. Commonly used architectures include Convolutional Neural Networks (CNN), artificial neural networks, and Long Short-Term Memory (LSTM) networks, often combined to improve performance [3]. CNN-based models are effective in capturing local n-gram features and salient patterns from word embeddings, while hybrid architectures combining CNN and LSTM have been proposed to model both local and sequential dependencies in text. For instance, attention-enhanced multi-level LSTM frameworks further improve sarcasm detection by capturing sentiment semantics

and contextual relationships across words. LSTM networks, in particular, are widely used due to their ability to model long-range dependencies and sequential information through gated mechanisms, enabling them to capture linguistic patterns such as negation and sentiment shifts [7]. However, despite these advantages, deep learning models still face limitations in sarcasm detection, especially in short texts such as headlines, where implicit meaning and contextual cues are limited [8]. Sequential architectures such as LSTM may lose global contextual information and struggle to capture subtle semantic contradictions, motivating the development of attention-based and transformer models for improved contextual understanding [9], [10].

Recent advancements in sarcasm detection have been driven by transformer-based architectures, which address the limitations of earlier sequence models. Traditional sequence-to-sequence models struggle with long-range dependencies and lack parallelization capabilities; this led to the introduction of the transformer architecture, which leverages attention mechanisms to model global contextual relationships within text [3]. Transformer models such as BERT have demonstrated strong performance in sarcasm detection tasks by learning contextual representations from large-scale corpora, eliminating the need for extensive feature engineering. Empirical studies show that transformer-based models outperform both traditional machine learning and earlier deep learning approaches, with BERT achieving notable improvements in precision, recall, and F1-score over LSTM-based models [11]. These models are particularly effective in capturing subtle semantic contrasts and contextual cues required for sarcasm detection. However, despite their advantages, transformer models still face challenges in handling implicit world knowledge and contextual ambiguity, especially in short texts where external context is limited. Additionally, their performance is highly dependent on the quality and domain relevance of the training data, highlighting the importance of dataset design in domain-specific applications.

## 3) DATA COLLECTION

### A. Data Source

To construct a domain-specific dataset for sarcasm detection, political news headlines were collected from prominent Indian news platforms, including The Hindu and The Indian Express. The resulting dataset, referred to as the Indian Political Sarcasm Dataset [12], forms the foundation for all experiments conducted in this study. These sources were selected due to their wide coverage of national political events and consistent publication of concise, context-rich headlines, making them suitable for sarcasm analysis in short-text settings.

Data collection was performed through automated web scraping using Python-based tools. Headlines were extracted from multiple paginated sections of each website to ensure diversity and coverage across time. For The Hindu, both standard paginated pages and dynamically loaded content (via asynchronous fragment endpoints) were accessed to retrieve extended archives beyond the initial pages. For The Indian Express, headlines were extracted from structured metadata embedded within the webpage (JSON-LD format), enabling reliable parsing of article titles across multiple pages.

To ensure robustness and avoid request blocking, HTTP requests were issued with appropriate headers, including user-agent and referrer information, and controlled using time delays between successive requests. Extracted headlines were cleaned by removing HTML encodings, filtering out very short or incomplete entries, and eliminating duplicates to maintain dataset quality. The final dataset consists of unique political headlines stored in a structured format for further annotation and experimentation.

### B. Data Annotation

The collected headlines were annotated using a semi-automated approach guided by a structured prompt-based framework. Specifically, an AI-assisted annotation strategy was employed, where each headline was evaluated using predefined linguistic and contextual rules designed to capture sarcasm in political discourse. The annotation prompt incorporated multiple rule categories, including sentiment-polarity mismatch, exaggeration, logical incongruity, structural contradiction, linguistic markers, and context-specific political cues. These rules enabled systematic identification of sarcasm through both explicit and subtle indicators such as irony, mock praise, skepticism, and implicit criticism.

The annotation process followed a multi-step reasoning pipeline, where the literal meaning and implied meaning of each headline were first analyzed, followed by rule-based evaluation to determine the presence of sarcasm. A headline was labeled as sarcastic if one or more rules indicating irony or contradiction were triggered, even in subtle cases. Additionally, a confidence score was assigned to reflect the strength of sarcasm, allowing differentiation between strong and weak signals.

To improve consistency and reduce noise, post-processing steps were applied, including removal of duplicate entries, filtering of incomplete or ambiguous headlines, and balancing of class distributions. While the semi-automated approach enables scalable annotation, a small degree of label noise may be present due to the inherently subjective nature of sarcasm. Nevertheless, the structured rule-based prompting ensures a consistent and linguistically grounded labeling process suitable for comparative model evaluation. The annotation process was applied to the dataset introduced in [12].

### C. Dataset Statistics

The final dataset consists of 11,784 political headlines, evenly distributed across two classes to ensure balanced model training and evaluation. As shown in Table 1, the dataset contains 5,892 sarcastic and 5,892 non-sarcastic instances, eliminating class imbalance bias and enabling fair comparison across machine learning and deep learning models.

**Table 1: Dataset Distribution**

Label	Count
Non-Sarcastic	5892
Sarcastic	5892
<b>Total</b>	<b>11784</b>

The dataset captures a diverse range of political narratives, including election coverage, party dynamics, governance issues, and public discourse. Sarcastic headlines typically exhibit features such as contradiction, rhetorical framing, and implicit criticism, while non-sarcastic headlines follow a more neutral and factual reporting style.

These examples illustrate the contrast between sarcastic and non-sarcastic expressions. Sarcastic headlines often rely on implicit meaning, irony, or contextual critique, whereas non-sarcastic headlines convey direct and literal information. The inclusion of both explicit and subtle sarcasm ensures that the dataset reflects real-world linguistic variability, making it suitable for evaluating the effectiveness of different modeling approaches.

**Table 1: Dataset Examples**

Label	Example
Sarcastic	“When needed I was deshbhakt, now I am deshdrohi”
Sarcastic	“Ajit Pawar was blackmailed for 10 years... shows why he wanted to go to BJP”
Non-Sarcastic	“As Uttarakhand political winds indicate a shift, tight contest on the cards...”
Non-Sarcastic	“Sena helped BJP to rise in Maharashtra, can cause its downfall too”

## 4) METHODOLOGY

### A. Problem Definition

Sarcasm detection in political headlines is formulated as a binary text classification problem, where the goal is to determine whether a headline expresses sarcasm. Let a headline be represented as a sequence of tokens:

$$H = w_1, w_2, \dots, w_n \quad [1]$$

The task is to learn a function:

$$f(H) \rightarrow y \quad [2]$$

where  $y = 1$  denotes a *sarcastic* headline and  $y = 0$  denotes a *non-sarcastic* headline.

headline.

Unlike standard text classification tasks, sarcasm detection involves identifying implicit semantic relationships such as irony, contradiction, and contextual incongruity, which are not directly observable from surface-level text. This challenge is further amplified in political headlines, which are typically short, context-dependent, and often rely on cultural or domain-specific knowledge for correct interpretation.

To address this, the problem is approached using different textual representations depending on the modeling paradigm. For machine learning models, headlines are transformed into

sparse vector representations using techniques such as TF-IDF, capturing term frequency and importance. For deep learning models, the input is represented using dense embeddings or tokenized sequences, enabling the model to learn contextual and semantic relationships within the text.

The objective of this study is not only to perform accurate classification but also to compare the effectiveness of machine learning and deep learning approaches in capturing sarcasm within a domain-specific dataset of Indian political headlines.

## B. Preprocessing

Text preprocessing plays a critical role in transforming raw political headlines into structured representations suitable for machine learning and deep learning models. Due to differences in modeling paradigms, separate preprocessing pipelines were employed for traditional machine learning approaches and deep learning models.

### 1) Preprocessing for Machine Learning Models

For machine learning models, a comprehensive preprocessing pipeline was applied to reduce noise and standardize textual input. Initially, all headlines were converted to lowercase to ensure uniformity. Punctuation symbols were removed to eliminate non-informative characters, followed by stopword removal using the NLTK stopword corpus to discard common words that do not contribute to semantic meaning.

Additional text normalization steps were performed to handle irregularities in real-world data. Unicode quotation marks were standardized, ellipses were normalized, and non-alphanumeric characters were removed while preserving key punctuation such as apostrophes and question marks when relevant. Excess whitespace was also eliminated to maintain clean token boundaries.

Tokenization was performed using the TweetTokenizer, which is well-suited for short and informal text such as headlines. The tokenized output was then converted back into a string representation for compatibility with vectorization techniques.

### 2) Preprocessing for Deep Learning and Transformer Models

In contrast, deep learning and transformer-based models require minimal preprocessing, as they are capable of learning contextual and semantic representations directly from raw text. Therefore, only lightweight cleaning was performed, including the removal of URLs, HTML tags, and excessive whitespace.

Unlike traditional approaches, no stopword removal or aggressive normalization was applied, as such operations may discard important contextual cues necessary for sarcasm detection. The cleaned text was then directly used for tokenization within the respective deep learning and transformer architectures.

This dual preprocessing strategy ensures that each modeling approach receives input in a form best suited to its representational capabilities, enabling a fair and effective comparison between machine learning and deep learning techniques.

## C. Feature Representation

Effective feature representation is crucial for capturing the linguistic and contextual characteristics of sarcasm in text. In this study, different representation techniques were employed based on the modeling paradigm, including TF-IDF for machine learning models, embedding-based representations for neural networks, and tokenizer-based encodings for transformer models.

### 1) TF-IDF Representation for Machine Learning

For traditional machine learning models, textual data was transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) representation. This approach captures the importance of words based on their frequency within a document and across the corpus.

The TF-IDF vectorizer was configured with a maximum vocabulary size of 5000 features to control dimensionality and reduce noise. Both unigrams and bigrams were included (n-gram range of (1,2)) to capture local word patterns relevant to sarcasm. Additionally, terms appearing in fewer than two documents were removed to eliminate rare noise, while overly frequent terms (appearing in more than 90% of documents) were filtered out. Sublinear term frequency scaling was applied to improve representation quality.

This representation produces sparse high-dimensional vectors that are well-suited for classical classifiers such as Support Vector Machines and Random Forests, but lacks the ability to capture deeper contextual semantics.

### 2) Embedding-Based Representation for Neural Networks

For deep learning models, text was represented using dense vector embeddings initialized with pretrained GloVe embeddings of 100 dimensions [13]. Tokenization was performed using a vocabulary size limited to the top 5000 most frequent words, with out-of-vocabulary tokens handled explicitly.

Each headline was converted into a sequence of integer indices and padded to a fixed length of 80 tokens to ensure uniform input size. An embedding matrix was constructed by mapping words to their corresponding GloVe vectors, while unseen words were initialized with random values.

The embedding coverage achieved approximately 90%, indicating that the majority of the vocabulary was successfully mapped to pretrained semantic representations. This embedding-based approach enables neural networks to capture semantic similarity and contextual patterns beyond surface-level features.

### 3) Tokenizer-Based Representation for Transformer Models

For transformer-based models, contextual representations were generated using pretrained tokenizers associated with the selected model architecture (e.g., BERT). Instead of manual feature engineering, raw text was directly tokenized into subword units using a pretrained tokenizer.

Each input headline was converted into input IDs and attention masks, with padding and truncation applied to a fixed sequence length of 128 tokens. This representation allows the model to dynamically learn contextual relationships between tokens using self-attention mechanisms.

Unlike TF-IDF and static embeddings, transformer-based representations capture context-dependent meanings, enabling more effective modeling of sarcasm, which often relies on subtle semantic interactions and implicit contradictions within the text.

Overall, the use of multiple feature representation strategies enables a comprehensive comparison between traditional and modern approaches to sarcasm detection.

#### D. Models

To evaluate sarcasm detection performance, a range of models spanning traditional machine learning, deep learning, and transformer-based approaches were implemented. This selection enables a comparative analysis across different levels of representational complexity.

##### 1) Machine Learning Models

Traditional machine learning models were used as baseline approaches, operating on TF-IDF feature representations.

A Linear Support Vector Machine (Linear SVM) was employed due to its effectiveness in high-dimensional sparse text data and its strong performance in classification tasks [14]. Additionally, a Random Forest classifier was used to capture non-linear relationships through an ensemble of decision trees, improving robustness and generalization [15].

These models primarily rely on surface-level lexical features and do not explicitly capture contextual or sequential dependencies within the text.

##### 2) Deep Learning Models

Deep learning models were implemented to capture semantic and sequential patterns in the data. Two CNN-LSTM hybrid architectures were used [16], [7].

The first model consists of a single convolutional layer followed by an LSTM layer, where the convolutional component extracts local n-gram features and the LSTM captures sequential dependencies. The second model extends this architecture by incorporating three convolutional layers before the LSTM, enabling hierarchical feature extraction and improved representation of complex patterns.

These models leverage embedding-based representations to learn richer semantic features compared to traditional machine learning approaches.

##### 3) Transformer Models

Transformer-based models were used to leverage contextual embeddings generated through self-attention mechanisms. Pretrained models including BERT (`\textit{bert-base-uncased}`), RoBERTa, and DistilBERT were fine-tuned for the sarcasm detection task [10], [17], [18].

Unlike traditional and standard deep learning models, transformers capture long-range dependencies and context-sensitive meanings by dynamically attending to different parts of the input sequence. This makes them particularly effective for detecting sarcasm, which often relies on subtle contextual cues and implicit semantic relationships.

Overall, the combination of these models enables a comprehensive comparison between traditional, deep learning, and transformer-based approaches for sarcasm detection.

#### 5) EXPERIMENTAL SETUP

To evaluate the performance of the proposed sarcasm detection models, a consistent experimental framework was adopted across all machine learning, deep learning, and transformer-based approaches.

The dataset was divided into training and testing subsets using an 80–20 split, ensuring that model performance was evaluated on unseen data. Stratified sampling was applied to maintain equal class distribution in both splits, preserving the balance between sarcastic and non-sarcastic instances.

All experiments were conducted using a combination of cloud-based computational resources. Machine learning and deep learning models were trained on a Google Cloud instance with 8 vCPUs and 32 GB RAM, while transformer-based models were fine-tuned using Google Colab with NVIDIA T4 GPU acceleration to handle the computational demands of self-attention mechanisms.

Hyperparameter tuning was performed to optimize model performance. For machine learning models, parameters such as regularization strength and tree depth were adjusted, while for deep learning and transformer models, training configurations including learning rate, batch size, and number of epochs were tuned. This ensured that each model was evaluated under near-optimal conditions for fair comparison.

Model performance was assessed using standard classification metrics, including Accuracy, Precision, Recall, and F1-score.

Accuracy measures the overall correctness of predictions and is defined as:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad [3]$$

where TP denotes true positives, TN true negatives, FP false positives, and FN false negatives.

Precision measures the proportion of correctly predicted positive instances among all predicted positives:

Recall measures the proportion of correctly predicted positive instances among all actual positives:

$$Precision = TP / (TP + FP) \quad [4]$$

Recall measures the proportion of correctly predicted positive instances among all actual positives:

$$Recall = TP / (TP + FN) \quad [5]$$

The F1-score, which provides a balance between Precision and Recall, is defined as:

$$F1 = 2 * ((Precision * Recall) / (Precision + Recall)) \quad [6]$$

These metrics provide a comprehensive evaluation of model performance, particularly in capturing both correctness and class-wise prediction quality in sarcasm detection.

#### 6) RESULTS AND ANALYSIS

The performance of the evaluated models was assessed using Accuracy, Precision, Recall, and F1-score on the test dataset. Table 3 summarizes the results obtained from machine learning, deep learning, and transformer-based models after hyperparameter optimization.

## A. Quantitative Results

TABLE I. Quantitative Results

Model	Accuracy	Precision	Recall	F1-score
SVM	0.68	0.68	0.68	0.68
Random Forest	0.65	0.65	0.65	0.64
CNN-LSTM (Shallow)	0.68	0.68	0.68	0.68
CNN-LSTM (Deep)	0.69	0.69	0.69	0.68
BERT	0.69	<b>0.79</b>	0.52	0.63
RoBERTa	0.65	0.60	<b>0.95</b>	0.73
DistilBERT	<b>0.70</b>	0.70	0.71	<b>0.71</b>

## B. Comparison

The results reveal clear differences between machine learning, deep learning, and transformer-based approaches for sarcasm detection.

### 1) Machine Learning vs Deep Learning

Traditional machine learning models, particularly Linear SVM with TF-IDF features, achieved moderate performance with an accuracy of approximately 68%. These models rely on sparse, frequency-based representations and are effective at capturing surface-level lexical patterns such as word occurrence and local co-occurrence. However, they lack the ability to model sequential dependencies and contextual relationships between words.

Deep learning models, specifically CNN-LSTM architectures, demonstrate improved performance by leveraging dense embeddings and sequential modeling. The convolutional layers capture local n-gram patterns, while the LSTM component models temporal dependencies within the text. The deeper CNN-LSTM variant achieves higher accuracy compared to the shallow architecture, indicating that hierarchical feature extraction contributes to better representation of complex linguistic patterns.

Despite these improvements, the performance gain of deep learning models over traditional machine learning approaches remains relatively modest. This suggests that while embedding-based representations capture richer semantics than TF-IDF, they still struggle to fully model the implicit and context-dependent nature of sarcasm, particularly in short text such as headlines.

### 2) Transformer Models vs Deep Learning

Transformer-based models significantly outperform both machine learning and conventional deep learning approaches. Among the evaluated models, DistilBERT achieves the best overall performance, with the highest accuracy and F1-score, demonstrating strong capability in capturing contextual relationships.

Unlike CNN-LSTM architectures, transformer models utilize self-attention mechanisms to dynamically model

interactions between all words in a sequence, regardless of their position. This allows them to capture long-range dependencies and subtle semantic contradictions that are characteristic of sarcasm.

The performance variation among transformer models also provides important insights. BERT exhibits high precision but relatively lower recall, indicating conservative predictions where fewer sarcastic instances are identified. In contrast, RoBERTa achieves very high recall but lower precision, suggesting a tendency to over-predict sarcasm. DistilBERT achieves a balanced trade-off between precision and recall, making it more reliable for this task.

### 3) Precision-Recall Trade-offs

An important observation across the evaluated models is the variation in precision-recall behavior, particularly among transformer-based architectures. BERT achieves high precision but relatively lower recall, indicating that it makes more conservative predictions and identifies fewer sarcastic instances while maintaining correctness. In contrast, RoBERTa demonstrates significantly higher recall but lower precision, suggesting that it tends to classify more instances as sarcastic, including a higher number of false positives.

This trade-off reflects different model sensitivities to sarcasm cues. Conservative models such as BERT prioritize precision, making them suitable for applications where false positives are costly. On the other hand, recall-oriented models such as RoBERTa are more effective in scenarios where capturing all potential sarcastic instances is critical, even at the expense of misclassification.

DistilBERT achieves a more balanced trade-off between precision and recall, resulting in the highest F1-score among the evaluated models. This balance indicates that it effectively captures sarcastic patterns while maintaining generalization, making it more suitable for real-world sarcasm detection tasks where both precision and recall are important.

### 4) Impact of Contextual Representation

The superior performance of transformer-based models highlights the importance of contextual representation in sarcasm detection. Unlike machine learning models based on TF-IDF and deep learning models relying on static embeddings, transformers utilize self-attention mechanisms to capture relationships between all tokens in a sequence. This enables them to identify subtle semantic contradictions, implicit cues, and contextual dependencies that are essential for detecting sarcasm.

Sarcasm in political headlines often depends on implicit meaning, cultural references, and contextual framing rather than explicit lexical signals. Transformer models are better equipped to capture these characteristics, as they dynamically adjust token representations based on surrounding context, allowing for more nuanced interpretation of text.

Additionally, transformer models demonstrate greater robustness to label noise present in the dataset. Given that the dataset was annotated using a semi-automated approach, some degree of labeling inconsistency is expected. Traditional machine learning models, which rely on surface-level features, are more sensitive to such noise and may overfit to incorrect patterns. In contrast, transformer-based models leverage

contextual embeddings and large-scale pretraining, enabling them to generalize better and mitigate the impact of noisy labels.

This robustness, combined with their ability to model complex semantic relationships, explains why transformer-based approaches consistently outperform both traditional machine learning and conventional deep learning models in sarcasm detection tasks.

## 7) DISCUSSION

The results of this study highlight the critical role of contextual understanding in sarcasm detection, particularly within the domain of political headlines. Unlike traditional text classification tasks, sarcasm detection requires identifying implicit meaning, contradiction, and subtle linguistic cues that are often not explicitly expressed. The comparative evaluation across machine learning, deep learning, and transformer-based models demonstrates that performance improves as models gain the ability to capture richer semantic and contextual relationships.

Machine learning models, while efficient and effective for baseline performance, are limited by their reliance on surface-level representations such as TF-IDF. These representations fail to encode word order, context, and semantic dependencies, which are essential for interpreting sarcasm. As a result, such models struggle to distinguish between literal and non-literal language, especially in short and ambiguous text such as political headlines.

Deep learning models provide improvements by incorporating dense embeddings and sequential modeling, enabling better representation of semantic relationships. However, their reliance on static embeddings and limited capacity to model long-range dependencies restricts their ability to fully capture the nuanced and context-dependent nature of sarcasm. This is particularly evident in cases where sarcasm arises from subtle contradictions or requires broader contextual awareness.

Transformer-based models address these limitations by leveraging self-attention mechanisms to dynamically model relationships between all tokens in a sequence. This allows for a more comprehensive understanding of contextual meaning, making them particularly effective for sarcasm detection. The observed performance improvements, especially in models such as DistilBERT, indicate that contextual embeddings are better suited for capturing implicit semantic patterns and handling variability in linguistic expression.

Another important observation is the robustness of transformer-based models to label noise. Given that the dataset was constructed using a semi-automated annotation process, some degree of labeling inconsistency is expected. Despite this, transformer models demonstrate stable and balanced performance, suggesting their ability to generalize beyond noisy or imperfect annotations. This highlights their suitability for real-world applications, where perfectly labeled datasets are often unavailable.

Overall, the findings of this study emphasize that sarcasm detection is inherently a context-driven problem. Models that incorporate contextual representation and dynamic interaction between words significantly outperform those relying on static or frequency-based features. These results reinforce the

importance of advancing toward context-aware architectures for improved performance in complex natural language understanding tasks such as sarcasm detection.

## 8) CONCLUSION

This paper presented a comparative study of machine learning, deep learning, and transformer-based approaches for sarcasm detection in Indian political headlines using a domain-specific dataset. The results demonstrate that traditional machine learning models achieve moderate performance, while deep learning models provide incremental improvements through semantic and sequential modeling. Transformer-based models, particularly DistilBERT, achieve the best overall performance, highlighting the effectiveness of contextual embeddings in capturing implicit and nuanced linguistic patterns.

The study confirms that sarcasm detection is inherently a context-dependent problem, and models capable of capturing contextual relationships significantly outperform those relying on surface-level representations. Additionally, the robustness of transformer models to noisy, semi-automated annotations further supports their suitability for real-world applications where perfect labeling is difficult to achieve.

Overall, this work emphasizes the importance of context-aware modeling for improving performance in sarcasm detection and similar natural language understanding tasks.

## 9) FUTURE WORK

Future work can focus on expanding the dataset with more diverse political headlines across sources and time periods to improve generalization, as well as incorporating multilingual data, particularly regional Indian languages, to better capture culturally specific sarcasm. Integrating external contextual information such as political events or background knowledge may further enhance the detection of implicit sarcasm. Additionally, exploring advanced transformer architectures and improved fine-tuning strategies could yield performance gains. Finally, reducing label noise through partial manual validation or active learning, along with incorporating explainability techniques, can help build more reliable and interpretable sarcasm detection systems for real-world applications.

## REFERENCES

- [1] O. Prakash, "Sarcasm Detection for Sentiment Analysis using Deep Learning Enhancement," *Procedia Computer Science*, vol. 258, pp. 203–212, Jan. 2025, doi: <https://doi.org/10.1016/j.procs.2025.04.233>.
- [2] Merriam-Webster, "Definition of SARCASM," Merriam-webster.com, 2019. <https://www.merriam-webster.com/dictionary/sarcasm>
- [3] A. Alqahtani, Lubna Alhenaki, and Abeer Alsheddi, "Text-based Sarcasm Detection on Social Networks: A Systematic Review," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, Jan. 2023, doi: <https://doi.org/10.14569/ijacsa.2023.0140336>.
- [4] J. Pradhan, R. Verma, S. Kumar, and V. Sharma, "An Efficient Sarcasm Detection using Linguistic Features and Ensemble Machine Learning," *Procedia Computer Science*,

- vol. 235, pp. 1058–1067, Jan. 2024, doi: <https://doi.org/10.1016/j.procs.2024.04.100>.
- [5] M. Bhakuni, K. Kumar, Sonia, C. Iwendi, and A. Singh, “Evolution and Evaluation: Sarcasm Analysis for Twitter Data Using Sentiment Analysis,” *Journal of Sensors*, vol. 2022, pp. 1–10, Oct. 2022, doi: <https://doi.org/10.1155/2022/6287559>.
- [6] R. S. Patil and S. R. Kolhe, “Supervised classifiers with TF-IDF features for sentiment analysis of Marathi tweets,” *Social Network Analysis and Mining*, vol. 12, no. 1, May 2022, doi: <https://doi.org/10.1007/s13278-022-00877-w>.
- [7] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [8] Y. Tay, Anh Tuan Luu, Siu Cheung Hui, and J. Su, “Reasoning with Sarcasm by Reading In-Between,” arXiv (Cornell University), Jan. 2018, doi: <https://doi.org/10.18653/v1/p18-1093>.
- [9] A. Vaswani et al., “Attention Is All You Need,” Cornell University, Jun. 12, 2017. <https://arxiv.org/abs/1706.03762>
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv:1810.04805 [cs], May 2019, Available: <https://arxiv.org/abs/1810.04805#>
- [11] E. Scola and I. Segura-Bedmar, “Sarcasm Detection with BERT Detección de Sarcasmo con BERT,” doi: <https://doi.org/10.26342/2021-67-1>.
- [12] A. Muddamsetty, S. S. Kumar, Y. Tyagi, and A. Gaddam, “arunkalyan12/indian-political-sarcasm-dataset,” GitHub, 2026. <https://github.com/arunkalyan12/indian-political-sarcasm-dataset> (accessed Apr. 15, 2026).
- [13] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.
- [14] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: <https://doi.org/10.1007/BF00994018>.
- [15] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 2018.
- [17] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” arXiv (Cornell University), vol. 1, Jul. 2019, doi: <https://doi.org/10.48550/arxiv.1907.11692>.
- [18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” arXiv.org, Feb. 29, 2020. <https://arxiv.org/abs/1910.01108v4>