Hinterwelt@LT-EDI 2025: A Transformer-Based Approach for Identifying Racial Hoaxes in Code-Mixed Hindi-English Social Media Narratives

Md. Abdur Rahman[1] **MD AL AMIN**[2] **Sabik Aftahee**[3] **Md Ashiqur Rahman**[1]
[1]Southeast University, Dhaka, Bangladesh
[2]St. Francis College, Brooklyn, New York, USA
[3]Chittagong University of Engineering & Technology (CUET), Chittagong, Bangladesh
{2021200000025@seu.edu.bd, alaminhossine@gmail.com,
u1904024@student.cuet.ac.bd, ashiqur.rahman@seu.edu.bd}

Abstract

This paper presents our system for the detection of racial hoaxes in code-mixed Hindi-English social media narratives, which is in reality a form of debunking of online disinformation claiming fake incidents against a racial group. We experiment with different modeling techniques on HoaxMixPlus dataset of 5,102 annotated YouTube comments. In our approach, we utilize traditional machine learning classifiers (SVM, LR, RF), deep learning models (CNN, CNN-LSTM, CNN-BiLSTM), and transformer-based architectures (MuRIL, XLM-RoBERTa, HingRoBERTa-mixed). Experiments show that transformer-based methods substantially outperform traditional approaches, and the HingRoBERTa-mixed model is the best one with an F1 score of 0.7505. An error analysis identifies the difficulty of recognizing implicit bias and nuanced contexts in complex hoaxes. Our team was 5th place in the challenge with an F1 score of 0.69. This work contributes to combating online misinformation in low-resource linguistic environments and highlights the effectiveness of specialized language models for code-mixed content.

1 Introduction

Communication has undergone massive changes since the use of social media in the current world. People all across the globe can connect with each other and social media platforms such as YouTube have made this form of communication easier than ever. Any form of communication these days has its fair share of issues. Misinformation is a prime example of its misuse, with Racial hoaxes being a type that poses a significant threat to social cohesion.

Racial hoaxes falsely link individuals or groups to crimes or incidents, perpetuate and reinforce harmful stereotypes and accelerate ethnic tensions. In India's multi-language setting, the informal blending of Hindi and English on the internet creates a code-mixed environment which makes such identification difficult because of the informal usage, linguistic complexity and limited datasets. The LT-EDI 2025 task introduces a dataset (Chakravarthi, 2020), comprising 5,105 YouTube comments in code-mixed Hindi-English, annotated as racial hoax or non-racial hoax. The dataset addresses the scarcity of resources for misinformation detection in low-resource, code-mixed settings. The critical contributions of this work are:

- Developed several machine learning, deep learning, and BERT-based models for detecting racial hoaxes in code-mixed Hindi-English YouTube comments.

- Evaluated multiple Machine Learning, Deep Learning and Transformer-based models for racial hoax detection, providing a comparative analysis to identify the most effective approach for this low-resource setting.

2 Related Works

Existing literature focuses on issues of identifying abusive content in code switched Hindi-English social media texts. Patil et al. (2023) presented HingBERT, a model pre-trained on code-mixed data, which performs better than the non domain-specific models for tasks such as hateful content detection, pointing to the relevance of domain specific models to learn linguistic nuances. For hate speech, which is related to racial hoax identification, Mazumder et al. (2024) showed that the addition of native Hate samples improved the multilingual language model as more training data, but not for subjective or sarcastic which is a general characteristic of racial hoax. S et al. (2024) used ensemble stacked model with XLM-RoBERTa model for Tamil code-mix content, with a weighted F1-score of 0.72. Their method using LSTM and GRU with multilangual embeddings may also be applied to Hindi-English

Proceedings of the Fifth Workshop on Language Technology for Equality, Diversity, Inclusion, pages 121–126
September 9, 2025 ©2025 Association for Computational Linguistics

contexts. In an attempt to account for sarcasm in racial hoaxes, Sahu and R (2024) introduced a model, based on BERT embeddings and contextual embeddings, which is able to detect sarcasm and irony within code-mixed social media data. In the case of low-resource languages like Dravidian languages, Hande et al. (2021) employed a sample of pseudo-labeling techniques to augment the dataset, and their fine-tuned ULMFiT model scored competitive results with a technique which may be used for Hindi-English racial hoax detection. Kumar et al. (2023) also investigated offensive text classification using mBERT-like transformer based models, for some Indian languages with traditional approaches and achieved competitive F1-scores up to 0.95 in Malayalam, demonstrating that transformers can capture code-mixed text well. Anbukkarasi et al. (2022) worked on Tamil-English hate speech detection using a synonym-based Bi-LSTM model and tackled standardization problems for Indian languages. These experiments demonstrate the effectiveness of multilingual embeddings, domain-specific pre-training, and data augmentation. Our future work will fine-tune models such as HingBERT on racial hoax specific datasets and gain true evaluation metrics to avoid lopsided performance.

3 Task and Dataset Description

We present our system for the shared task for the task of identifying racial hoaxes present in the code-mixed Hindi-English social media narratives (Chakravarthi et al., 2025). It addresses the growing threat of online disinformation by focusing in on racial hoaxes, false claims about a crime or incident that pin the blame on someone or a group based on race, create harm and increase discord in communities affected. We used the HoaxMixPlus dataset (Chakravarthi, 2020), a novel collection of 5,102 YouTube comments curated and annotated for this tricky phenomenon. This annotated corpus that has been pre-processed and split into training (3,060 samples), dev (1,021 samples) and test (1,021 samples), is one of the few such resources which is highly valuable for detecting this type of manipulative content in a low-resource, code-mixed linguistic environment. The comments are divided binarily (hoax/non-hoax), and their objective is to report on the performance of systems. Table 1 summarizes the data splits and overall Dataset statistics. And Figure 1 illustrates the overall class distribution across the combined Train, Dev, and Test sets. The implementation code can be accessed via the GitHub repository[1].

Class	Train	Dev	Test
Total Samples	3060	1021	1021
Not Racial Hoax	2319	774	774
Racial Hoax	741	247	247

Table 1: Dataset Split Statistics per Class

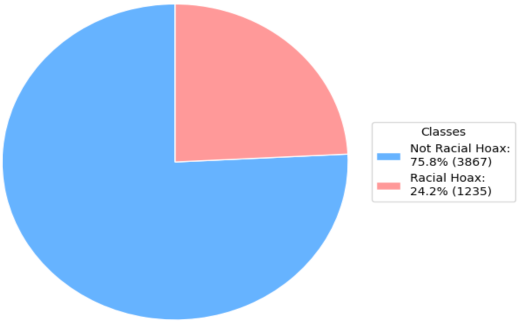


Figure 1: Overall Class Distribution

4 Methodology

This section describes the methodologies used for the the Text Classification from code-mixed Hindi-English Social Media Data. A wide variety of models ranging from traditional machine learning ones to deep learning architectures and modern transformer-based methods were evaluated and systematically fine-tuned through hyperparameter optimization to improve classification performance. The architectural frameworks utilized for the detection of racial hoaxes is illustrated in Figure 2

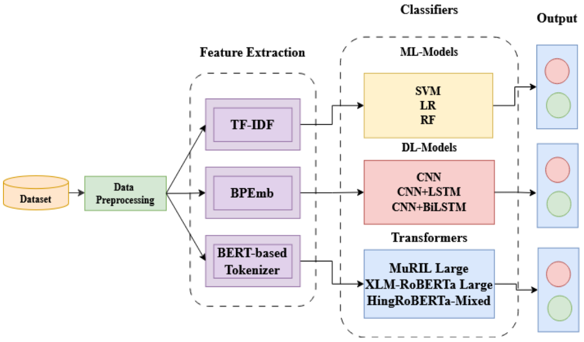


Figure 2: Schematic process for Detecting Racial Hoaxes

[1] https://github.com/borhanitrash/LT-EDI-2025

4.1 Data Preprocessing

We used the official shared task dataset, which consists of 3060 training, 1021 validation, and 1021 test samples. Our pre-processing steps were the same across all of our models: we normalized column names by removing punctuation and spaces, and replaced any missing `clean_text` fields with empty strings. In the context of ML models, these cleaned texts were directly served in successive feature extraction. For our DL architectures, texts were tokenized with BPEmb, followed by padding or truncating to a maximum length of 128 (with dedicated ID for padding). For transformer-based models, we used their AutoTokenizers, padding or truncating the sequences to 128 tokens, and creating attention masks.

4.2 Feature Extraction

We developed different feature extraction approaches for the particular needs of both modeling paradigms. For classical ML models, we used Scikit-learn's[2] TF-IDF vectorization to map preprocessed texts into numerically representative terms, including unigrams and bigrams, with a maximum of 20 000 features. The DL models all used 100-dimensional subword embeddings obtained from BPEmb(Heinzerling and Strube, 2018) and fine-tuned during training in order to better capture task-specific semantics from the input strings. For our transformer-based systems, we took the contextualized word embedding obtained from their pre-trained tokenizers. Classification was typically determined by the final hidden state of the special `[CLS]` token, which was then fed through a linear layer.

4.3 Machine Learning Models

To provide strong baseline performances, we tested a variety of traditional machine learning classifiers, all using the above described TF-IDF features. Our set consisted of three groups of models: SVM, LR, and RF. We used a linear kernel for the SVM. Both models SVM and LR, were set with a regularization parameter C of 1.0 and balanced class weight considerations to avoid data imbalance, and LR used liblinear solver and 200 iterations. The Random Forest classifier had 150 estimators, `balanced_subsample` class weights and specific tree controls. The Key hyperparameters for these models are detailed in Table 2.

Table 2: Key hyperparameter settings for the ML models.

Classifier	Parameter	Value
SVM	kernel	`linear`
	C	1.0
	class_weight	`balanced`
Logistic Regression	solver	`liblinear`
	C	1.0
	class_weight	`balanced`
	max_iter	200
Random Forest	n_estimators	150
	max_depth	`None`
	min_samples_split	5
	min_samples_leaf	2
	class_weight	`balanced_subsample`

4.4 Deep Learning Models

We experimented with several deep learning architectures, and all of them used 100-dimensional BPEmb subword embeddings. These models were a Convolutional Neural Network (CNN), a CNN along with a single-layer Bidirectional LSTM (CNN-LSTM), and a CNN along with a two-layer Bidirectional LSTM with attention (CNN-BiT-LSTM). All these model architectures shared a CNN part that used 128 filters of size [3,4,5] and had a dropout layer with factor of 0.5 after the convolutional layers and before the final output layer. AdamW optimizer was employed to train all the deep learning models. Table 3 lists some important training and RNN-specific hyperparameters for DL Models.

Table 3: Key hyperparameter settings for DL models. LR denotes Learning Rate, BS denotes Batch Size and Att denotes Attention Mechanism

Model	RNN Configuration	LR	Max Epochs (Patience)	BS
CNN	-	1e-3	50 (10)	32
CNN-LSTM	1xBiLSTM(128)	1e-3	50 (10)	32
CNN-BiLSTM	2xBiLSTM(128) + Att	1e-3	50 (10)	32

4.5 Transformer-Based Models

Our primary approach leveraged pre-trained multilingual Transformer models (Vaswani et al., 2017) which are known to be capable of capturing complex contextual cues and long-range dependencies via self-attentive mechanisms. We adopted models pre-trained on the Hugging Face Transformer library[3], and fine-tuned them to make the representations tuned to our classification task. The models used were MuRIL-large (Khanuja et al., 2021), as it performs well for an Indian language as well as for the transliterated text; XLM-RoBERTa-large (Con-

[2] https://scikit-learn.org/stable/

[3] https://huggingface.co/transformers

neau et al., 2019), a strong cross-lingual model; and HingRoBERTa-mixed (Nayak and Joshi, 2022) by L3Cube-Pune, developed for Hindi-English code-mixed text. For fine-tuning, input texts were tokenized with the tokenizer of the corresponding model, while the sequence was padded or truncated to a maximum length of 128 tokens. A standard sequential classification head was used on top of the transformer encoder. Optimizer was AdamW (Loshchilov and Hutter, 2017) with a linear learning rate scheduler followed by 10% warm-up steps to the entire training steps. We used CrossEntropyLoss as the loss function. To prevent overfitting and obtain robust generalization, early stopping was used, with a patience of 3 epochs, based on the macro F1-score on the validation set. Learning rates and weight decay were tuned for each model for performance. A full list of such and other important hyperparameters such as batch size and maximum epochs are given in Table 4.

Table 4: Key hyperparameters for Transformer-Based models. LR: Learning Rate, WD: Weight Decay, BS: Batch Size. Max EP (Patience) indicates maximum epochs with early stopping patience.

Model	LR	WD	BS	Max EP (Patience)
MuRIL-large	1e-5	0.01	16	10 (3)
XLM-RoBERTa-large	3e-6	0.05	16	10 (3)
HingRoBERTa-mixed	3e-6	0.05	16	10 (3)

5 Result Analysis

Table 5 summarizes the performance metrics Precision, Recall, and F1 Score of all evaluated classifiers on the test set, and disaggregates by model categories.

Model	P	R	F1
ML Models			
LR	0.6613	0.6686	0.6646
SVM	0.6625	0.6629	0.6627
RF	0.7600	0.6352	0.6564
DL Models			
CNN	0.6957	0.6610	0.6734
CNN+LSTM	0.6582	0.6449	0.6505
CNN+BiLSTM	0.6709	0.6800	0.6750
Transformer Models			
MuRIL-large	0.6906	0.6535	0.6662
XLM-RoBERTa-large	0.7242	0.7361	0.7296
HingRoBERTa-mixed	**0.7674**	**0.7382**	**0.7505**

Table 5: Performance Comparison of All Models

The Machine Learning (ML) models Logistic Regression (LR) and Support Vector Machine (SVM) exhibited close performance levels, with F1 Scores of 0.6646 and 0.6627, respectively. Random Forest (RF) reached relatively high precision (0.7600) but its recall value (0.6352) was not high enough, F1 Score being 0.6564.

CNN+BiLSTM was the best-performing model in the DL features group with an F1 Score of 0.6750. This slightly outperformed the plain CNN (0.6734), meaning a marginal gain from having bidirectional context, since CNN+LSTM scored a bit worse (0.6505). Although the performance of these DL models was generally superior to those based on ML, the gain was marginal.

The performance improvement was most significant for Transformer-based models. Our HingRoBERTa-mixed system was evidently the best performing model out of all, with a better F1 Score of 0.7505, with good Precision (0.7674) and Recall (0.7382). XLM-RoBERTa-large also proved to show strong performance with an F1 Score of 0.7296. MuRIL-large's F1 Score (0.6662) was similar to ML and DL high performers.

For all, Transformer-based models, especially HingRoBERTa-mixed, yield considerably better results than DL and traditional ML techniques. This demonstrates the high-level contextual knowledge and expressive capability of large pre-trained language models in addressing the complexities of this problem. A detailed error analysis is provided in Appendix A.

6 Conclusion

The paper introduced a complete framework for detecting racial hoaxes in code-mixed Hindi-English social media narrative. From the experimental results, it can be concluded that transformer-based models such as the HingRoBERTa-mixed are particularly effective, with the best F1 score at 0.7505, significantly exceeding conventional machine learning and deep learning algorithms. The error analysis shows difficulties in identifying implicit bias and the context subtleties in novel sophisticated hoaxes. These results suggest the necessity of custom language model for code-mixes in combating this pressing problem of online disinformation. In the future, we would like to improve identification of implicit racial sentiments, including more context features and developing ensemble methods to achieve model adaptability across such diversity of linguistic expressions in these low-resource environments.

Limitations

Despite HingRoBERTa-mixed's promising performance, several limitations need to be addressed. Our models fail to detect implied bias and context-sensitive clues in advanced racial hoaxes that use sarcasm or veiled inference. The code-switched nature of the corpus brings in its own set of challenges in representing the cross-lingual linguistic nuances. The amount of data used (5,102) also limits the capability of our model to provide generalization among different types of racial hoaxes. The large gap between the amount of true and false positives (140 and 107) shows current methods are still far too conservative in finding actual racial hoaxes, with a danger of overlooking toxic content in real-world applications.

Acknowledgments

This work was supported by Southeast University, Bangladesh.

References

S. Anbukkarasi, Anbukkarasi Sampath, and S. Varadhaganapathy. 2022. Deep learning-based hate speech detection in code-mixed tamil text. *IETE Journal of Research*.

Bharathi Raja Chakravarthi. 2020. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Shanu Dhawale, Saranya Rajiakodi, Sajeetha Thavareesan, Subalalitha Chinnaudayar Navaneethakrishnan, and Durairaj Thenmozhi. 2025. Overview of the shared task on detecting racial hoaxes in code-mixed hindi-english social media data. In *Proceedings of the Fifth Workshop on Language Technology for Equality, Diversity and Inclusion*, Italy. Fifth Conference on Language, Data and Knowledge (LDK2025).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Adeep Hande, Karthik Puranik, Konthala Yasaswini, R. Priyadharshini, Sajeetha Thavareesan, Anbukkarasi Sampath, Kogilavani Shanmugavadivel, Durairaj Thenmozhi, and Bharathi Raja Chakravarthi. 2021. Offensive language identification in low-resourced code-mixed dravidian languages using pseudo-labeling. *arXiv preprint arXiv:2105.03983*.

Benjamin Heinzerling and Michael Strube. 2018. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, and 1 others. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

R. Prasanna Kumar, G. Bharathi Mohan, S. Ajith, R Sudarshan, and Vinitha Sree. 2023. Empowering multilingual insensitive language detection: Leveraging transformers for code-mixed text analysis. In *Proceedings of the International Conference on Network, Multimedia and Information Technology (NMITCON)*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Debajyoti Mazumder, Aakash Kumar, and Jasabanta Patro. 2024. Improving code-mixed hate detection by native sample mixing: A case study for hindi-english code-mixed scenario. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Aryan Patil, Varad Patwardhan, Abhishek Phaltankar, Gauri Takawane, and Raviraj Joshi. 2023. Comparative study of pre-trained bert models for code-mixed hindi-english data. In *Proceedings of the 8th IEEE International Conference for Convergence in Technology (I2CT)*.

Vishak Anand S, Ishwar Prathap, Deepa Gupta, and Aarathi Rajagopalan Nair. 2024. Enhancing hate speech detection in tamil code-mix content: A deep learning approach with multilingual embeddings. In *Proceedings of the 5th IEEE Global Conference for Advancement in Technology (GCAT)*.

Pinaki Sahu and Nagaraja S R. 2024. Enhancing sentiment analysis using bert-hybrid model for detection of irony and sarcasm in code-mixed social media. *International Journal of Scientific Research in Engineering and Management*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A Error Analysis

To perform a comprehensive analysis of our system, we executed a detailed error analysis of the system and examined the predictions of our best model HingRoBERTa-Mixed.

A.1 Quantitative Analysis

The Confusion matrix for HingRoBERTa-Mixed model on test set is shown in Figure 3. The model is also capable of identifying 'Non-Hoax' cases (704 true negatives) with authentic content. But the primary problem is how to spot 'Racial Hoax' content. Of particular interest, 107 'Racial Hoax' cases were incorrectly predicted as 'Non-Hoax' (false negatives) by HingRoBERTa-Mixed, which reveals that some subtle hoax signs are occasionally overlooked. Furthermore, 70 'Non-Hoaxes' were erroneously labeled as 'Racial Hoax' (false positives). Although the 140 true positives we obtained for 'Racial Hoax' with HingRoBERTa-Mixed, the elevated false negative rate of True Class in this case indicates that more improvements are required. The disparity indicates a conservative bias in fact-checking on how content is labeled as a racial hoax, which might be due to either subtleties of the hoax or peculiar linguistic expressions in the training data.

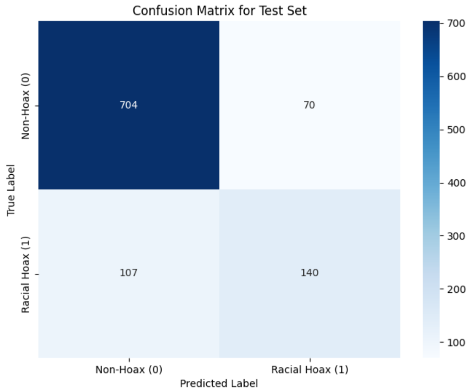
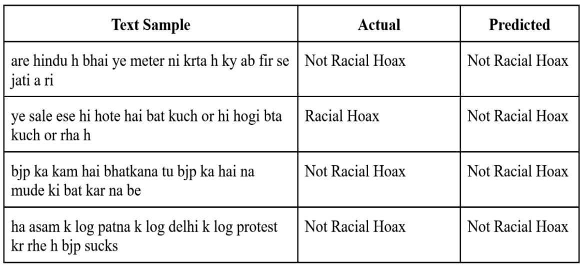


Figure 3: Confusion matrix of the proposed model (fine-tuned HingRoBERTa-Mixed) on test set

A.2 Qualitative Analysis

A qualitative analysis of HingRoBERTa-Mixed misclassified samples (shown in Figure 4) is instructive. Many of the false negatives are things that involve difficult-to-nail-down linguistic cues or contextual things that the model just struggles with. For example, the mis-classified *Racial Hoax* text sample 2 (Predicted: Not Racial Hoax), uses slang and indirect insinuation, with no direct triggers the model might prioritize. This suggests that HingRoBERTa-Mixed may not fully capture the implicit hate or sarcasm conveyed in more sophisticated hoaxes. On the flip side, false positives often include texts with a high degree of negative sentiment or those containing terms that, though controversial, aren't necessarily hoaxes. The model may over-sensitize to terms co-occurring with hoaxes in training, causing misattributions. These results also show that it is a very hard task for HingRoBERTa-Mixed to differentiate between true hoaxes and regular negative news, especially based on the content level clues only. Such subtle differences need to be further investigated in the future.



Text Sample	Actual	Predicted
are hindu h bhai ye meter ni krta h ky ab fir se jati a ri	Not Racial Hoax	Not Racial Hoax
ye sale ese hi hote hai bat kuch or hi hogi bta kuch or rha h	Racial Hoax	Not Racial Hoax
bjp ka kam hai bhatkana tu bjp ka hai na mude ki bat kar na be	Not Racial Hoax	Not Racial Hoax
ha asam k log patna k log delhi k log protest kr rhe h bjp sucks	Not Racial Hoax	Not Racial Hoax

Figure 4: Few examples of predictions produced by the proposed HingRoBERTa-Mixed model on the Test Set