A TriFusion Model for Offensive Language Detection in Dravidian Code-Mixed Text

Raksha Adyanthaya[1,†], Rathnakara Shetty P[2,*]

[1]Yenepoya Institute Of Arts, Science, Commerce, and Management. Yenepoya (Deemed to be University), Mangalore, India
[2]Yenepoya Institute Of Arts, Science, Commerce, and Management. Yenepoya (Deemed to be University), Mangalore, India

Abstract

The social media boom in multilingual communities has resulted in the proliferation of offending and harmful content, particularly in low-resourced Dravidian languages, where the overall phenomenon of code-mixing with English is widespread. TriFusion is a multi-model pipeline to detect offensive language in Kannada, Tulu, Tamil, and Malayalam. The TriFusion system combines TFIDF with Ensemble classifiers, IndicBERT with MLP, and XLM-RoBERTa with XGBoost, with further improvement of majority-vote fusion. In the DravidianCodeMix 2025 shared task, the proposed system ranked Rank 3rd in Malayalam (macro F1-Score = 0.750), 5th in Tulu (macro F1-Score = 0.730), 7th in Kannada (macro F1-Score = 0.375), and 7th in Tamil (macro F1-Score = 0.369). On the whole, the findings indicate that transformers can be generalized well across languages, and Ensemble models are more effective under low-resource settings.

Keywords

Offensive Identification, Offensive Language, Indian Languages, Code-Mixed, Classification, Dravidian Language, Low-Resource Language, Multilingual, XLM-Roberta, Ensemble, TriFusion.

1. Introduction

Offensive speech is communication that irritates, offends, or hurts feelings. These kinds of content are spreading hate, fostering violence, and targeting individuals or groups based on identity on platforms such as Facebook, Twitter, YouTube, and Instagram. Filtering and censoring the violence is thus necessary regarding the development of healthy internet communication and protection of vulnerable groups [1]. The offensive language identification task [2] focuses on under-resourced code-mixed Dravidian languages Kannada, Tulu, Malayalam, and Tamil, which are widely used in South Indian social media. Such languages are usually written in Roman script, and their intermingling with English is challenged by variation in spellings, syntactic ambiguity, and restricted resources. The task divides user-generated content into such categories as Not_offensive, Offensive_Targeted_Insult_Group, Offensive_Targeted _Insult_Individual, Offensive_Untargeted, and Not-Kannada/Malayalam/Tamil and Tulu. The dataset is a collection of manually annotated, code-mixed comments of native speakers [3]. The obstacles are class imbalance, informal language, and vague intent. To counter this, the study suggests a holistic solution involving a combination of traditional machine learning and transformer models. Preprocessing was done with language-aware normalization followed by Term Frequency Inverse Document Frequency (TFIDF) using Ensemble classifiers such as Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), Multi Layer Perceptron(MLP), Support Vector Machine(SVM), K-Nearest Neighbour (KNN) and Random Forest (RF) using Voting Classifier. In parallel, contextual embeddings were extracted from two transformer models, IndicBERT and XLM-RoBERTa, and used with MLPClassifier and XGBoost, respectively.The TriFusion system further enhances majority-vote fusion by combining TFIDF with Ensemble classifiers, IndicBERT with MLP, and Cross-Lingual Model RoBERTa(XLM-RoBERTa) with XGBoost.This diverse combination of lexical and semantic representations ensured robust performance across varying comment styles and code-mixed inputs.

Forum for Information Retrieval Evaluation, December 17-20, 2025, India
*Corresponding author.
rakshaadyanthaya11@gmail.com (R. Adyanthaya); rathnakar.sp@gmail.com (R. S. P)

2. Related Work

Automated moderation systems have been required by the propagation of abusive and hateful speech on social media platforms [4]. Numerous studies have suggested various computational techniques to combat the growing prevalence of hostile language, from sophisticated deep learning and transfer learning models[5] to conventional machine learning. In the study of Adyanthaya and Shetty [6], minimal preprocessing of code-mixed data was incorporated to maintain the natural structure of multilingual inputs, following feature extraction approaches such as TFIDF and Word2Vec, ensemble classification techniques including SVM, RF, BiLSTM and mBERT were used. These techniques are highly relevant for identifying inappropriate language in noisy, code-mixed situations, even though they were mainly used for sentiment analysis. Lexical characteristics like TFIDF representations and n-grams were the mainstay of early methods, which also used classifiers like SVM, Naive Bayes, and LR. Davidson et al. [7] showed that n-gram-based TFIDF features successfully classified tweets into neutral, offensive, and hateful categories. However, these conventional approaches struggle when it came to low-resource and code-mixed languages like Kannada, where word order, spelling variants, and transliteration add complexity. Kumar et al. [8] and Talat and Hovy [9] tried to address these issues by utilizing ensemble classifiers and feature fusion like sentiment features, character- and word-level n-grams. As social media content in Indian languages, particularly code-mixed forms like Kanglish, Tanglish, and Manglish [5], has become more and more popular, several initiatives have been devoted to identifying multilingual offensive language, including HASOC [10] and DravidianCodeMix [11]. The unique linguistic difficulties are demonstrated by benchmark datasets created by Chakravarthi et al. [11] in Tamil, Malayalam, and Kannada. To improve comprehension of semantics, an increasing number of researchers are adopting transformer-based models like XLM-RoBERTa [12], IndicBERT [13], and BERT [14]. Because these models were pretrained on various multilingual datasets, they are especially well-suited to managing the intricacies of Indian languages. In contrast to Tamil and Malayalam, Tulu and Kannada are still comparatively understudied despite these developments.

3. Datasets description

The DravidianCodeMix@FIRE 2025 shared task on Offensive Language Detection, which aims to detect offensive content in code-mixed YouTube comments, is the source of the datasets used in this study [15][3]. Tamil, Malayalam, Kannada, and Tulu are the four Dravidian languages included in the datasets; the distribution of these instances is shown in Table 1.

Language	Train	Dev	Test	Total
Tamil	35,139	4,388	4,392	43,919
Malayalam	16,010	1,999	2,001	20,010
Kannada	6,217	777	778	7,772
Tulu	2,692	577	576	3,845

Table 1
Dataset statistics

4. System Description

The proposed offensive language detection system has been illustrated in Figure 1, which incorporates preprocessing, feature extraction, classification, and majority voting.

4.1. Preprocessing

To prepare the code-mixed datasets for classification of offensive language, a thorough preprocessing pipeline was implemented to reduce noise and normalize inputs.

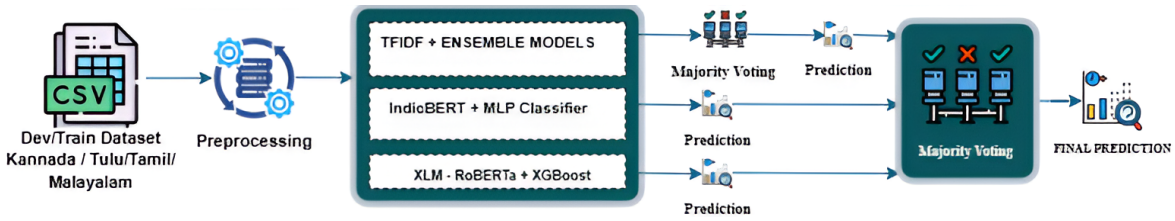

Figure 1: Proposed methodology for offensive language detection

The following are the stages:

- **Noise reduction:** Unnecessary elements that are commonly included in social media comments, including links, mentions, hashtags, and irregular symbols, are eliminated. We kept only English and language-specific characters and eliminated URLs, user mentions, hashtags, digits, and special characters.
- **Emoji Management:** The function demojize() converted emojis into their descriptive text equivalents. To preserve sentiment cues, this semantic information was retained. Emoji descriptors were then surrounded by colons, which were removed to simplify the token structure.
- **Unicode normalization:** With an emphasis on language-specific characters, we performed Unicode normalization using the IndicNormalizerFactory from the indic-nlp-library.

4.2. Baseline Ensemble Model with Traditional Classifier

The proposed system integrates a soft-voting ensemble of six traditional machine learning classifiers [16], including LR, XGB [17], MLP, RF, Support Vector Classifier(SVC), and KNN. The score of each model is a probability distribution over the target classes, which is averaged to select the most probable class based on the mean probability, therefore the ensemble can take advantage of individual model confidence and generalize across models better than through hard voting. Lowercasing, removing punctuations as well as normalizing code-mixing content are done on the input text, which then get TFIDF vectorization [6] and the limit of features was set to 50000 with the n-gram range of (1, 2). The system training and testing use each language's data separately, class labels are encoded via LabelEncoder, and the results are assessed by using Accuracy, macro F1-score (mF1-score), and weighted F1-score (wF1-score). The resulting sparse representations are the inputs to the ensemble, which makes the final prediction.

4.3. IndicBERT -based Deep Representation with MLP Classifier

The IndicBERT , a transformer multilingual model, which is optimized on Indian languages, to create dense contextual embedding of the input text. we fine-tune the representations of the [CLS] token of the last hidden layer based on the maximum sequence length of 128. These embeddings are fed into a pre-trained MLP classifier, which consists of two hidden layers having 256 and 128 nodes, respectively, an activation function of the form ReLU, and an optimizer of the form Adam, trained with up to 300 iterations. The system is trained on each language dataset respectively using LabelEncoder to encode labels and scored against the development set using Precision, Recall and macro F1-score. Extending the capacity of IndicBERT [18] to support rich semantic and syntactic features and that of MLP to estimate non-linear decision boundaries, multi-lingual classification can be performed robustly using this fusion technique.

4.4. XLM-RoBERTa Embeddings with XGBoost and Class Balancing

The XLM-RoBERTa [19] transformer model that our system uses is pre-trained on more than 100 languages, including some Indian languages, which is favorable to multilingual and code-mixed scenarios. Each input sentence has its embedding as [CLS] token (768-dimensional) extracted at the final hidden

layer, and with maximum length up to 128 tokens. Then, these embeddings get passed to an XGBoost classifier that is set to multi-class classification and that has a maximum depth of 6,200 estimators, and a learning rate of 0.1. To overcome the impact of imbalance between classes, we calculate weights of classes inversely proportional to the frequencies and use them in training. The system is optimized on development set based on the usual classification accuracy indices and the final predictions on test set are produced through mapping between the model output and the original label set. The strategy would blend the strong contextual ability of XLM-R with the gradient boosting functionality of XGBoost to be able to deal effectively with both high-dimensional embeddings and imbalanced data.

4.5. TriFusion: Majority Voting Ensemble of Three Systems

To take advantage of the complementary skill sets of our individual models, this study chose a majority voting ensemble approach. Namely, the predictions of three different systems were combined:

(i) an ensemble of a classical set of classifiers trained on the basis of TFIDF.

(ii) an IndicBERT model with a MLP classifier.

(iii) an XLM-RoBERTa model with an XGBoost classifier.

The three systems were queried in each test cases and the predicted label of the three systems per test case were used to obtain the final output label by adopting hard (majority) voting. Where no label had a majority then the mode function within the Pandas library was used to pick the most common label. The purpose of this ensemble technique is to reduce the bias of deep contextual embeddings and increase the overall robustness of approximations of the true model and achieve better overall performance in classification.The result is shown in Table 3

5. Results and Error Analysis

Table 2 presents the performance comparison of three models: TFIDF+Ensemble, IndicBERT +MLP, and XLM-R and Table 3 shows results of TriFusion model, across four languages: Kannada, Tulu, Tamil, and Malayalam. The metrics of evaluation are Accuracy, mF1-score, and wF1-score.

5.1. Results

The proposed models were tested on four Dravidian languages i.e. Tulu, Malayalam, Kannada and Tamil using three important evaluation measures such as Accuracy (2), mF1-score, and wF1-score. Such parameters combine two aspects of the prediction correctness overall and the capability of the model to manage the aspect of the imbalance of classes, which plays a significant role in offensive language detection. The findings show that transformer-based architectures are always superior to conventional ensemble models in most languages, especially in Malayalam and Tamil. These two languages had the best scores in the XLM-RoBERTa model with an mF1-score of 0.76 and 0.73 in this order, indicating that the model has a good grasp of the context of the code-mixed text. In the case of Tulu, though, TFIDF + Ensemble model made a competitive mark of 0.73 on mF1-score, meaning that classical statistical features have not yet been discarded in small or less diverse datasets. However, Kannada was more challenging with high data imbalance and morphological complexity, with TFIDF a little more successful than deep models on all three metrics. TriFusion ensemble, a sergeant that is composed of the results of TFIDF, IndicBERT, and XLM-RoBERTa based on majority voting, exhibited better language-independent generalization. It scored the best in Accuracy in Malayalam (0.92), and all mF1 and wF1-scores of all languages were constant, which narrowed the performance difference between the resource-rich and low-resource cases. In general, the comparison between these three parameters depicts that transformer-based models perform better in the case of well-represented datasets, whereas hybrid ensemble fusion offers a more robust and balanced result in detecting multilingual and code-mixed offensive language.

Language	TF-IDF + Ensemble			IndicBERT + MLP			XLM-R + XGBoost		
	Accuracy	mF1	wF1	Accuracy	mF1	wF1	Accuracy	mF1	wF1
Tulu	**0.78**	**0.73**	**0.77**	0.64	0.54	0.63	0.76	0.66	0.75
Malayalam	0.91	0.33	0.89	0.89	0.38	0.88	**0.96**	**0.76**	**0.95**
Kannada	**0.70**	**0.39**	**0.66**	0.67	0.29	0.61	0.66	0.27	0.59
Tamil	0.77	0.37	0.71	0.74	0.30	0.67	**0.77**	**0.73**	**0.76**

Table 2
Comparison of base model performance on Tamil, Malayalam, Tulu, and Kannada languages

Language	Accuracy	mF1-score	wF1-score
Tulu	0.77	0.71	0.76
Malayalam	**0.92**	0.41	**0.90**
Kannada	0.69	0.31	0.63
Tamil	0.79	**0.50**	0.73

Table 3
TriFusion Ensemble model performance across Tamil, Malayalam, Tulu, and Kannada languages

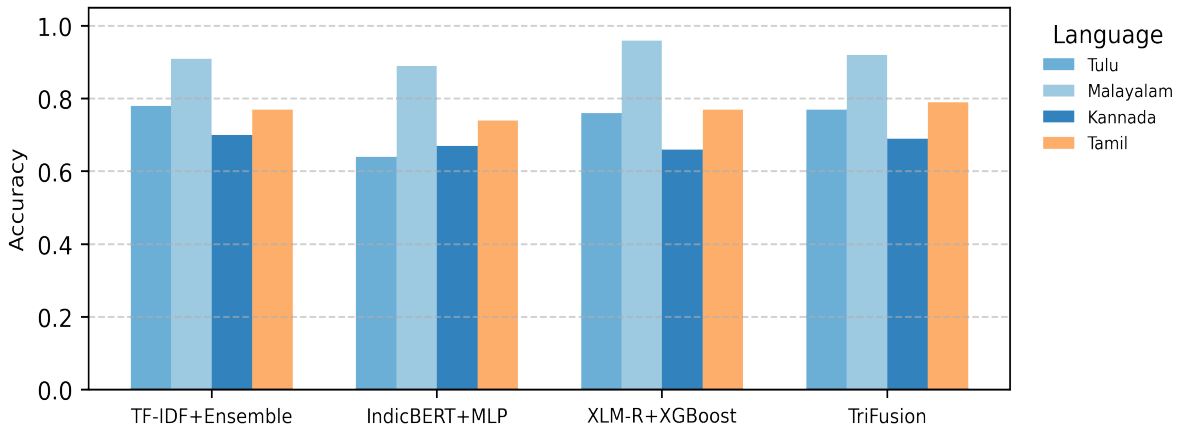

Figure 2: Accuarcy comparison of TFIDF+Ensemble, IndicBERT +MLP, XLM-R+XGBoost, and the TriFusion across the languages.

5.2. Error Analysis

The error analysis was used to analyze the differences in performance between models, with the main emphasis on misclassifications in the minority classes. The comparison of Accuracy, mF1-score, and wF1-Score was found to show that the Accuracy was good across the board, but low mF1-Scores were a manifestation of the way the models treated underrepresented classes like Offensive Targeted and Offensive Non-Targeted. As an example, Tamil and Kannada got a decent Accuracy (0.77 and 0.70) but significantly lower mF1-Score (0.37 and 0.39), indicating the issues of class imbalance.

The XLM-RoBERTa model was found to be able to capture contextual nuances and classify non-offensive categories, as well as generally overfit to the most common non-offensive classification types, but IndicBERT was unable to scale to fine-grained differences because it did not support code-mixed text representation. TriFusion ensemble alleviated such problems by taking advantage of model diversity, providing a higher balance between classes, especially in Malayalam (mF1-score 0.41). The remaining errors were due to the ambiguous nature of cases on sarcasm, transliterated slang, and mixed sentiment polarity where context on interpretation was very important. On balance, the findings indicate that

although hybrid models such as TriFusion can raise the level of robustness, more necessary steps must be implemented to reduce bias via data augmentation strategies, context-enriched embeddings, and contrastive learning to improve the detection of minority classes.

6. Data Availability

The scripts of the proposed methodology, including the preprocessing scripts, feature extraction, and model training modules, are publicly available on Github Repository.

7. Conclusion and Future Work

This paper introduced TriFusion, a multi-model ensemble that used offensive language detection in code-mixed Dravidian languages, namely, Kannada, Tulu, Tamil, and Malayalam. The system showed that in low-resource and linguistically diverse conditions, the combination of lexical and contextual features can help to boost the robustness of the traditional machine learning classifiers with the aid of transformer-based deep representations. The experiments also showed that the contextual cues are well captured by the transformer architectures like XLM-RoBERTa, and that the imbalance limitation typical of underrepresented languages is addressed by the hybrid fusion. TriFusion performed well on all languages, especially on Malayalam, and this proved that ensemble fusion is less biased compared to single models.

To overcome the drawbacks associated with minority class detection as noted in the present study, in the next work, suggested to apply the method of data augmentation and context-enriched embedding. Contrastive learning and cross-lingual alignment may also be used together to increase the resilience of differentiating subtle offensive material in underrepresented groups. These guidelines will assist in moving the field of offensive language detection on low-resource, code-mixed Dravidian languages to a more fair and generalized state.

Declaration on Generative AI

In the course of preparing this manuscript, the author(s) employed the generative AI tool ChatGPT. Its use was limited to performing checks for grammar and spelling. Following this, the author(s) conducted a thorough review and revision of the text and assume full responsibility for the final published content.

References

[1] S. Sai, Y. Sharma, Towards offensive language identification for dravidian languages, in: Proceedings of the first workshop on speech and language technologies for Dravidian languages, 2021, pp. 18–27.

[2] N. Sripriya, B. R. Chakravarthi, T. Durairaj, B. Bharathi, S. C. Navaneethakrishnan, P. K. Kumaresan, A. M D, P. R. Hegde, D. Vikram, Overview of the shared task on offensive language identification in dravidian code-mixed languages, in: Forum of Information Retrieval and Evaluation FIRE-2025, 2025.

[3] A. M. D, D. Vikram, B. R. Chakravarthi, P. R. Hegde, Overcoming low-resource barriers in tulu: Neural models and corpus creation for offensive language identification, 2025. URL: https://arxiv.org/abs/2508.11166. arXiv:arXiv:2508.11166.

[4] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text, Language Resources and Evaluation 56 (2022) 765–806.

[5] N. Sripriya, B. R. Chakravarthi, T. Durairaj, B. Bharathi, C. N. Subalalitha, P. K. Kumaresan, M. D. Anusha, P. R. Hegde, D. Vikram, Overview of the shared task on offensive language identification

in dravidian code-mixed languages, in: Forum of Information Retrieval and Evaluation FIRE-2025, 2025.

[6] R. Adyanthaya, R. Shetty, Yenlp_cs@ dravidianlangtech 2025: Sentiment analysis on code-mixed tamil-tulu data using machine learning and deep learning models, in: Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, 2025, pp. 288–292.

[7] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the international AAAI conference on web and social media, volume 11, 2017, pp. 512–515.

[8] R. Kumar, G. Bhanodai, R. Pamula, M. R. Chennuru, Trac-1 shared task on aggression identification: Iit (ism)@ coling'18, in: Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018), 2018, pp. 58–65.

[9] Z. Talat, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on twitter, in: Proceedings of the NAACL student research workshop, 2016, pp. 88–93.

[10] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german, in: Proceedings of the 12th annual meeting of the forum for information retrieval evaluation, 2020, pp. 29–32.

[11] B. R. Chakravarthi, R. Priyadharshini, N. Jose, T. Mandl, P. K. Kumaresan, R. Ponnusamy, J. P. McCrae, E. Sherly, et al., Findings of the shared task on offensive language identification in tamil, malayalam, and kannada, in: Proceedings of the first workshop on speech and language technologies for Dravidian languages, 2021, pp. 133–145.

[12] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, arXiv preprint arXiv:1911.02116 (2019).

[13] D. Kakwani, A. Kunchukuttan, S. Golla, G. NC, A. Bhattacharyya, M. M. Khapra, 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, 2020, pp. 4948–4961.

[14] J. Devlin, M.-W. Chang, K. Lee, 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.

[15] B. R. Chakravarthi, M. B. Jagadeeshan, V. Palanikumar, R. Priyadharshini, Offensive language identification in dravidian languages using mpnet and cnn, International Journal of Information Management Data Insights 3 (2023) 100151. URL: https://www.sciencedirect.com/science/article/pii/S2667096822000945. doi:https://doi.org/10.1016/j.jjimei.2022.100151.

[16] J. Preetham, J. Anitha, Offensive language detection in social media using ensemble techniques, in: 2023 International Conference on Circuit Power and Computing Technologies (ICCPCT), IEEE, 2023, pp. 805–808.

[17] R. Razzak, S. T. O. Taki, M. R. Mim, M. S. H. Patwary, M. I. Pavel, Vulgar comments classification: Comparison between cnn, xgboost and svm (2023).

[18] A. Garain, A. Mandal, S. K. Naskar, Junlp@ dravidianlangtech-eacl2021: Offensive language identification in dravidian langauges, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021, pp. 319–322.

[19] M. Xu, S. Liu, Rb_bg_mha: A roberta-based model with bi-gru and multi-head attention for chinese offensive language detection in social media, Applied Sciences 13 (2023) 11000.