

Self Attention K Folded With Xgboost

Amit Kumar Srivastava¹, Reena Srivastava²

¹School of Computer Application BBD University Lucknow, India amit_sri_in@yahoo.com

²School of Computer Application BBD University Lucknow, India dean.soca@bbdu.ac.in

Abstract—Sarcasm detection is a challenging task in Natural Language Processing (NLP) due to its reliance on contextual, contrastive, and figurative cues. While deep learning models such as BERT have advanced contextual representation learning, they often rely on computationally expensive end-to-end fine-tuning. This paper proposes a hybrid framework that integrates BERT embeddings with a custom self-attention mechanism for feature extraction, combined with XGBoost classification for efficient decision making. The self-attention layer highlights sarcasm-relevant tokens, producing contextualized, weighted sentence representations, which are then fed into XGBoost to establish robust decision boundaries without neural classification overhead. We further employ Stratified 6-fold cross validation to ensure balanced evaluation and reliable performance reporting. Experiments on benchmark sarcasm datasets demonstrate that the proposed approach achieves superior F1-scores compared to traditional baselines, validating the effectiveness of combining contextual embeddings with gradient-boosted classification. This work highlights the potential of attention-augmented embeddings with ensemble learning for advancing sarcasm detection in social media text.

Keywords- Sarcasm detection, mBERT, multilingual NLP, English, Hindi, Bengali, Punjabi

1. INTRODUCTION

Sarcasm is a challenging and nuanced form of figurative language that complicates the task of Natural Language Processing (NLP) [1] [2]. Unlike explicit sentiment expressions, sarcasm relies heavily on subtle contextual cues such as contrastive statements, irony, exaggeration, or the interplay between text and emojis. These cues are often implicit, making it difficult for computational models to distinguish sarcastic utterances from literal statements. Misinterpreting sarcasm can negatively impact applications such as sentiment analysis, opinion mining, and social media analytics, where accurate understanding of user intent is essential.

With the rise of deep learning, contextual embedding models such as Bidirectional Encoder Representations from Transformers (BERT) [3] [4] [5] have significantly advanced sarcasm detection by capturing long-range dependencies and nuanced linguistic relationships. However, end-to-end fine-tuning of large transformer models can be computationally expensive, data-hungry, and prone to overfitting when trained on noisy, limited sarcasm datasets. Conversely, traditional machine learning models such as logistic regression [6], SVMs [7], or tree-based classifiers offer computational efficiency but lack the ability to capture deeper contextual semantics, resulting in poor generalization to complex language patterns.

To address these limitations, we propose a hybrid sarcasm detection model that leverages the strengths of both contextual embeddings and ensemble learning. The model consists of three core components:

1. **Feature Extraction with BERT Embeddings** – Sentences are encoded into dense contextualized vectors that retain semantic and syntactic relationships relevant to sarcasm cues.

2. **Custom Self-Attention Layer** – A PyTorch-based self-attention mechanism re-weights token-level embeddings to emphasize sarcasm-inducing words, phrases, and symbols, producing a contextually weighted representation of the sentence.
3. **Classification with XGBoost and Stratified k-Fold Validation** – The attention-weighted embeddings are passed into Xtreme Gradient Boosting (XGBoost) [8] [9], an ensemble method that builds robust decision boundaries through iterative tree-based learning. To ensure fairness and reliability in performance evaluation, we employ Stratified k-Fold cross-validation (with k=6), which balances sarcasm and non-sarcasm samples across folds.

In summary, this work introduces a novel hybrid framework that integrates BERT embeddings, self-attention weighting, and XGBoost classification for sarcasm detection in social media text. By combining deep contextual feature extraction with ensemble-based learning, the model achieves both interpretability and scalability while avoiding the heavy computational overhead of end-to-end neural classifiers. The inclusion of Stratified k-Fold validation further strengthens the reliability of results, ensuring that improvements are not dataset-specific but reflect general performance. This hybrid approach demonstrates that uniting contextual embeddings with gradient-boosted decision trees provides a promising and efficient pathway for advancing sarcasm-aware NLP systems.

II. RELATED WORK IN SARCASM DETECTION

2.1 Sarcasm Detection with Traditional Machine Learning

Early research in sarcasm detection primarily employed traditional machine learning methods such as Support Vector Machines (SVMs) and Logistic Regression (LR). These approaches relied on handcrafted features including lexical patterns, punctuation, sentiment shifts, and n-gram representations. While computationally efficient, such models often failed to capture the deeper contextual dependencies required for sarcasm interpretation. Their reliance on surface-level linguistic cues limited their ability to generalize across domains, especially in informal and noisy environments such as social media.

2.2 Deep Learning Models for Sarcasm Detection

The emergence of deep learning significantly improved sarcasm detection by allowing models to learn hierarchical feature representations automatically. Convolutional Neural Networks (CNNs) [10] were used to capture local patterns in text, while Long Short-Term Memory (LSTM) [11] networks modeled sequential dependencies across tokens. More recently, transformer-based architectures such as BERT have achieved state-of-the-art performance by leveraging bidirectional contextual embeddings. Fine-tuning BERT for sarcasm detection has shown promising results, as the model can capture long-range dependencies and subtle semantic contrasts. However, such approaches are computationally intensive and require substantial amounts of labeled data, which is often scarce for sarcasm-specific corpora.

2.3 Attention Mechanisms in NLP

Attention mechanisms [12] have become a core component in NLP, enabling models to selectively focus on the most relevant parts of an input sequence when generating predictions. In the context of sarcasm detection, attention has been particularly useful for highlighting sarcasm-inducing cues such as contrastive sentiment words (“great” vs. “broken”), hyperbolic phrases, or emojis. Recent studies have explored self-attention

layers as an effective means of modeling context without relying on recurrent architectures [13] [14], demonstrating improved performance in tasks where nuanced dependencies must be captured. Nevertheless, most attention-based approaches have been integrated into fully neural classification pipelines, leaving little exploration of how attention-enhanced embeddings could be leveraged outside neural classifiers.

2.4 Gradient Boosting Approaches in Text Classification

Beyond neural architectures, ensemble methods such as Gradient Boosting Machines (GBM) and their optimized variant XGBoost have proven effective for a variety of text classification tasks. These models combine multiple weak learners (decision trees) to form a strong classifier capable of handling non-linear relationships and reducing overfitting through regularization. XGBoost in particular has been widely adopted for its scalability and robust performance in high-dimensional feature spaces. However, its application in sarcasm detection has been limited, as it traditionally depends on manually engineered features rather than deep contextual embeddings.

2.5 Research Gap

Although significant progress has been made in sarcasm detection through traditional ML [15], deep learning [16], and attention-based models, few works have investigated the potential of hybrid architectures that integrate the strengths of both paradigms. Specifically, the combination of BERT embeddings for contextual feature extraction, custom self-attention for token-level weighting, and XGBoost for efficient classification remains underexplored. This gap motivates our proposed framework, which seeks to unify deep contextual representation learning with ensemble-based decision making to achieve robust and scalable sarcasm detection.

III. METHODOLOGY

This section presents the architecture and operational flow of the proposed hybrid sarcasm detection model. The model integrates BERT embeddings for contextual feature extraction, a custom self-attention mechanism for token-level weighting, Stratified k-Fold cross-validation for reliable evaluation, and an XGBoost classifier for efficient and robust decision-making. The workflow ensures that both contextual understanding and computational efficiency are maintained without relying on heavy neural classifiers.

3.1 Feature Extraction Using BERT

The first stage in the pipeline involves generating high-quality contextual representations of input text using Bidirectional Encoder Representations from Transformers (BERT) [17] [18]. Each input sentence is tokenized into subword units and passed through a pre-trained BERT model. The model outputs contextual embeddings that capture syntactic relationships, semantic dependencies, and sentiment nuances between words in a sentence—elements crucial for identifying sarcasm.

3.2 Custom Self-Attention Mechanism

Although BERT provides strong contextual embeddings, not all tokens contribute equally to detecting sarcasm. To address this, a custom self-attention layer is designed and implemented using the PyTorch framework. This mechanism assigns a learnable attention weight to each token, allowing the model to focus on sarcasm-inducing features such as contrastive words (“love” vs. “hate”), intensifiers, or emojis.

To optimize computational efficiency, an `encode_text()` function converts raw text into fixed-size vector embeddings using the pre-trained BERT model and the custom self-attention layer. Finally, the `squeeze()` operation removes unnecessary dimensions, yielding a compact representation ready for classification.

This step ensures that features most indicative of sarcasm receive higher weights, enhancing the downstream classifier's interpretability and performance.

3.3 Stratified k-Fold Cross-Validation

To ensure reliable and unbiased model evaluation, we implement Stratified k-Fold cross-validation with $k = 6$ [19]. This technique splits the dataset into six equal subsets (folds) while maintaining the same ratio of sarcastic and non-sarcastic samples in each fold.

- `n_splits = 6` ensures balanced evaluation across folds.
- `shuffle = True` randomizes the dataset to prevent bias from sample order.
- `random_state = fixed` ensures reproducibility of the splits.

Each fold is used once as a validation set while the remaining five are used for training. The F1-score is computed for every fold and stored in an array. The final performance is reported as the average F1-score across all folds. This setup provides a robust measure of model generalization and minimizes overfitting risks.

3.4 XGBoost Classification

The classification layer of the model employs XGBoost (Extreme Gradient Boosting), a high-performance ensemble learning algorithm that constructs an ensemble of decision trees. Each tree sequentially corrects the errors made by previous trees, effectively reducing bias and variance in predictions.

By feeding the attention-weighted BERT embeddings into XGBoost, the model combines the representational richness of deep learning with the interpretability and efficiency of ensemble methods. This approach eliminates the need for dense neural classification layers, substantially reducing training complexity [20] [21].

Key hyperparameters used in XGBoost include:

- `n_estimators`: number of trees; more trees capture nuanced sarcasm patterns.
- `learning_rate`: smaller values (e.g., 0.05–0.1) promote gradual learning and better generalization.
- `max_depth`: controls tree depth; deeper trees capture complex patterns but may overfit.
- `min_child_weight`: ensures splits occur only when the data is informative.
- `reg_alpha` / `reg_lambda`: apply L1 and L2 regularization to mitigate overfitting from rich BERT features.
- `subsample` / `colsample_bytree`: introduce randomness to improve generalization and robustness.

The classifier outputs binary predictions — 0 indicating Not Sarcastic and 1 indicating Sarcastic.

IV. DATASET

To evaluate the performance and generalizability of our proposed hybrid sarcasm detection model, we employ the same multilingual sarcasm-labeled corpus introduced in our previous work on sarcasm-aware preprocessing (ALDEA). This ensures consistency in benchmarking and enables a direct comparison between preprocessing and model-level advancements. The dataset encompasses four major linguistic categories—English, Hindi, Bengali, and Hinglish (code-mixed Hindi-English)—capturing both high-resource and under-resourced languages.

The complete dataset consists of 126,983 entries, stratified by language as follows:

- **English:** 67,967 samples (53%)
- **Hindi:** 16,179 samples (12%)
- **Bengali:** 42,837 samples (33%)
- **Hinglish:** integrated subset within Hindi samples for code-mixed evaluation

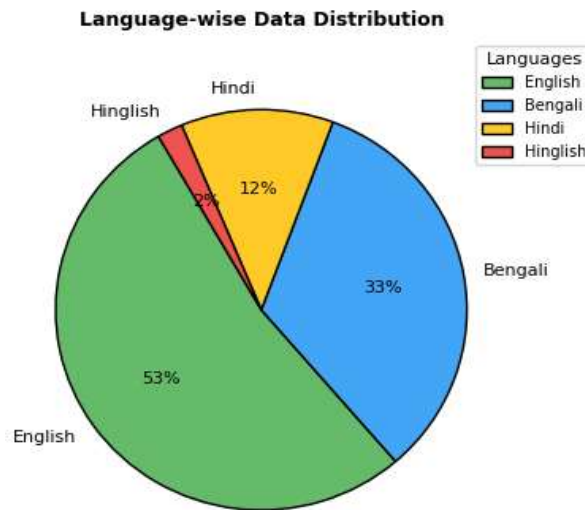


Figure 1. Language- wise dataset distribution

The data is split into 80% training and 20% testing subsets. This multilingual and balanced design ensures robust evaluation across varying linguistic structures, humor conventions, and syntactic forms.

4.1 English Dataset

The English dataset comprises sarcasm-annotated social media posts collected from platforms such as Twitter and Reddit. The dataset captures a wide range of sarcasm types—from subtle irony and understatement to overt mockery—expressed through both textual and symbolic cues such as emojis and punctuation. English data serves as a baseline for testing model performance on high-resource languages and standard sarcasm detection benchmarks.

4.2 Hindi Dataset

The Hindi dataset includes sarcasm-labeled social media content sourced primarily from Twitter and YouTube comments by native Hindi speakers. It exhibits unique challenges such as free word order, regional idiomatic expressions, and mixed script writing

(Devanagari and Roman) [22] [23]. These features make the dataset ideal for evaluating how effectively the hybrid model leverages BERT's multilingual embeddings and self-attention mechanisms to capture sarcasm in morphologically flexible and linguistically rich languages.

4.3 Bengali Datasets

The Bengali portion of the corpus combines three distinct resources: Bangla-SARC, Bangla-SARC3, and Ben-SARC [24] [25] [17]. These datasets encompass diverse lexical and stylistic variations of Bengali sarcasm, reflecting regional humor, cultural idioms, and social media-specific expressions. As Bengali is an under-resourced language in NLP, incorporating these datasets allows us to assess the generalization capability of the hybrid model and its adaptability to morphologically complex languages [26].

4.4 Hinglish Dataset

A dedicated Hinglish dataset—comprising code-mixed Hindi-English text—is used to evaluate the model's robustness in multilingual and informal communication contexts. Hinglish text frequently mixes Romanized Hindi vocabulary with English syntax and includes sarcasm-inducing features such as phonetic spellings, hashtags, and emojis (e.g., “Waah kya baat hai 😂”) [27] [28]. This dataset tests the model's ability to handle code-switching, script variation, and cross-lingual sarcasm markers within a single sentence.

The unified multilingual corpus exhibits the following characteristics:

- **Source:** Social media posts, comments, and tweets from multiple platforms.
- **Languages:** English, Hindi, Bengali, and Hindi-English (Hinglish code-mixed).
- **Size:** 126,983 samples, with approximately equal distribution between sarcastic and non-sarcastic content.
- **Labels:** Binary — 0: Non-Sarcastic, 1: Sarcastic.
- **Metadata:** Includes platform information, language tags, and presence of emojis or hashtags.

By maintaining linguistic diversity and balancing sarcastic vs. non-sarcastic samples, this corpus provides a comprehensive evaluation environment for multilingual sarcasm detection [29] [30]. The use of this dataset in both preprocessing (ALDEA) and model-level research ensures methodological continuity, allowing the present work to focus on performance gains introduced by the hybrid BERT–Self-Attention–XGBoost architecture.

V. MODEL IMPLEMENTATION ALGORITHM

Input:

Raw sarcasm-labeled multilingual dataset $D=\{x_1, x_2, \dots, x_n\}$ with labels $Y=\{y_1, y_2, \dots, y_n\}$
 Predicted sarcasm label, where 0 = Non-Sarcastic and 1 = Sarcastic

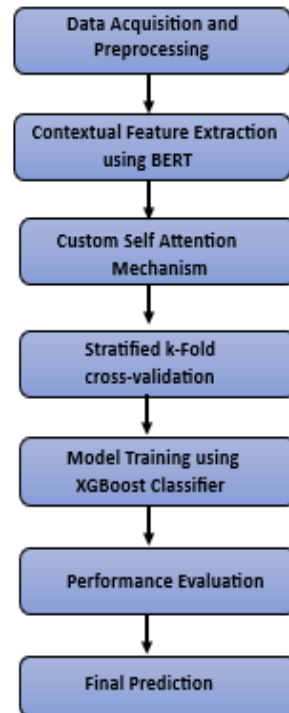


Figure 2. Model implementation pipeline

Step 1: Data Acquisition and Preprocessing

- 1.1 Import the multilingual sarcasm dataset (English, Hindi, Bengali, Hinglish).
- 1.2 Apply ALDEA sarcasm-aware preprocessing to preserve emojis, contrastive cues, and contextual tokens.
- 1.3 Perform text normalization, token cleanup, and label encoding.

Step 2: Contextual Feature Extraction using BERT

The Contextual Feature Extraction using BERT module employs the pre-trained bert-base-multilingual-uncased model to generate deep contextual embeddings from input text. The tokenizer and BERT model convert sentences into meaningful vector representations that capture semantic and syntactic relationships. These embeddings are enhanced using a Self-Attention mechanism for better focus on important tokens. Finally, Stratified K-Fold Cross-Validation ensures balanced and robust evaluation during training.

```

# def main():
tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-uncased')
bert = BertModel.from_pretrained('bert-base-multilingual-uncased').to(device)
attn = SelfAttention(768).to(device)
# attn = MultiheadAttention(embed_dim=768, num_heads=8).to(device)

x = encode_text(df['headline'].tolist(), tokenizer, bert, attn)
y = df['is_sarcastic'].values

skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
f1_scores = []

```

Step 3: Custom Self-Attention Encoding

The Self-Attention module computes context-aware representations of input embeddings by learning how each token relates to others within the same sequence. It first projects

the input embeddings through a small feed-forward network consisting of linear transformations and a non-linear activation function (Tanh) [31] to generate attention scores. These scores are masked where necessary (to ignore padding or future tokens) and normalized using a softmax function to produce attention weights. Finally, the module computes a weighted sum of the embeddings based on these attention weights, resulting in a contextually enhanced representation that captures dependencies across the input sequence.

```
class SelfAttention(nn.Module):
    def __init__(self, hidden_dim):
        super(SelfAttention, self).__init__()
        self.projection = nn.Sequential(
            nn.Linear(hidden_dim, hidden_dim),
            nn.Tanh(),
            nn.Linear(hidden_dim, 1)
        )

    def forward(self, embeddings, mask):
        scores = self.projection(embeddings).squeeze(-1)
        scores = scores.masked_fill(mask == 0, -1e9)
        weights = torch.softmax(scores, dim=-1).unsqueeze(-1)
        weighted = torch.sum(weights * embeddings, dim=-1)
        return weighted
```

Step 4: Stratified k-Fold Cross-Validation Setup

The Contextual Feature Extraction using BERT module leverages a pre-trained BERT multilingual model (bert-base-multilingual-uncased) [32] [33] to generate deep contextualized embeddings for input text. The tokenizer converts the input sentences into subword tokens compatible with BERT's vocabulary, while the model encodes these tokens to produce dense vector representations that capture semantic and syntactic nuances across multiple languages. These contextual embeddings serve as high-quality input features for downstream tasks such as sarcasm detection. To further refine these embeddings, a self-attention layer is applied, enhancing the model's ability to focus on the most informative parts of the text. The setup uses **Stratified K-Fold cross-validation** to ensure robust evaluation and balanced performance across data splits.

```
Fold 1
F1 score of Fold 1: 86.080586

Fold 2
F1 score of Fold 2: 85.934718

Fold 3
F1 score of Fold 3: 85.635113

Fold 4
F1 score of Fold 4: 85.017629

Fold 5
F1 score of Fold 5: 85.745538

Fold 6
F1 score of Fold 6: 85.312330
precision    recall  f1-score   support

      0       0.89       0.84       0.87      11424
      1       0.82       0.88       0.85       9739

 accuracy          0.86      21163
 macro avg          0.86       0.86      21163
weighted avg          0.86       0.86       0.86      21163

Average f1 score is: 85.62098565311523
1416.8261499404907
```

Step 5: Model Training using XGBoost Classifier

5.1 Initialize XGBoost with optimized hyperparameters:

- `n_estimators`: captures fine-grained sarcasm cues.
- `learning_rate`: promotes smoother convergence.
- `max_depth`: balances model complexity and generalization.
- `min_child_weight`: avoids noisy splits.
- `reg_alpha`, `reg_lambda`: regularization to prevent overfitting.
- `subsample`, `colsample_bytree`: introduce randomness for diversity.

5.2 Train XGBoost on the attention-weighted BERT embeddings(F_i).

5.3 After each fold, record Precision, Recall, and F1-score metrics.

```
model = XGBClassifier(
    n_estimators = 350,
    max_depth=8,
    learning_rate=0.05,
    subsample=0.8,
    colsample_bytree=0.8,
    use_label_encoder=False,
    eval_metric='logloss',
    gamma=0.4,
    reg_alpha=0.7,
    reg_lambda=1.0,
    min_child_weight=4,
    random_state=42,
    scale_pos_weight=1.5
)
```

Step 6: Performance Evaluation

6.1 For each fold f_i :

- Compute Precision, Recall, and F1-score.
- Store F1-score in results array.

6.2 Compute average F1-score across all folds:

$$F1_{avg} = \frac{1}{k} \sum_{i=1}^k F1_{f_i}$$

6.3 Generate classification report summarizing results.

Step 7: Final Prediction

7.1 For unseen text input x_{new} :

- Convert text into BERT embeddings.
- Pass embeddings through the self-attention layer to obtain F_{new} .
- Feed F_{new} into the trained XGBoost classifier.

7.2 Output:

$$y' = \begin{cases} 0, & \text{if non-sarcastic} \\ 1, & \text{if sarcastic} \end{cases}$$

VI. COMPARATIVE ANALYSIS

To evaluate the effectiveness of the proposed SAKXGB (Self-Attention with K-Fold XGBoost) model, we conducted a comprehensive comparative study against established machine learning and deep learning baselines. All models were trained and evaluated on the same multilingual sarcasm-labeled corpus, utilizing identical preprocessing pipelines (ALDEA framework) and data partitioning strategies to ensure fair comparison.

5.2.1 Baseline Models

The comparative analysis includes the following baseline approaches:

Traditional Machine Learning Models:

- Logistic Regression
- Naïve Bayes
- Support Vector Machine (SVM)
- Decision Tree
- Random Forest
- K-Nearest Neighbors (KNN)

Deep Learning Baseline:

- **BERT-base-uncased (fine-tuned)**

All traditional models were implemented using scikit-learn with default hyperparameters, while BERT was fine-tuned using the Hugging Face Transformers library with standard configurations.

5.2.2 Evaluation Metrics

Model performance was assessed using F1-score as the primary metric, given the potential class imbalance in sarcasm detection tasks. Additionally, relative training time was measured to evaluate computational efficiency, normalized with respect to Logistic Regression as the baseline (1×).

5.2.3 Results and Discussion

Table 1 presents the comparative performance of all evaluated models. The results demonstrate that the proposed SAKXGB model achieves an F1-score of 87.00%, substantially outperforming all traditional machine learning approaches. Among the baseline models, Random Forest [34] achieved the highest F1-score (85.50%), followed by Decision Tree (83.23%) and SVM (80.93%). The KNN classifier [35] exhibited the poorest performance at 68.00%, likely due to the high-dimensional nature of textual features and sensitivity to feature scaling.

Table 1. Performance and Efficiency Comparison of Sarcasm Detection Models

Model	F1-Score (%)	Relative Training Time
Logistic Regression	80.83	1×
Naïve Bayes	77.23	1×

Model	F1-Score (%)	Relative Training Time
Support Vector Machine	80.93	2×
Decision Tree	83.23	2×
Random Forest	85.50	3×
K-Nearest Neighbors	68.00	1.5×
BERT-base-uncased	89.24	8×
Proposed SAKXGB	87.00	2×

Note: Relative training time is normalized with respect to Logistic Regression (1×). BERT training time represents end-to-end fine-tuning on the dataset.

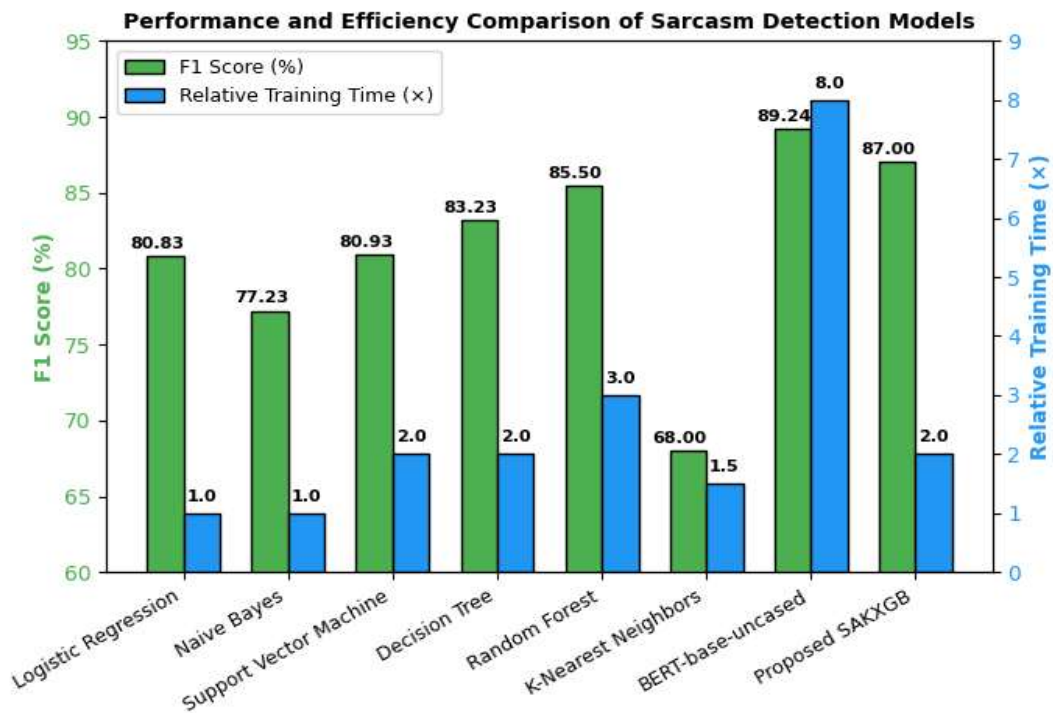


Figure 3. Comparison of Sarcasm detection models with respect to F1 score and relative training time

The BERT-base-uncased model achieved the highest F1-score of 89.24%, establishing the upper bound for performance on this task. However, this performance gain comes at a considerable computational cost, requiring approximately eight times the training time of Logistic Regression and four times that of the proposed SAKXGB model. This computational overhead stems from the full transformer architecture fine-tuning process, which involves updating millions of parameters across multiple attention layers.

The SAKXGB model bridges the performance gap between traditional machine learning and deep learning approaches, achieving 97.5% of BERT's F1-score while requiring only 25% of its training time. This efficiency gain is attributed to the hybrid

architecture design: SAKXGB leverages frozen pre-trained BERT embeddings for contextualized representation learning, eliminating the need for gradient propagation through the transformer layers during training. The custom self-attention mechanism then selectively weights sarcasm-relevant features, and XGBoost performs efficient gradient boosting for final classification

VII. FUTURE SCOPE

The proposed model SAKXGB demonstrates strong performance by using the combination of BERT embeddings, custom self-attention and XGBoost classification, reducing the training time as compared to BERT model and outperforming traditional models. This research lays a strong foundation for future research work in sarcasm detection. The future work could incorporate multimodal settings to detect sarcasm in text, audio, and video. The data set can be further expanded to include other low level regional languages that can improve NLP tasks across cultural contexts. Transformer architectures like mBERT and XLM-RoBERTa that are trained on large datasets of multiple Indian and foreign languages can further enhance contextual feature extraction in multilingual sarcasm detection leading to improved results.

REFERENCES

- [1] Joseph, Thomas. "Natural language processing (NLP) for sentiment analysis in social media." *International Journal of Computing and Engineering* 6.2 (2024): 35-48.
- [2] Chen, Shijie, Yu Zhang, and Qiang Yang. "Multi-task learning in natural language processing: An overview." *ACM Computing Surveys* 56.12 (2024): 1-32.
- [3] Gardazi, Nadia Mushtaq, et al. "BERT applications in natural language processing: a review." *Artificial Intelligence Review* 58.6 (2025): 1-49.
- [4] Meng, Jiana, et al. "Sarcasm detection based on BERT and attention mechanism." *Multimedia Tools and Applications* 83.10 (2024): 29159-29178.
- [5] Javed, Tayyaba, Muhammad Asif Nouman, and Rushna Zahid. "BERT model adoption for sarcasm detection on Twitter data." *VFAST Transactions on Software Engineering* 12.3 (2024): 177-198.
- [6] Bharati, Rajesh, et al. "SVM-based Sarcasm Detection System: NLP Using Heuristic Approach." *2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*. IEEE, 2024.
- [7] Starbuck, Craig. "Logistic regression." *The fundamentals of people analytics: With applications in R*. Cham: Springer International Publishing, 2023. 223-238.
- [8] Sharma, Surbhi, and Nisheeth Joshi. "A fusion approach to detect sarcasm using NLTK models BERT and XG Boost."
- [9] Bohacek, Matyas, and Michal Bravansky. "When XGBoost outperforms GPT-4 on text classification: A case study." *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*. 2024.
- [10] Pandey, Rajnish, et al. "A hybrid convolutional neural network for sarcasm detection from multilingual social media posts." *Multimedia Tools and Applications* 84.16 (2025): 15867-15895.
- [11] Pandey, Rajnish, and Jyoti Prakash Singh. "BERT-LSTM model for sarcasm detection in code-mixed social media post." *Journal of Intelligent Information Systems* 60.1 (2023): 235-254.
- [12] Meng, Jiana, et al. "Sarcasm detection based on BERT and attention mechanism." *Multimedia Tools and Applications* 83.10 (2024): 29159-29178.
- [13] Mienye, Ibomoiye Domor, Theo G. Swart, and George Obaido. "Recurrent neural networks: A comprehensive review of architectures, variants, and applications." *Information* 15.9 (2024): 517.

- [14] Orvieto, Antonio, et al. "Resurrecting recurrent neural networks for long sequences." International Conference on Machine Learning. PMLR, 2023.
- [15] Šandor, Daniel, and Marina Bagić Babac. "Sarcasm detection in online comments using machine learning." Information Discovery and Delivery 52.2 (2024): 213-226.
- [16] Ali, Rasikh, et al. "Deep learning for sarcasm identification in news headlines." Applied Sciences 13.9 (2023): 5586.
- [17] Anan, Ramisa, et al. "Interpretable bangla sarcasm detection using bert and explainable ai." 2023 IEEE 13th annual computing and communication workshop and conference (CCWC). IEEE, 2023.
- [18] Aftan, Sulaiman, and Habib Shah. "A survey on bert and its applications." 2023 20th Learning and Technology Conference (L&T). IEEE, 2023.
- [19] Szeghalmy, Szilvia, and Attila Fazekas. "A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning." Sensors 23.4 (2023): 2333.
- [20] Ren, Yafeng, et al. "A knowledge-augmented neural network model for sarcasm detection." Information Processing & Management 60.6 (2023): 103521.
- [21] Pal, Moumita, and Rajesh Prasad. "Sarcasm detection followed by sentiment analysis for Bengali language: neural network & supervised approach." 2023 International Conference on Advances in Intelligent Computing and Applications (AICAPS). IEEE, 2023.
- [22] Ganganwar, Vaishali, et al. "Sarcasm and humor detection in code-mixed Hindi data: A survey." International Conference on Computing and Machine Learning. Singapore: Springer Nature Singapore, 2024.
- [23] Indira Priyadarshini, Y., et al. "Sarcasm Detection in Hindi Social Media Text Using HindiBERT-LSTM and XGBoost Models." International Conference on Emerging Research in Computing, Information, Communication, Artificial Intelligence and Machine Learning. Singapore: Springer Nature Singapore, 2024.
- [24] Biswas, Susmoy, et al. "BanglaSarc3: A benchmark dataset for Bangla sarcasm detection from social media to advance Bangla NLP." Data in Brief (2025): 111953.
- [25] Lora, Sanzana Karim, et al. "Ben-Sarc: A self-annotated corpus for sarcasm detection from Bengali social media comments and its baseline evaluation." Natural Language Processing 31.2 (2025): 674-699.
- [26] Chakraborty, Sovon, et al. "An Analytical Review of Preprocessing Techniques in Bengali Natural Language Processing." IEEE Access (2025).
- [27] Prajapati, Abhinav, et al. "Sarcasm Detection in English-Hindi Code-Mixed Text Data." 2024 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT). Vol. 1. IEEE, 2024.
- [28] Tejasvi, Koti, et al. "Sarcasm Detection for Hindi English Code Mixed Twitter Data." International Journal for Research in Applied Science and Engineering Technology 11.3 (2023): 159-164.
- [29] Kumar, Akshi, et al. "Hybrid deep learning model for sarcasm detection in Indian indigenous language using word-emoji embeddings." ACM Transactions on Asian and Low-Resource Language Information Processing 22.5 (2023): 1-20.
- [30] Sultana, Zinnia, et al. "Multilingual Sarcasm Detection Using Deep Learning and Transformed Based Model." 2025 2nd International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM). IEEE, 2025.
- [31] Mercioni, Marina Adriana, and Stefan Holban. "A brief review of the most recent activation functions for neural networks." 2023 17th International Conference on Engineering of Modern Electric Systems (EMES). IEEE, 2023.
- [32] Islam, Muhammad, and Muhammad Azhar. "Sarcasm Detection in Multilingual Text through Embedding-Enhanced Language Models: BERT Variants." 2024 26th International Multi-Topic Conference (INMIC). IEEE, 2024.

- [33] Ghous, Hamid, et al. "Navigating sarcasm in multilingual text: An in-depth exploration and evaluation." *Journal of Computing & Biomedical Informatics* (2024).
- [34] John-Otumu, A. M., et al. "A Machine Learning Approach to Irony Detection in Text Using TF-IDF and Random Forests."
- [35] Fkih, Fethi, Delel Rhouma, and Hajar Alghofaily. "A semantic approach for sarcasm identification for preventing fake news spreading on social networks." *International Journal of Information Technology* (2024): 1-19.