Leveraging XLM-RoBERTa with CNN and BiLSTM for Hinglish Toxicity Detection

Nikita Singhal, Avadhesh Yadav, Ankush, Giriraj Singh, and Ronak Kumar

Original scientific article

***Abstract*—Toxicity in online communication, particularly in code-mixed languages like Hinglish, is a growing concern across social media platforms. Hinglish, a blend of Hindi and English, is widely used in informal online conversations, making it challenging for traditional toxicity detection models to accurately identify harmful content. This issue is compounded by the limited availability of resources and models specifically trained to handle Hinglish. This work presents the XLM-RoBERTa-CNN-BiLSTM (XCB) model, a novel architecture for toxicity detection in Hinglish on various social media platforms. This work compares the XCB model with the SOTA models mBERT, XLM-RoBERTa (XLM-R), and Indic-BERT. It was made on three publicly available datasets: Constraint, Facebook, and HASOC. The XCB model achieved macro F1 scores of 0.81, 0.73, and 0.82 and inference times of 0.24 s, 0.48 s, and 0.22 s on the Constraint, Facebook, and HASOC datasets, respectively. XCB not only outperforms existing romanized Hinglish models but also matches the macro F1 scores of existing SOTA multilingual models, requiring only half the training time—with extremely low inference times unlike the existing state-of-the-art models, thus making it a much more efficient candidate for large-scale real-time toxicity detection in Hinglish.**

***Index Terms*—Toxicity Detection, Hinglish, Code-Mixed Language, XCB Model, Real-Time Moderation, Multi-Lingual Models, Efficiency**

I. Introduction

Toxic, harmful, and offensive content has proliferated as an issue across social media platforms in recent years. This type of content, which also is hate speech, cyberbullying, and a form of harassment, can result in some serious issues and ramifications for all of us, such as violence, emotional damage, and even worse, the social climate we live in becoming intolerant and divided. It is increasingly becoming difficult to detect such toxicity in languages that are code-mixed, such as Hinglish (a blend of English and Hindi), as traditional language models fail to understand the hybrid forms of languages. Detecting toxicity in Hinglish is important for making online realms safer and more inclusive, where users can engage without having to worry about finding any harmful content.

Hinglish is a hybrid language, which refers to the usage of Hindi and English languages, spoken mainly in informal online spaces such as social media, forums, and messaging platforms, and specifically targets the professional community as an urgent need for a viable toxicity detection model. Conventional models usually operate on either or one language and thus are incapable of operating on code-mixed data. In addition, toxic content detection in Hinglish needs models to capture the subtleties of both Hindi and English, as well as the cultural meaning behind some words. The growing volume of code-mixed text on the internet further complicates this problem, making it imperative to develop specialized models that can process these hybrid linguistic inputs efficiently.

This work introduces the Hinglish toxicity detection task and the XCB model, an efficient new architecture for Hinglish toxicity detection with competitive performance. XCB is evaluated against SOTA multilingual models (mBERT and XLM-R) to show that it achieves comparable macro F1 scores with a dramatic reduction in training time. By addressing the challenge of toxicity detection in code-mixed content, our model is an effective approach to dynamic, large-scale analysis of toxicity in online spaces. It is hoped that this work can be used as a practical solution and integrated into content moderation systems to detect and filter toxic, harmful language and improve digital interactions, especially in the case of Hinglish.

The primary contributions of this work are

1) Introducing a novel model architecture, the XCB model, for mitigating toxic content in the Hinglish language on the internet in real time.
2) Evaluating the performance of the XCB model on three publicly available Hinglish datasets and comparing its results with transformer models using metrics such as training time, inference time, and F1 score.

The structure of the paper is organized as follows: Section II covers the related works, while Section III presents the proposed methodology. The experimental setup and results are discussed in Section IV, followed by a review of challenges and future research directions in Section V. Finally, the paper concludes in Section VI.

II. Related Works

However, recent years have seen improvements in identifying toxic content in Hinglish—that mix of Hindi and English that is increasingly finding its way to social media platforms. The aforementioned challenges of code-mixed languages have compelled researchers to experiment with different machine learning (ML) and deep learning (DL) methodologies.

Manuscript received July 2, 2025; revised July 18, 2025. Date of publication October 20, 2025. Date of current version October 20, 2025. The associate editor prof. Maja Braović has been coordinating the review of this manuscript and approved it for publication.

Authors are with the Department of Computer Engineering, Army Institute of Technology, Pune, Maharashtra, 411015, India (e-mails: {ngupta, avadheshyadav_21144, ankush_21074, girirajsingh_21053, ronakkumar_21068}@aitpune.edu.in).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0133

1845-6421/12/2025-0133 © 2025 CCIS

Anjum et al. (1) explains the subtleties of online toxicity detection and limits of existing ML-based approaches and mentions how newer generation model-based techniques are needed to understand Hinglish properly. In order to alleviate the lack of big labeled datasets for Hinglish, Yadav et al., targeting Hindi-English social media material, (2) suggested a multilingual hate speech detection method. To categorize hate speech in code-mixed data, their research explored both conventional and deep learning techniques, such as Bi-LSTM. The findings showed that supervised models could be applied to Hinglish and indicated that big language models or zero-shot approaches may be investigated further for improved performance on languages with limited resources.

Biradar et al. (3) developed the transformer-based (TrB) interpreter and feature extraction model on the foundation of the DNN for Hinglish hate speech classification. Their method demonstrated an improved accuracy of 73% compared to the SOTA approaches and insight about TrB models for the code-mixed dataset. Similarly, Hakim et al. (4) used IndoBERTweet, a version of BERT trained with Indonesian Twitter data, and BiLSTM and CNN architectures to detect hate speech. They showed that on a task-sensitive corpus, language-specific models might bring about the need to uncover the social media language and its specificities. In addition, Miran and Abdulazeez (5) took a review of DL techniques for toxic speech detection with insights on the different evolution of different models and how they apply across a range of different languages, including code-mixed ones like Hinglish.

Singh et al. (6) examines moderating toxic comments in YouTube that are in Hinglish, utilizing DL models and feature extraction methods and noting the challenges contributed by the existence of misspelled offensive terms and obfuscation techniques for code-mixed languages. Varade and Pathak (7) surveyed toxic speech detection techniques used on social media with the analysis of a range of datasets and methodologies while including the application of LSTM models built with word embeddings to detect hate speech in Hinglish. Sharma et al. (8) improved Hindi–English code-mixed hate speech detection by fine-tuning multilingual transformers (mBERT, XLM-RoBERTa) on annotated datasets and augmenting with synthetically generated code-mixed text. Their study demonstrated that transformer-based models can effectively detect toxicity in Hinglish and suggested that future work could explore larger pre-trained Hinglish-specific models or few-shot learning to boost performance.

The novelty of the proposed XCB model is that it is made for the Hinglish language. The XCB model is leveraging the TrB embeddings along with deep learning models to detect the toxicity in the Hinglish language accurately and quickly. The XCB model is unique since it uses only the embeddings from the XLM-R model, not the entire transformer model, and puts those TrB embeddings into deep learning models. CNN, BiLSTM, and, just like GloVe and Word2Vec, are used. The previous research done on combining transformer architecture with deep learning models has not explored the results in Hinglish, which is specifically focused on in this paper. Also, previous research done did not give any importance to inference times of the models or the efficacy of the model for real-time detection use cases, which has been thoroughly discussed in this paper.

This paper is unique in that it gives importance to inference time. This paper is an attempt to present a model that can be used in real-time environments for detecting toxicity on the web. For example, the XCB model can be used in mobile devices or web browser extensions to mitigate toxic content in real time. The model can be integrated into apps like WhatsApp to prevent toxic content in local languages.

Although TrB models have been shown to deliver strong performance in Hinglish toxicity detection, they often require significant computational power and are therefore difficult to deploy in real-time. To this end, a new architecture called the XCB model is presented that achieves competitive performance while being quickly trained. The performance of XCB is evaluated with respect to mBERT, XLM-R, and IndicBERT on three publicly released datasets, namely Constraint, Facebook, and HASOC. The findings highlight XCB's ability to achieve comparable F1 scores while significantly reducing training time, making it a practical choice for large-scale deployment in social media moderation.

III. Methodology

A. Baseline Models

Three multilingual baseline models were used, which are mBert, XLM-R, and IndicBERT.

mBERT is a modification of the BERT model trained on a Wikipedia dataset from 104 different languages, so it has become a strong model for multilingual NLP tasks (9). While mBERT is not implicitly trained towards cross-lingual (CL) tasks, it has been demonstrated to transfer knowledge between languages effectively that share a script. Many previous language models have used BERT because of its common use and popularity in the literature as a strong baseline for text classification, NER, and sentiment analysis, and hence, a possible strong baseline to be implemented for detecting hostility and hate speech in code-mixed multilingual datasets.

XLM-R, an extension of RoBERTa, for multilingual text. It was trained on 2.5 TB of data from CommonCrawl covering 100 languages, after which it bested mBERT on many CL benchmark evaluations (10). It eliminates the need for training independent embeddings for each language, as well as it performs much better than XLM in zero-shot and low-resource settings. Due to its robust multilingual contextual understanding, it serves as an excellent baseline for hostility detection in both Hindi and English-Hindi code-mixed data.

IndicBERT is a language-agnostic and lightweight TrB model for Indic languages. Though mBERT and XLM-R are pre-trained on a wider collection of languages, IndicBERT is specifically optimized on Indian languages, stating exactly 12 languages, wherefore making it more fitting for low-resource and code-mixed scenarios (11). Trained using a large Indian language text dataset known as IndicCorp, this language model fits classification, sentiment analysis, and offensive speech detection use cases very well. Due to its focus on Indic languages, it provides a more contextually rich understanding for the task of Hindi-based hostility detection.

B. Proposed Methodology

XCB is an architecture for efficient and effective toxicity detection, especially in Hinglish text. The architecture consists of three powerful architectures combined: XLM-R, CNN, and BiLSTM. These components enable the model to capture rich contextual information in code-mixed text, widely found on social media platforms. The model is called the XCB model, with the design taking into account performance and computational efficiency, allowing for real-time processing while maintaining a strong level of accuracy. The architecture of the XCB model is displayed in Figure 1.

1) XLM-R Embeddings: The model is based on XLM-R, a SOTA pre-trained multilingual language model capable of handling text data in over 100 languages, including Hindi and English. It produces contextually rich embeddings for each individual token in the input sentence. This includes the definitions of terms as well as the context in which they appear; hence, these embeddings are vital for code-mixed sentences in particular, where the definitions of words can greatly differ based on their adjacent usage.

The pre-trained embeddings are extracted and used to form a fixed embedding matrix where the weights are frozen and not updated further. This decision saves on computation while enabling the model to take advantage of learned representations of language without the need for problem-specific feature extraction. As the first layer, the embedding layer converts input sequences into a three-dimensional tensor (batch size × sequence length × embedding dimension), the output of which is provided as inputs to the two parallel branches of the model.

2) CNN Branch: The first part of the model applies several CNN layers over the embedded inputs. CNNs are excellent at identifying spatial and local hierarchies in data. In textual data, CNNs have proven effective for detecting essential phrases, slang, blocks of abusive words, and other short toxic patterns, which correlate poorly with sequential analysis (12).

It uses 4 layers of 1D convolutions followed by max pooling operations. Each layer applies multiple filters (kernel size) of size 3 with the same padding to keep the dimension of the input the same and to enable learning of n-gram features efficiently. The max pooling layers take the most prominent elements in each context and shrink the two-dimensional array, making the model more resistant to slight variations in phrasing or spelling—a commonality of Hinglish text.

The last convolution and pooling layer flattens the final output into a one-dimensional vector, and then it goes through a dense layer compression and refining those extracted features. This branch ends with a softmax layer, which outputs the probabilities of the classes.

3) BiLSTM Branch: Similar to the CNN pathway, the second branch of the model feeds into a BiLSTM layer. LSTM networks are a specific type of RNN that can learn long-term dependencies and contextual patterns present in forming sequences, which is imperative to capture sentiment and intent that permeates long phrases or full sentences. Thus, for each point in the sequence, it has access to both previous context and upcoming context, bi-directionally. This is important in making sense of confusing or sarcastic Hinglish sayings, where meaning can rely on knowing what precedes and succeeds a word (13).

The output of the BiLSTM is subjected to a dense layer with leaky ReLU activation, which is followed by a dropout layer that randomly sets 10% of the neurons to zero during training. This dropout is important for avoiding overfitting when using relatively small or noisy datasets. Its output is then fed through its own softmax classification layer, similar in structure to that of the CNN branch (14).

4) Weighted Fusion of CNN and BiLSTM Outputs: The output from the two branches is averaged using a weighted average after both branches independently process the embedded input. The fusion operation is performed in the end, which takes fixed weights for each branch, such as 50% from CNN output and 50% from BiLSTM output, and merges the prediction scores. These weights are determined empirically by performance and reflect the tendency for this task that, as can be seen in the outputs, local feature patterns (which are handled by CNN) are sometimes overrepresented against long-range dependencies (15). The fusion weight ablation study charts for Constraint, Facebook, and HASOC datasets are shown in Figures 2, 3, 4, and 5 respectively. We can see that the fusion weight 0.5 for CNN and BiLSTM gives highest performance across the three datasets.

This strategy provides the model dual advantages of both local and global textual features, which preserve an overall information of the input text. Individual output is limited to a form of multi-modal fusion between both outputs, so instead of loading both into a purely concatenate layer or a separate architecture for each, the model effectively generalizes better and protects itself against the heterogeneous lexicology used in Hinglish.

IV. Experimental Setup

This section discusses in detail the experiments conducted in this research and their results and analysis.

A. Datasets

This section describes the datasets and the preprocessing steps used for these datasets.

1) Dataset Description: Three datasets of Hinglish were used for our research work:Constraint (16), Facebook (17), and HASOC dataset (18). The class statistics of these datasets are shown in Table I.

TABLE I
Dataset Class Distribution

Dataset	Class Statistics
Constraint	Toxic - 1330 Neutral - 4398 Total - $5728 \approx 6k$
Facebook	Toxic - 9725 Neutral - 2274 Total - $11,999 \approx 12k$
HASOC	Toxic - 2469 Neutral - 2194 Total - $4663 \approx 5k$

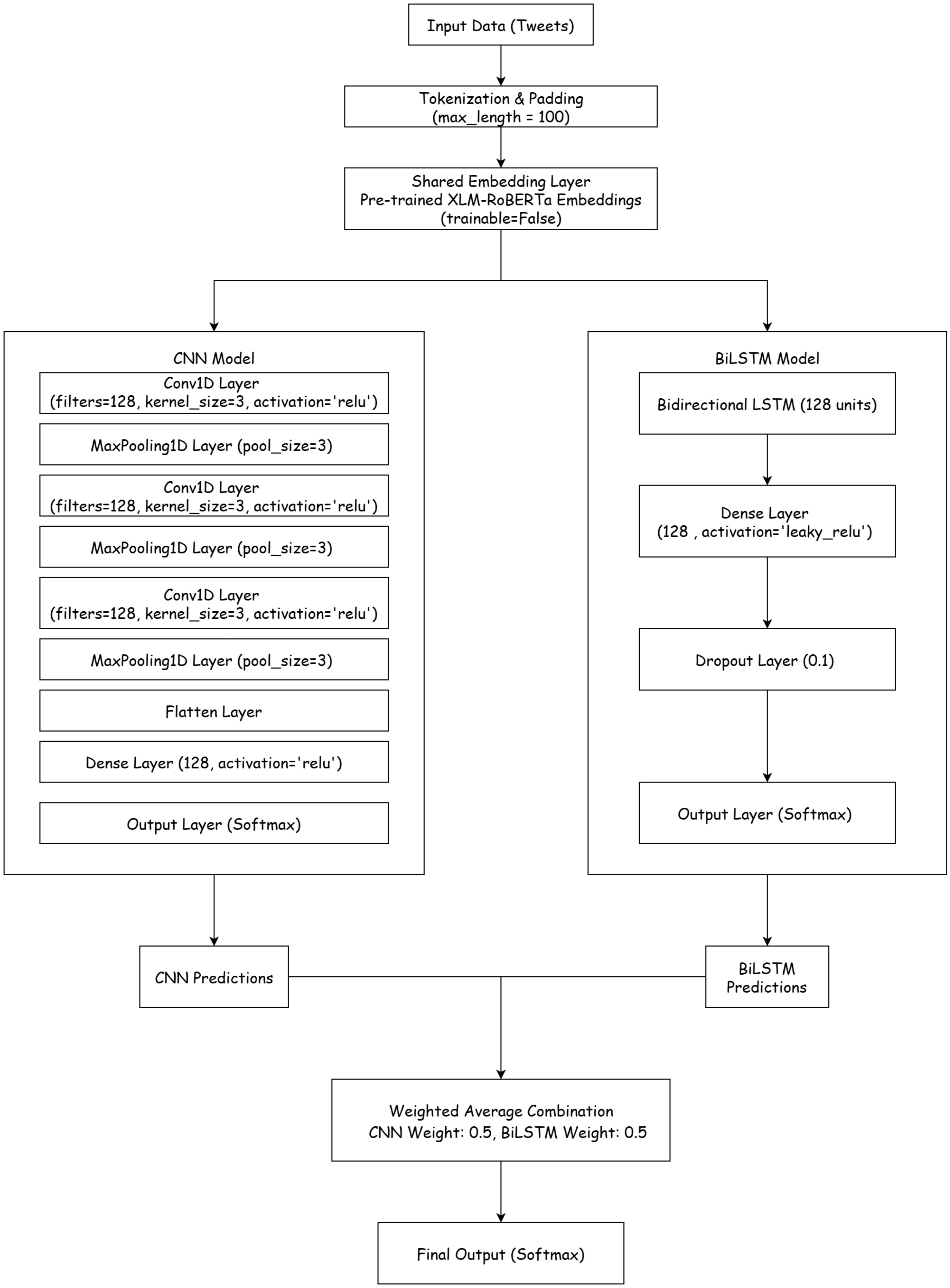

Input Data (Tweets)
Tokenization & Padding
(max_length = 100)
Shared Embedding Layer
Pre-trained XLM-RoBERTa Embeddings
(trainable=False)
CNN Model
Conv1D Layer
(filters=128, kernel_size=3, activation='relu')
MaxPooling1D Layer (pool_size=3)
Conv1D Layer
(filters=128, kernel_size=3, activation='relu')
MaxPooling1D Layer (pool_size=3)
Conv1D Layer
(filters=128, kernel_size=3, activation='relu')
MaxPooling1D Layer (pool_size=3)
Flatten Layer
Dense Layer (128, activation='relu')
Output Layer (Softmax)
BiLSTM Model
Bidirectional LSTM (128 units)
Dense Layer
(128 , activation='leaky_relu')
Dropout Layer (0.1)
Output Layer (Softmax)
CNN Predictions
BiLSTM Predictions
Weighted Average Combination
CNN Weight: 0.5, BiLSTM Weight: 0.5
Final Output (Softmax)

Fig. 1. XCB Model Architecture

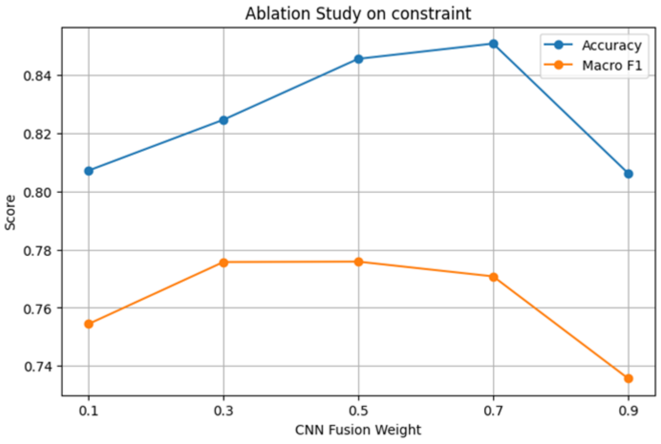


Fig. 2. Fusion weight ablation study on Constraint dataset

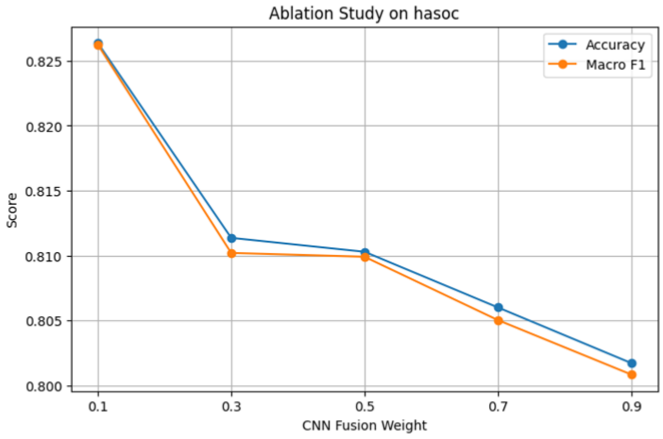


Fig. 4. Fusion weight ablation study on HASOC dataset

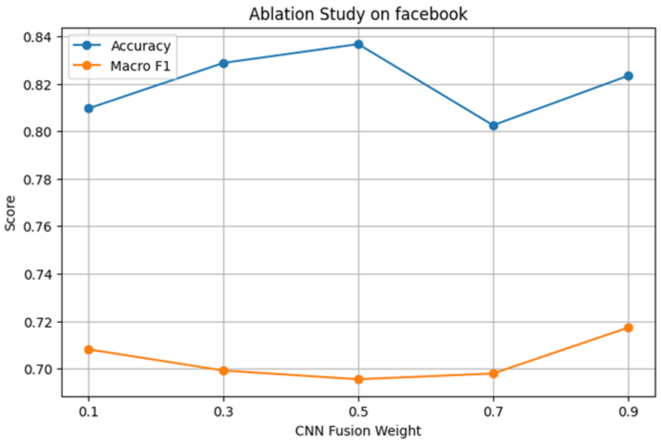


Fig. 3. Fusion weight ablation study on Facebook dataset

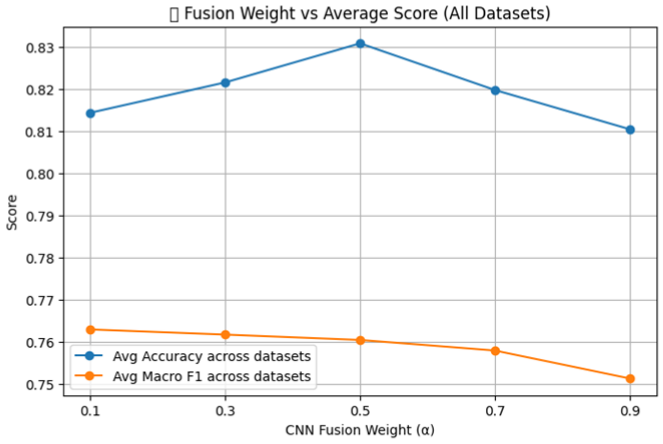


Fig. 5. Fusion weight ablation study average scores for all datasets

2) Dataset Preprocessing: All offensive, hate, and related categories were merged into a single ’toxic’ class during the preprocessing stage. Thus, the dataset is of two classes: toxic and neutral, and this suits well for our toxicity detection task. The datasets were preprocessed using the following steps:

- Lowercasing: Convert the text to lowercase to maintain consistency and prevent recognizing the same word in different cases as separate entities (i.e., ”Hello” and ”hello” are the same).
- Removing Emojis: Emojis are often non-linguistic information that may not matter for the analysis of the text in this work. They were removed using regex that identifies Unicode characters typically used for emojis.
- Removal of @Mentions: Any username or mention of @username was removed to prevent bias brought in from specific users. For the current task, these references do not add any semantic input to the text.
- Deleting URLs: URLs and hyperlinks (for example, http://example.com) were deleted; these are not beneficial for text analysis in this specific work. Such links frequently add irrelevant information that does not contribute to the task of classifying the content.
- Dropping ’RT’ (Retweets): The ’RT’ prefix, which stands for ”retweet,” was dropped as it became redundant. The added precaution is to avoid noise in the data due to retweeting of original tweets.
- Removal of Numbers: Numbers were removed given that they were not critical to the text’s semantic meaning. This is not something that numbers are necessarily meaningful for in the context of detecting hate speech or so-called offensive content.
- Punctuation Removal: All punctuation except for spaces and Devanagari script characters was removed. This is done in order to make sure that the analysis only checks the content of the text method, not the punctuation symbols used in the text, as they have nothing to do with the overall meaning of the text in this example.
- English and Devanagari are preserved: Text only preserved English (Latin alphabet) and Devanagari script. This was essential to make sure that the preprocessing was appropriate for the multilingual context of the dataset, which consists of both Hindi and English. Limiting the analysis to these scripts allowed pinpointing the linguistic features relevant to the tasks of interest.
- Removal of Extra Spaces: Any redundant spaces were removed to make sure the text is tidy and uniform, with

no extra whitespace disrupting the analysis.

B. Environment Used

Training was performed on Kaggle's GPU environment, utilizing NVIDIA Tesla T4 with 16 GB GDDR6 memory and 29 GB system RAM. The datasets were divided into 80% for training and 20% for testing purposes. The models IndicBERT, mBERT, and XLM-RoBERTa were trained with the following hyperparameters:

- Learning Rate: 1e-5—A low learning rate was adopted to enable a stable training process and to avoid overfitting, as the models are pre-trained, so only fine-tuning is needed, not training from scratch.
- Batch Size: 8—Balancing model performance considerations, a batch size of 8 was selected, considering GPU memory constraints as well.
- Number of Epochs: 10—The models were trained over 10 epochs, providing plenty of time to tune the models without too much overfitting.
- Optimizer: AdamW— Since XLM-RoBERTa is a TrB model, the AdamW optimizer was used, which separates weight decay from optimization steps of the model.
- Loss Function: Cross-Entropy Loss—used class weights for loss function to overcome class imbalance in the dataset. Since the proportions of the multi-label classes in the training data are extremely imbalanced, the inverse of the number of classes was used to balance the loss.
- Class Weights: These were computed based on the class frequencies, trying to give a higher weight to the minority class.

These hyperparameters were consistent across the baseline models (mBERT, XLM-RoBERTa, and IndicBERT), ensuring a fair comparison between the models.

We have also provided the results of the CNN and BiLSTM models trained individually on the pre-trained XLM-R embeddings. For the CNN model, the hyperparameters are as follows: 128 filters are used in each convolutional layer, with a kernel size of 3 applied to all convolutional layers. The 'ReLU' activation function is applied to each convolutional layer, and MaxPooling with a pool size of 3 is used after each convolutional layer to reduce dimensionality. For the BiLSTM model, the hyperparameters include 128 units in the LSTM layer, with the LSTM wrapped in a bidirectional wrapper to capture both forward and backward sequences. The 'LeakyReLU' activation function is used for the dense layer following the LSTM, and a dropout rate of 0.1 is applied to prevent overfitting.

For the XCB model, the hyperparameters were largely similar with some additional parameters specific to the architecture:

1) CNN Layer Settings: Four convolutional layers were used with kernel sizes of 3, and pooling was applied after each layer to reduce dimensionality.
2) BiLSTM Settings: A BiLSTM layer with 128 hidden units was used to capture sequential dependencies in the text.
3) Fusion Weights: The final output of the model is obtained by a weighted fusion of the CNN and BiLSTM branches, where cnn_weight=0.5 and bilstm_weight=0.5. This fusion weight reflects the relative importance of the CNN and BiLSTM branches in the final prediction.

C. Evaluation Metrics

Macro F1 (19) was used since the datasets are imbalanced, so macro F1 gives a correct estimation about the models' performance. The F1 score is sufficient alone since it gives us an idea of both the precision and recall. Ideally, we would want our model to give high results in both precision and recall in a balanced way, and the F1 score gives us exactly that, which is an estimation of how well the model balances precision and recall; that is, how good the model is considering both the precision and recall. Over that, we have used macro-averaged F1 so that the result is not affected by class imbalance. Macro-averaged means that it averages the performance of the model on each class, so we get an idea of the model's performance on each class taken individually. Training time per epoch and inference time of the models were also noted to know about the real-time efficiency of the models.

V. Results and Discussion

A comprehensive assessment of SOTA models for hostile content detection across multiple datasets is done with emphasis on macro F1 scores and training times. The main goal was to measure the balance between model performance (macro F1 scores) and computational efficiency, especially in large-scale real-world applications.

To note, one good contribution of this experiment is the introduction of the novel XCB model, which achieves comparable macro F1 scores as the much more complex multilingual models while using half the training time. The macro F1 scores and the per-epoch training time of the models on various datasets are illustrated in Table II and Table III, respectively. The inference time of the XCB model in seconds for the three datasets is shown in Table IV. Figure 6 displays the comparison of training time of the different models. Figure 7 displays the comparative macro F1 scores of the models across the three datasets. Figure 8 compares the inference time in seconds of the XCB model with the heavy TrB models.

TABLE II
Macro F1 Scores

Model	Constraint Dataset	Facebook Dataset	HASOC Dataset
mBERT	0.79	0.75	0.80
XLM-RoBERTa	0.82	0.75	0.83
IndicBERT	0.78	0.73	0.80
XLMR-CNN	0.78	0.71	0.78
XLMR-BiLSTM	0.80	0.72	0.82
Proposed XCB	0.81	0.73	0.82

TABLE III
TRAINING TIME PER EPOCH (SECONDS)

Model	Constraint Dataset	Facebook Dataset	HASOC Dataset
mBERT	135	276	108
XLM-RoBERTa	155	317	124
IndicBERT	108	220	86
XLMR-CNN	32	80	25
XLMR-BiLSTM	30	80	25
Proposed XCB	34	81	28

TABLE IV
INFERENCE TIME (SEC) RESULTS

Model	Constraint Dataset	Facebook Dataset	HASOC Dataset
mBERT	8.68	18.25	7.16
XLM-RoBERTa	8.12	16.92	6.28
IndicBERT	8.06	16.73	6.07
XLMR-CNN	0.11	0.16	0.10
XLMR-BiLSTM	0.22	0.41	0.19
Proposed XCB	0.24	0.48	0.22

A. *Macro F1 Score Comparison*

The macro-averaged F1 scores of the XCB model over the three datasets (Constraint, Facebook, and HASOC) exhibit a strong performance. For instance, on the Constraint dataset, XCB obtained a macro-averaged F1 score of 0.81, which is very close to the score of XLM-Roberta (0.82) and slightly better than Indic-BERT (0.78). In the same way, on the Facebook and HASOC datasets, XCB achieved comparable performance levels to the larger, more complex multilingual models, i.e., mBERT and XLM-Roberta, but still exhibited strong macro-averaged F1 scores of 0.73 and 0.82, respectively.

B. *Confusion Matrices and Error Analysis*

The confusion matrices of the model after training it on the constraint, Facebook, and HASOC datasets are shown in Figures 9, 10, and 11, respectively. The false negatives (FN) and false positives (FP) on the Constraint dataset are nearly equal. The FN in the Facebook dataset is double compared to FP. The same ratio follows for the HASOC dataset as well. This shows that the model gives less FP, which is actually good for real-life usage scenarios. The Figures 12, 13, and 14 represent some of the misclassified samples for all three Hinglish datasets taken into consideration.

C. *Inference Efficiency*

One of the most amazing features of the XCB model is its ability to give predictions efficiently. Even while achieving strong macro F1 scores, XCB had substantially lower inference times than the heavy multilingual models. The XCB model had significantly lower inference times such as 0.24 seconds on the Constraint Dataset, whereas the TrB models took 8 seconds for inference on the same dataset. This shows how fast the proposed XCB model is for employing in real time toxicity mitigation tasks. The XCB Model also had shorter training times. On the Constraint dataset, for example, XCB finished each training epoch in 34 seconds, half the time taken to complete training epochs by mBERT (135 seconds) and XLM-Roberta (155 seconds). It was found that XCB took just 81 seconds per epoch on the Facebook dataset, many times faster than mBERT (276 seconds) and XLM-Roberta (317 seconds). This significant decrease in the amount of training required is a testament to the effectiveness of XCB, which is important when computational resources are scarce or when real-time processing is desired.

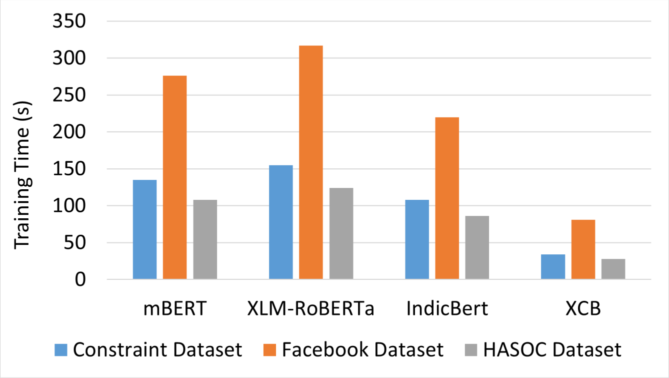


Fig. 6. Comparison of Training Time (sec)

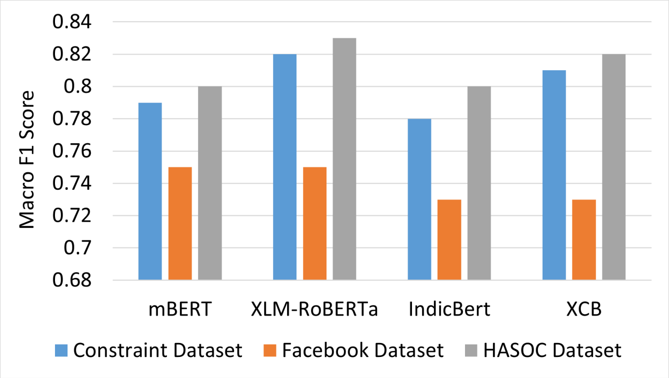


Fig. 7. Comparison of Macro F1-score

D. *Implications*

These results emphasize the practical advantages of the XCB formulation, which is found to be computationally efficient, acquiring the best performance over CPU time when compared to all existing SOTA. With a smaller than average size for a multilingual model and competitive macro F1 scores, XCB is a low-latency choice that can therefore be deployed both in resource-limited environments and as part of real-time content moderation systems.

Overall, the XCB model is a valuable solution for toxic content detection, with performance comparable with heavy multilingual models but much lower training time. Such efficiency with competitive macro F1 performance makes XCB a suitable candidate for practical real-time applications on large-scale datasets where both accuracy and computational efficiency are important.

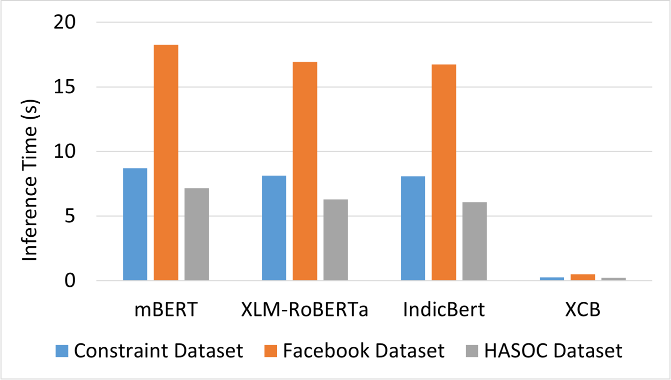


Fig. 8. Comparison of Inference Time (sec)

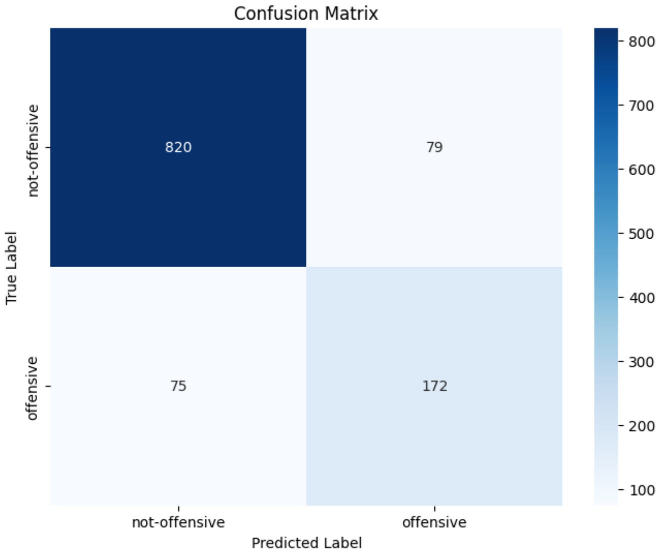


Fig. 9. Confusion matrix of model trained on Constraint dataset

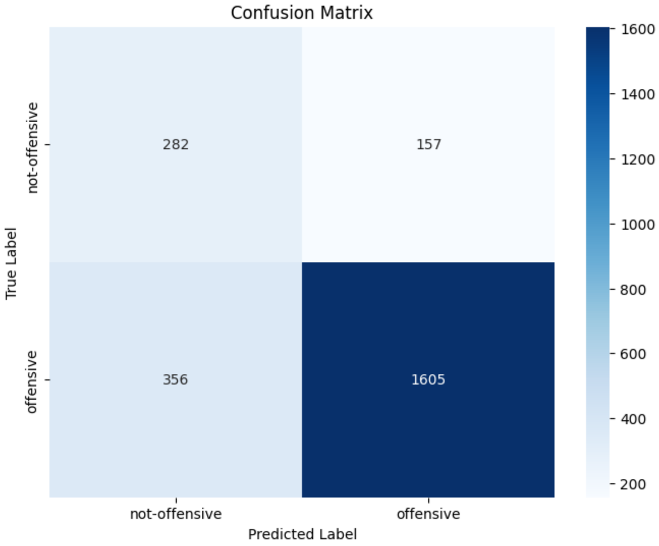


Fig. 10. Confusion matrix of model trained on Facebook dataset

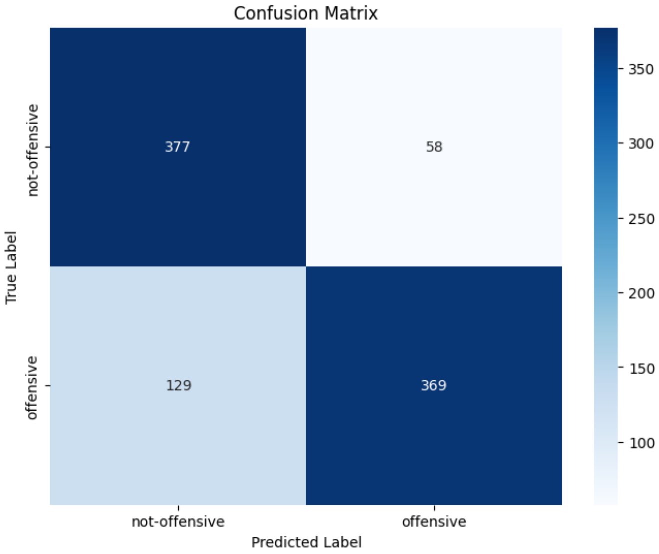
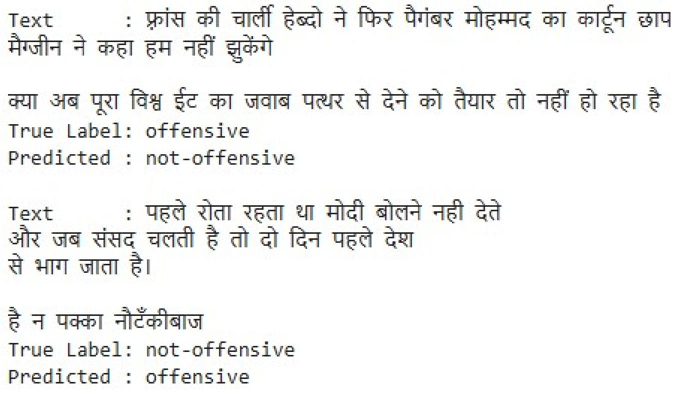


Fig. 11. Confusion matrix of model trained on HASOC dataset

VI. Challenges and Future Research Opportunities

While this study demonstrates the promising results of XCB, there are numerous possible directions for future research and improvements:

- Evaluation on Additional Datasets: The current evaluation only considers a subset of the publicly available datasets. Future work may include also validating and testing the XCB model on more datasets, especially in other languages and domains, to confirm the ability of the model to generalize and remain robust across a wider range of hostile content types.
- Fine-Tuning and Hyperparameter Optimization: While XCB performed well, there is space for improvement. Gradually improving on any existing data or conducting more thorough adjustment of the hyper-parameters could increase the accuracy and efficiency, especially with domain-specific tasks.
- Multilingual and Code-Mixed Data: This research was only limited to datasets containing Hindi and English content, and therefore, an extremely promising expansion for next projects would be to apply the XCB model to more multilingual and code-mixed data. This may include training the model to better accommodate “code switching”—a ”typical phenomenon of online and social media posts, where users might mix languages in a single post.
- Training the XCB model on other local Indian languages like Marathi, Bengali, Tamil, Telugu, Bhojpuri, etc. This will help to detect toxicity in local languages, increasing the ease of use and availability of the toxicity mitigation solution.
- Integrating explainability in hate speech is an important consideration in almost all real-world applications, since hate speech often requires an explanation for the deployment. It would also be interesting to see if explainability can be included in XCB to provide reasoning on model



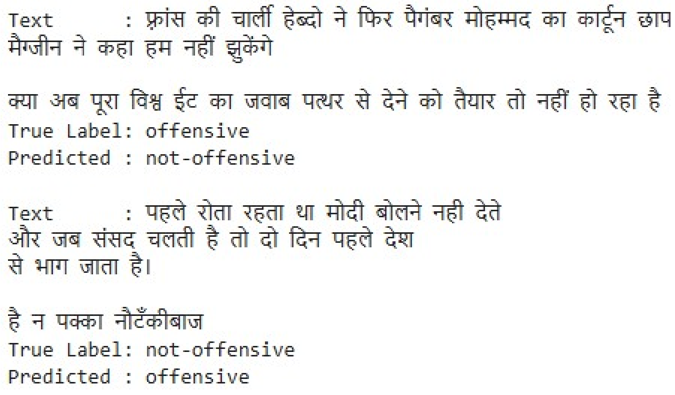
```
Text      : फ्रांस की चार्ली हेब्दो ने फिर पैगंबर मोहम्मद का कार्टून छाप
मैग्जीन ने कहा हम नहीं झुकेंगे

क्या अब पूरा विश्व ईट का जवाब पत्थर से देने को तैयार तो नहीं हो रहा है
True Label: offensive
Predicted : not-offensive

Text      : पहले रोता रहता था मोदी बोलने नही देते
और जब संसद चलती है तो दो दिन पहले देश
से भाग जाता है।

है न पक्का नौटँकीबाज
True Label: not-offensive
Predicted : offensive
```

Fig. 12. Misclassified Samples on Constraint dataset



```
Text      : फ्रांस की चार्ली हेब्दो ने फिर पैगंबर मोहम्मद का कार्टून छाप
मैग्जीन ने कहा हम नहीं झुकेंगे

क्या अब पूरा विश्व ईट का जवाब पत्थर से देने को तैयार तो नहीं हो रहा है
True Label: offensive
Predicted : not-offensive

Text      : पहले रोता रहता था मोदी बोलने नही देते
और जब संसद चलती है तो दो दिन पहले देश
से भाग जाता है।

है न पक्का नौटँकीबाज
True Label: not-offensive
Predicted : offensive
```

Fig. 13. Misclassified Samples on Facebook dataset

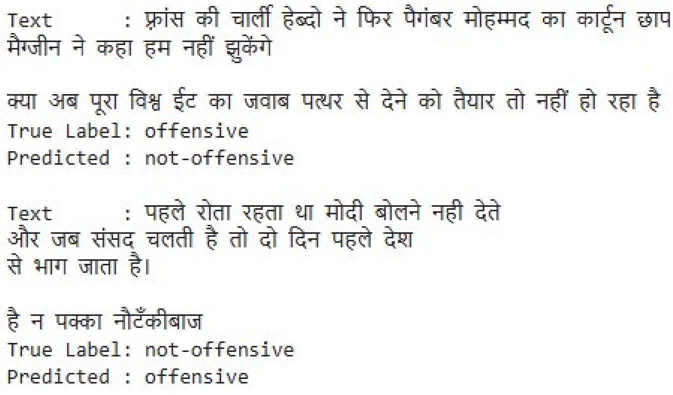
decisions, thus improving trust and adoption in practice.

- Real-Time Applications: In order to adapt the model for real-world hostile content detection on a large scale (social media platforms, online forums, etc.), future work can address optimizing XCB further with respect to latency and ease of deployment. This may include techniques such as model compression, knowledge distillation, or utilizing edge computing facilities.

Overall, XCB is a method to make detection of hostile content more accurate when traditional methods may lack and/or be inefficient. Future work discussed herein will enable further refinement and application of XCB, reinforcing the utility of the approach across diverse domains.

VII. Conclusion

This work shows how well the XCB model performs in the area of toxic content detection in a variety of datasets. Through the analysis of macro F1 scores, inference times, and training times of different SOTA models, it was demonstrated that XCB combines both performance and efficiency in a highly competitive way. Despite requiring significantly shorter training time than other models such as mBERT and XLM-Roberta, the performance of the XCB model is comparable to these models on the Constraint, Facebook, and HASOC benchmarks. The XCB model displays very low inference times, making it a viable choice for use cases where computational



```
Text      : फ्रांस की चार्ली हेब्दो ने फिर पैगंबर मोहम्मद का कार्टून छाप
मैग्जीन ने कहा हम नहीं झुकेंगे

क्या अब पूरा विश्व ईट का जवाब पत्थर से देने को तैयार तो नहीं हो रहा है
True Label: offensive
Predicted : not-offensive

Text      : पहले रोता रहता था मोदी बोलने नही देते
और जब संसद चलती है तो दो दिन पहले देश
से भाग जाता है।

है न पक्का नौटँकीबाज
True Label: not-offensive
Predicted : offensive
```

Fig. 14. Misclassified Samples on HASOC dataset

efficiency is paramount along with high accuracy, like real-time content moderation systems or operating systems with resource constraints.

Additionally, XCB is tailored to provide quick training times without compromising the prediction quality while being a good fit for scenarios where large-scale deployments, processing speeds, and scalability are important factors. The new architecture of XCB, therefore, offers a practical and scalable means towards alternative use cases such as hostile content detection systems for toxic language filtering, among others where multilingual models are currently going to be computationally expensive.

References

[1] Anjum and R. Katarya, "Hate speech, toxicity detection in online social media: a recent survey of state of the art and opportunities," *International Journal of Information Security*, vol. 23, no. 1, pp. 577–608, 2024.

[2] N. Vashistha, A. Zubiaga, and S. Sharma, "Online multilingual hate speech detection: Experimenting with hindi and english social media," *Information*, vol. 12, no. 1, p. 5, 2021.

[3] S. Biradar, S. Saumya, and A. Chauhan, "Fighting hate speech from bilingual hinglish speaker's perspective, a transformer-and translation-based approach." *Social Network Analysis and Mining*, vol. 12, no. 1, p. 87, 2022.

[4] A. N. Hakim, Y. Sibaroni, and S. S. Prasetyowati, "Detection of hate-speech text on indonesian twitter social media using indobertweet-bilstm-cnn," *2024 12th International Conference on Information and Communication Technology (ICoICT)*, pp. 374–381, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:273222115

[5] A. Z. Miran and A. M. Abdulazeez, "Detection of hate-speech tweets based on deep learning: A review," *JISA (Jurnal Informatika dan Sains)*, vol. 6, no. 2, pp. 174–188, 2023.

[6] A. Singh, K. Vaibhav, and M. Arora, "A machine learning approach for moderating toxic hinglish comments of youtube videos," in *International Conference on Data Science and Applications*. Springer, 2023, pp. 173–187.

[7] R. S. Varade and V. B. Pathak, "Detection of hate speech in hinglish language," in *Machine Learning and Information Processing: Proceedings of ICMLIP 2019*. Springer, 2020, pp. 265–276.

[8] A. Sharma, A. Kabra, and M. Jain, "Ceasing hate with moh: Hate speech detection in hindi–english code-switched language," *Information Processing Management*, vol. 59, no. 4, p. 102760, 2022.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[10] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451.

[11] D. Kakwani, A. Kunchukuttan, S. Golla, P. Bhattacharyya, M. M. Khapra, and P. Kumar, "Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. ACL, 2020, pp. 4948–4961. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.445

[12] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," *International Journal of Data Science and Analytics*, vol. 7, no. 2, pp. 171–183, 2018, originally developed for a Kaggle toxicity classification competition.

[13] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4. IEEE, 2005, pp. 2047–2052.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] A. Srivastava, "Neural approaches for code-mixed nlp in low-resource conditions," Ph.D. dissertation, IIIT Hyderabad, 2022, available at https://web2py.iiit.ac.in/research_centres/publications/download/mastersthesis.pdf.9ca77a2aeea0275c.494949545f485f5468657369732e706466.pdf.

[16] M. Bhardwaj, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, "Hostility detection dataset in hindi," *arXiv preprint arXiv:2011.03588*, 2020.

[17] R. Kumar, A. N. Reganti, A. Bhatia, and T. Maheshwari, "Aggression-annotated corpus of Hindi–English code-mixed data," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1226/

[18] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, and A. Patel, "Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages," in *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation*, 2019, pp. 14–17.

[19] S. Riyanto, S. S. Imas, T. Djatna, and T. D. Atikah, "Comparative analysis using various performance metrics in imbalanced data for multi-class text classification," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023.

Nikita Singhal is an Assistant Professor in the Department of Computer Engineering, Army Institute of Technology, Pune. She completed her Ph.D. in Computer Science and Engineering from SAGE University Indore, her MTech (Gold Medalist) in Computer Science and Engineering from Defence Institute of Advanced Technology (DIAT) Pune, and her BE (Silver Medalist) in Information Technology from RGPV Bhopal. She has 14 years of academic and research experience. She is a reviewer at various reputed conferences and journals. Her research interests include deep learning, machine learning, and image processing.

Avadhesh Yadav is a final-year undergraduate student pursuing a Bachelor's degree in Computer Engineering at the Army Institute of Technology, Pune. With a strong foundation in computer science, his research interests focus on the fields of Natural Language Processing and Computer Vision. He is particularly passionate about developing innovative solutions that bridge the gap between artificial intelligence and real-world applications.

Ankush is a final-year undergraduate student pursuing a Bachelor's degree in Computer Engineering at the Army Institute of Technology, Pune. He is driven by a fascination for decoding complex patterns and applying deep learning and data science to solve real-world challenges.

Giriraj Singh is a final-year student pursuing a Bachelor's degree in Computer Engineering at the Army Institute of Technology, Pune. His research interests include Natural Language Processing and data collection and filtering for training machine learning models. Currently, he is working on developing a browser plugin for detecting hate speech, using NLP techniques to promote a safer online environment and foster societal peace.

Ronak Kumar is a final-year Computer Engineering student at the Army Institute of Technology, Pune. His research interests lie in Natural Language Processing (NLP) and the processes of data collection and filtering for training machine learning models. He is currently developing a browser plugin aimed at detecting hate speech, leveraging NLP techniques to create a safer online space and support societal harmony.