Hinterwelt@LT-EDI 2025: A Transformer-Based Detection of Caste and Migration Hate Speech in Tamil Social Media

MD AL AMIN[1*] **Sabik Aftahee**[2*] **Md. Abdur Rahman**[3*]
Md Sajid Hossain Khan[2] **Md Ashiqur Rahman**[4]
[1]St. Francis College, Brooklyn, New York, USA
[2]Chittagong University of Engineering & Technology (CUET), Chittagong, Bangladesh
[3]Southeast University, Dhaka, Bangladesh
[4]School of Computer and Cyber Sciences, Augusta University, Georgia, USA
{alaminhossine@gmail.com, u1904024@student.cuet.ac.bd,
2021200000025@seu.edu.bd, u1904069@student.cuet.ac.bd, mdrahman@augusta.edu}

Abstract

This paper presents our system for detecting caste and migration-related hate speech in Tamil social media comments, addressing the challenges in this low-resource language setting. We experimented with multiple approaches on a dataset of 7,875 annotated comments. Our methodology encompasses traditional machine learning classifiers (SVM, Random Forest, KNN), deep learning models (CNN, CNN-BiLSTM), and transformer-based architectures (MuRIL, IndicBERT, XLM-RoBERTa). Comprehensive evaluations demonstrate that transformer-based models substantially outperform traditional approaches, with MuRIL-large achieving the highest performance with a macro F1 score of 0.8092. Error analysis reveals challenges in detecting implicit and culturally-specific hate speech expressions requiring deeper socio-cultural context. Our team ranked 5th in the LT-EDI@LDK 2025 shared task with an F1 score of 0.80916. This work contributes to combating harmful online content in low-resource languages and highlights the effectiveness of large pre-trained multilingual models for nuanced text classification tasks.

1 Introduction

The ability to communicate with anyone from any part of the globe has been enabled through social media platforms, which have optimistic advantages. Though its merits are many, social media platforms have worked as a catalyst and sometimes as an efficient medium for hate speech propagation aimed at various communities, spreading derogating remarks based on caste or migration. This is a great matter of concern which deeply threatens the social unity of India. As concerning as it is, there's a clear lack of resources to tackle this problem. Moreover, tackling the problem of detecting caste and migration related hate speeches in low resource languages like Tamil is extremely difficult due to the lack of bounded datasets, intricate language forms, and anthropological aspects, which have their own complexities. With the intention of promoting multilingual Natural Language Processing (NLP) along with ethical Artificial Intelligence (AI) gives a Tamil dataset of 7875 social media comments which were previously marked as caste/migration hate speeches with non hate speeches.

Using Exploratory Data Analysis (EDA), we have bounced protection methods and comparative analysis structures to devise informative classifiers based on the dataset attributes which exhibit 61.8:38.2 class imbalance along with other characteristics like equal text length across all classes, balancing the classification approach. Our model tries to balance the detection of hate speech using various approaches hails from the low resource and imbalanced face of the task. This particular investigation works towards creating more active digital environments by building stronger systems for Tamil hate speech detection.

The critical contributions of this work are:

- Developed several machine learning, deep learning, and BERT-based models for detecting caste and migration-related hate speech in Tamil social media comments, optimizing performance for a low-resource language setting.
- Evaluated the performance of employed models and provided a comparative analysis to identify the most effective approach for hate speech detection in Tamil.
- Conducted comprehensive EDA to characterize the Tamil dataset, revealing linguistic and statistical properties of caste and migration-related hate speech.

[*] Authors contributed equally to this work.

2 Related Works

Our work addresses a significant gap in hate speech detection for Tamil, specifically targeting casteist and migration-related content on social media, a context largely underrepresented in existing resources. We engage with both categories in a unified setting, offering detection in dataset and evaluating modern classification strategies.

While prior research has made strides in Tamil hate speech detection, most efforts have focused on offensive language in general or on single categories. Mohan et al. (2025) introduced a multimodal dataset for casteist content, but its limited size restricts scalability. Reddy et al. (2024) used ensemble classifiers combining SVM, Random Forest, and Naive Bayes, achieving promising results but highlighting challenges like ineffective POS tagging for Tamil. Deep learning-based approaches, like the CNN-BiLSTM + transformer models used by Sangeetham et al. (2024), demonstrate the value of contextual embeddings but were limited by dataset scope. Efforts like Shahiki Tash et al. (2024) focused solely on migration discourse.

Our approach builds on these foundations by exploring transformer-based models fine-tuned specifically for caste and migration hate speech. offers a valuable opportunity to evaluate classification models in a dual-category, low-resource setting, contributing an important benchmark for hate speech detection in Tamil and related languages.

Looking ahead, we aim to expand dataset coverage and evaluate techniques like domain-adaptive pre-training and cross-lingual transfer from code-mixed Hindi-English hate speech models, contributing toward building fairer, safer online spaces for marginalized communities.

3 Task and Dataset Description

We participated in the Shared Task on Caste and Migration Hate Speech Detection at LT-EDI@LDK 2025 (Rajiakodi et al., 2025, 2024). The goal was to automatically classify Tamil social media text as either ’Caste/Migration-related Hate Speech’ (label 1) or ’Non-Caste/Migration-related Hate Speech’ (label 0). The provided CSV dataset (Chakravarthi, 2020) comprised 5,512 training, 787 development, and 1,576 test instances, with a notable class imbalance favoring non-hate speech. Performance was officially evaluated using the macro F1-score. 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
Total Samples	5512	787	1576	7875
Not Caste/Migration Hate Speech	3415	485	970	4870
Caste/Migration Hate Speech	2097	302	606	3005

Table 1: Dataset Split Statistics per Class

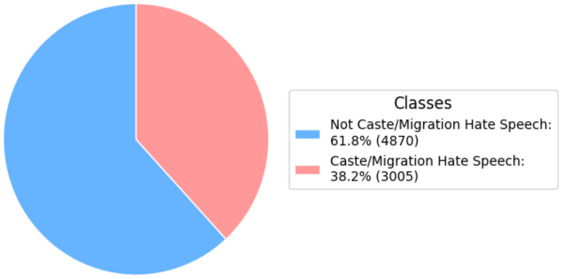


Figure 1: Overall Class Distribution

4 Methodology

Several machine learning (ML), deep learning (DL), and transformer-based models were employed to establish a robust baseline, as illustrated in Figure 2.

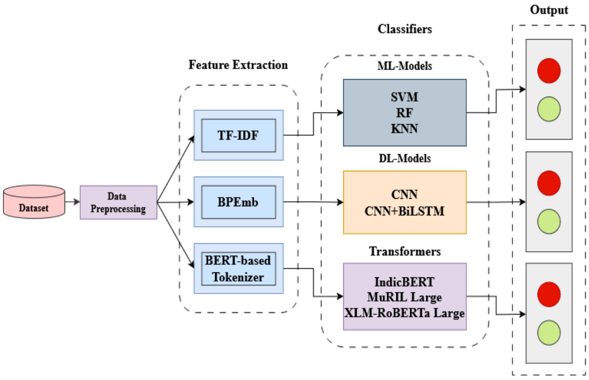


Figure 2: Schematic process for Caste and Migration Hate Speech Detection

4.1 Data Preprocessing

For our participation in the shared tasks, we utilized the officially provided datasets. A common initial data treatment step for all models involved addressing missing text entries by substituting them with empty strings. For our classical machine learning

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

approaches (SVM, RF, KNN), textual features were derived using TF-IDF vectorization, incorporating unigrams and bigrams alongside frequency-based term pruning. Deep learning architectures based on BPEmb tokenized the textual inputs, which were subsequently padded or truncated to a 256-token maximum sequence length. Our Transformer-based systems (MuRIL, IndicBERT, XLM-R) leveraged their respective AutoTokenizers for sequence preparation, also standardizing to 256 tokens with padding/truncation and generating attention masks; `keep_accents=True` was specifically employed for the IndicBERT and XLM-R tokenizers.

4.2 Feature Extraction

For our classical machine learning models, Scikit-learn's[2] TF-IDF vectorization to transformed texts into numerical features using unigrams and bigrams, with a vocabulary capped at 50,000 terms. Our deep learning architectures (CNN, CNN-LSTM, CNN+BiLSTM) employed 100-dimensional BPEmb subword embeddings (Heinzerling and Strube, 2018). We chose BPEmb because its subword segmentation approach is particularly well-suited for a morphologically rich language like Tamil. It effectively mitigates the out-of-vocabulary (OOV) problem by breaking down unknown words, misspellings, or neologisms into known, meaningful sub-units. This preserves crucial semantic information often lost by traditional tokenizers. Furthermore, BPEmb provides lightweight, pre-trained embeddings, allowing us to build strong yet computationally efficient deep learning baselines without the high overhead of a full Transformer architecture. Transformer-based systems (MuRIL, IndicBERT, XLM-R) leveraged their inherent mechanisms to generate rich, contextualized embeddings from input tokens, with the representation of the `[CLS]` token typically feeding the final classification layer.

4.3 Machine Learning Models

We benchmarked three classical machine learning approaches: Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbors (K-NN). For SVM, a linear kernel with C=1.0 was utilized. The RF employed 100 estimators, no maximum depth, a minimum of 2 samples for splits, and 1 per leaf. K-NN used 5 neighbors with distance weighting and cosine similarity. All models were trained on the TF-IDF features described previously. Table 2 details these key hyperparameters.

Classifier	**Parameter**	**Value**
SVM	kernel	linear
	C	1.0
Random Forest	n_estimators	100
	max_depth	None
	min_samples_split	2
	min_samples_leaf	1
K-NN	n_neighbors	5
	weights	distance
	metric	cosine

Table 2: Key hyperparameter settings for the ML models.

4.4 Deep Learning Models

We explored two deep learning architectures utilizing 100-dimensional BPEmb embeddings. These included a 1D Convolutional Neural Network (CNN) and a hybrid model combining the CNN with a single-layer Bidirectional LSTM (CNN-BiLSTM). Both featured a common CNN structure with 128 filters (kernels [3,4,5]) and 0.3 dropout. All models were trained using the AdamW optimizer. Key training hyperparameters are summarized in Table 3.

Table 3: Key hyperparameter settings for DL models. LR denotes Learning Rate, BS denotes Batch Size, P denotes Patience

Model	**RNN Configuration**	**LR**	**Epochs (P)**	**BS**
CNN	-	1e-4	30 (6)	32
CNN-BiLSTM	1xBiLSTM(128)	1e-4	50 (10)	32

4.5 Transformer-Based Models

We employed several pre-trained Transformer models (Vaswani et al., 2017), recognized for their proficiency in capturing complex contextual information via self-attention. Our suite included: MuRIL-large (Khanuja et al., 2021), tailored for Indian languages; IndicBERT (Kakwani et al., 2020), a model from a suite designed for various Indic languages; and XLM-RoBERTa-large (Conneau et al., 2019), a robust multilingual model. For fine-tuning, inputs were tokenized using each model's specific tokenizer, with sequences standardized to 256 tokens through padding or truncation. A standard sequence classification head was appended to the encoder. Optimization was performed using

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

AdamW (Loshchilov and Hutter, 2017), a linear learning rate scheduler with 10% warmup steps, and CrossEntropyLoss. Early stopping, guided by validation macro F1-score with a patience of 4 epochs, was used to prevent overfitting. Table 4 outlines the key hyperparameters.

Table 4: Key hyperparameters for the fine-tuned MuRIL-large model (best model).

Hyperparameter	Value
Learning Rate	1e-5
Per Device Batch Size	8
Max Epochs (Patience)	15 (4)
Max Sequence Length	256
Loss Function	CrossEntropyLoss
Optimizer	AdamW
Weight Decay	0.01

5 Result Analysis

Table 5 presents the evaluation metrics of Precision, Recall, and F1 Score (macro average) for all evaluated models on the test set, categorized by their respective model families: Machine Learning (ML), Deep Learning (DL), and Transformer-based models.

Table 5: Performance Comparison of All Models (Macro Average)

Model	Precision	Recall	F1 Score
ML Models			
SVM	0.7204	0.6842	0.6906
Random Forest	0.8073	0.7646	0.7756
KNN	0.7606	0.7497	0.7539
DL Models			
CNN	0.7748	0.7488	0.7566
CNN+BiLSTM	0.7668	0.7559	0.7602
Transformer Models			
IndicBERT	0.7387	0.7401	0.7394
XLM-RoBERTa-large	0.8016	0.7915	0.7957
MuRIL-large	**0.8157**	**0.8046**	**0.8092**

The Machine Learning (ML) models demonstrated moderate performance, with Random Forest (RF) achieving the highest Accuracy (0.8001) and Macro F1 Score (0.7756) among them. RF notably balanced precision and recall better than both SVM and KNN, which showed weaker recall for the minority class (Class 1). SVM, while providing decent precision for Class 0, struggled on recall for Class 1 (0.4983), yielding a lower Macro F1 of 0.6906. KNN delivered a relatively competitive performance (F1: 0.7539) with balanced precision and recall values across both classes. Within the Deep Learning (DL) category, CNN+BiLSTM slightly outperformed the standalone CNN, with a Macro F1 Score of 0.7602 versus 0.7566. This suggests that integrating bidirectional sequence modeling into the CNN framework provides a marginal advantage in capturing sequential dependencies. Nonetheless, both DL models surpassed most ML baselines, particularly in balancing performance across both classes, though Random Forest remained competitive. Transformer-based models exhibited the strongest results overall. MuRIL-large achieved the highest overall test set Accuracy of 0.8223 and a Macro F1 Score of 0.8092. XLM-RoBERTa-large closely followed with an Accuracy of 0.8096 and a Macro F1 of 0.7957. IndicBERT, while trailing behind its Transformer peers, still outperformed most ML and DL models with a Macro F1 Score of 0.7394. Notably, both MuRIL-large and XLM-RoBERTa-large consistently demonstrated superior balance in precision and recall across both classes, indicating their effectiveness in addressing class imbalance challenges. Ultimately, Transformer-based architectures,particularly MuRIL-large and XLM-RoBERTa-large substantially outperformed both traditional Machine Learning and Deep Learning models. These results emphasize the advantage of leveraging large pre-trained multilingual models for nuanced, context-rich text classification tasks, affirming their suitability for complex applications such as hate speech detection in code-switched or multilingual social media content. A detailed error analysis is provided in Appendix A.

6 Conclusion

In this study, we addressed the challenging task of detecting caste and migration-related hate speech in Tamil social media content. We systematically evaluated a range of machine learning, deep learning, and Transformer-based models. Our findings indicate that Transformer architectures, particularly MuRIL-large, achieve superior performance, demonstrating the efficacy of large pre-trained multilingual models for this nuanced task. While these models show promise, error analysis reveals challenges with implicit hate and colloquialisms, suggesting avenues for future work in enhancing contextual understanding and incorporating cultural nuances to further improve detection accuracy and contribute to safer online environments.

Limitations

Our study, while demonstrating the efficacy of Transformer models, faces limitations. The dataset, though valuable, may not fully capture the diverse and evolving nature of hate speech, including implicit or coded language prevalent in Tamil social media. The models, particularly MuRIL-large, struggled with nuanced cultural references and sarcasm, indicating a need for enhanced contextual understanding. Furthermore, the class imbalance, despite mitigation efforts, might still influence model bias. Future work should explore larger, more diverse datasets and techniques to imbue models with deeper socio-cultural awareness for more robust hate speech detection.

Acknowledgments

The authors gratefully acknowledge *Centro Interuniversitario di Ricerca Scienze Umane e Sociali e Intelligenza Artificiale* (ELIZA) – University of Naples 'L'Orientale' for its support in covering the registration costs, which enabled their participation.

References

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.

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*.

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)*.

Divyanshu Kakwani, Anoop Kunchukuttan, Ankur Bapna, Mitesh M. Khapra Pratyush Kumar, and Pushpak Bhattacharyya. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Simran Khanuja, Anoop Kunchukuttan, Pushpak Bhattacharyya, and Mitesh M. Khapra Pratyush Kumar. 2021. MuRIL: Multilingual Representations for Indian Languages. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1933–1946, Online. Association for Computational Linguistics.

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

Jayanth Mohan, Spandana Reddy Mekapati, B Premjith, Jyothish Lal G, and Bharathi Raja Chakravarthi. 2025. A multimodal approach for hate and offensive content detection in tamil: From corpus creation to model development. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(3):1–24.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Rahul Ponnusamy, Prasanna Kumaresan, Sathiyaraj Thangasamy, Bhuvaneswari Sivagnanam, and Charmathi Rajkumar. 2024. Overview of shared task on caste and migration hate speech detection. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 145–151, St. Julian's, Malta. Association for Computational Linguistics.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Rahul Ponnusamy, Shunmuga Priya MC, Prasanna Kumar Kumaresan, Sathiyaraj Thangasamy, Bhuvaneswari Sivagnanam, Balasubramanian Palani, Kogilavani Shanmugavadivel, Abirami Murugappan, and Charmathi Rajkumar. 2025. Findings of the shared task on caste and migration hate speech detection. In *Proceedings of the Fifth Workshop on Language Technology for Equality, Diversity and Inclusion*, Italy. Fifth Conference on Language, Data and Knowledge (LDK2025).

A Ankitha Reddy, Ann Maria Thomas, Pranav Moorthi, and B Bharathi. 2024. Ssn-nova@lt-edi 2024: Pos tagging, boosting techniques and voting classifiers for caste and migration hate speech detection. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 233–237. Association for Computational Linguistics.

Saisandeep Sangeetham, Shreyamanisha C Vinay, Kavin Rajan G, Abishna A, and B Bharathi. 2024. Algorithm alliance@lt-edi-2024: Caste and migration hate speech detection. pages 254–258. Association for Computational Linguistics.

M. Shahiki Tash, Z. Ahani, M. T. Zamir, O. Kolesnikova, and G. Sidorov. 2024. Lidoma@lt-edi 2024: Tamil hate speech detection in migration discourse. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 184–189. Association for Computational Linguistics.

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

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

A Error Analysis

To conduct a comprehensive evaluation of our system, we performed a detailed error analysis focusing on the predictions of our best-performing model, MuRIL-Large, on the test set for Tamil hate speech detection.

A.1 Quantitative Analysis

The confusion matrix for the MuRIL-Large model on the test set is presented in Figure 3. The model shows strong competence in correctly identifying the Not Hate category, achieving 855 true negatives. However, the primary concern lies in accurately detecting Hate instances. The model misclassified 165 Hate samples as Not Hate (false negatives), revealing its occasional difficulty in capturing the more implicit or contextually nuanced hateful content present in Tamil social media text. On the other side, the model incorrectly labeled 115 Not Hate instances as Hate (false positives). This suggests that certain non-hateful messages possibly containing charged words or negative sentiment might trigger the classifier's decision boundary. Despite these misclassifications, the model correctly predicted 441 Hate cases (true positives), showcasing a reliable detection capacity overall. However, the relatively elevated number of false negatives compared to false positives suggests a moderate conservative bias, where the model errs on the side of caution in labeling messages as Hate. This conservativeness might stem from nuanced expression styles, code-mixed Tamil-English usage, or indirect hate rhetoric in the data. These patterns indicate areas for targeted model refinement, such as improved contextual embeddings or fine-tuning with domain-specific corpora enriched in subtle hate cues. Figure 3 shows the confusion matrix of the proposed model (fine-tuned HingRoBERTa-Mixed) evaluated on the test set.

A.2 Qualitative Analysis

A qualitative review of MuRIL-Large's misclassified samples provides further insights into the model's limitations. Many of the false negatives consist of text samples employing colloquial, sarcastic, or indirect phrasing, often involving culturally specific insults or contextual cues that are challenging for a language model to discern without broader world knowledge or socio-cultural awareness. For instance, some messages utilize Tamil slang or implicit derogatory references that do not contain overtly hateful keywords but would be instantly recognizable to native speakers as offensive. These instances suggest that while MuRIL-Large is competent at identifying explicit hate, it struggles with coded language, sarcasm, or satire, which often require understanding not just of language structure but also of local idioms and cultural context. On the flip side, false positives typically include posts with strong negative sentiment, political criticism, or emotionally charged expressions, which, although not hate speech, might contain emotionally loaded terms co-occurring frequently with Hate labels in the training set. The model appears to overfit to these high-risk tokens, triggering misclassifications. These findings highlight the complexity of hate speech detection in Tamil, especially in a code-mixed, informal social media setting. Future work should explore integrating context-aware mechanisms, sarcasm detection modules, and cultural knowledge resources to improve the model's ability to parse implicit and nuanced hate expressions more effectively.

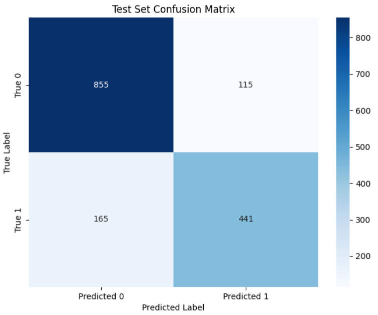
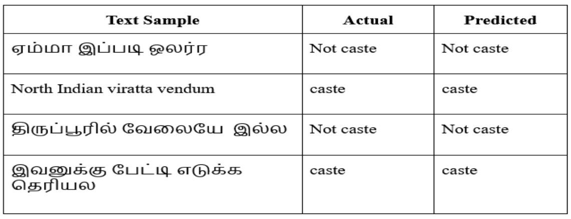


Figure 3: Confusion matrix of the proposed model (fine-tuned MuRIL-Large) on test set

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



Text Sample	Actual	Predicted
ஏம்மா இப்படி ஒலர்ர	Not caste	Not caste
North Indian viratta vendum	caste	caste
திருப்பூரில் வேலையே இல்ல	Not caste	Not caste
இவனுக்கு பேட்டி எடுக்க தெரியல	caste	caste

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