Exploring Classical Machine Learning and Deep Learning Approaches for Offensive Language Identification in Dravidian Code-Mixed Text

Rachana Nagaraju*,†, Hosahalli Lakshmaiah Shashirekha

Department of Computer Science, Mangalore University, Mangalore, Karnataka, India

Abstract

Offensive Language Identification (OLI) has become an important task, particularly in the context of social media, where harmful content such as hate speech, cyberbullying, and toxic discourse can spread rapidly and widely. The challenge becomes even more complex in low-resource, code-mixed settings across Dravidian languages such as Tamil, Malayalam, Kannada, and Tulu. To address these challenges, *Offensive Language Identification in Dravidian Code-Mixed Languages* shared task at FIRE 2025 released gold-standard annotated datasets of comments collected from online platforms in these four languages, with the goal of developing automated systems that can classify user comments as either offensive or non-offensive. We - team **MUCS**, participated in the shared task on OLI by developing two complementary pipelines: i) ***Off_ML*** - an ensemble of traditional Machine Learning (ML) estimators (Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF), and Naïve Bayes (NB) classifiers) with hard and weighted voting, trained with Term Frequency–Inverse Document Frequency (TF-IDF) of word ngrams in the range (1–3) as features, and ii) ***Off_DL*** - a hybrid Deep Learning (DL) architecture that combines Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM). These pipelines highlight the comparative strengths of feature-based ML models and representation-driven DL architectures, while emphasizing the benefits of ensembling for multilingual offensive language detection. Experimental results demonstrated that the classical ML pipeline consistently outperformed the DL architecture in all four languages. For Tamil, Off_ML model with hard voting obtained a macro-averaged F1-score of 0.452 (Rank 3) compared to that of DL's 0.350 (Rank 8). For Malayalam, Off_ML model with weighted voting reached 0.712 macro-averaged F1-score (Rank 4), while that of DL model achieved only 0.350 (Rank 8). For Kannada, Off_ML model with hard voting obtained a macro-averaged F1-score of 0.421 (Rank 5), whereas DL model managed to get 0.344 macro-averaged F1-score (Rank 8). Finally, for Tulu, Off_ML model with weighted voting delivered its strongest performance with 0.790 macro-averaged F1-score (Rank 2), while DL model recorded that of 0.610 (Rank 7). These results highlight that, despite the increasing prominence of neural methods, traditional ML models with careful feature engineering remain highly competitive and, in several cases, superior for OLI in low-resource, code-mixed Dravidian languages such as Tamil, Malayalam, Kannada, and Tulu.

Keywords

Offensive Language Identification, Code-Mixed Text, Dravidian Languages, Social Media

1. Introduction

OLI is an important application in Natural Language Processing (NLP), particularly in the context of social media platforms where these platforms serve as primary spaces for public discourse. Online platforms often suffer from toxic interactions, hate speech, cyber bullying, and abusive content, making the detection of offensive language a necessary step toward ensuring safe and inclusive communication. Unlike monolingual formal text, user-generated social media content tends to be short, noisy, and highly informal, presenting significant challenges for automated text processing systems [1, 2]. The complexity of this task increases in *code-mixed* environments where the practice of blending words and/or phrases from multiple languages in a single utterance is widespread. Code-mixed data often employ a combination of Roman and native scripts, phonetic transliteration, and inconsistent grammar, all of which complicate the application of standard monolingual models [3].

Forum for Information Retrieval Evaluation, December 17-20, 2025, Varanasi, India

*Corresponding author.

rachananagaraju20@gmail.com (R. Nagaraju); hlsrekha@mangaloreuniversity.ac.in (H. L. Shashirekha)

0000-0002-9421-8566 (H. L. Shashirekha)

Table 1
Sample Code-mixed text in Dravidian Languages

Text	Labels	Language
Ee trailer kandapo KGF ormavanavar adikk...	Not_offensive	Malayalam
pulimurukan trailer ano kanunath	Not_offensive	Malayalam
Varki annu perillatha edavakakar undo	Not_offensive	Malayalam
My god Trailer kandittu pediyayi	Not_offensive	Malayalam
200 kodi clubil orapullavar like pls	Not_offensive	Malayalam
Mass ka baap ka BAAPP	Not_Language	Malayalam
Great work bro....n Keep it up	Not_Language	Kannada
ASN ಖಂಡಿತವಾಗ್ಲೂ ಚರಿತ್ರೆ ಶೃಷ್ಟಿಸುವ ಅವತಾರ....	Not_offensive	Kannada
Masthmaga na duplicate madbitaru Amar sir	Offensive_Targeted_ Insult_Individual	Kannada
Suuuuuuuuuuprb bro waiting next video bro	Not_Language	Kannada
Super star Darshan fans ikada	Not_Language	Kannada
Views tumba kadime ide.	Offensive_Untargeted	Kannada
எத்தனை தடவை பார்த்தாலும் போர் அடிக்கவில்லை டீசர்	Not_offensive	Tamil
It s Amazing Thala pola varuma	Not_offensive	Tamil
No doubt utterflop like another Puli	Offensive_Targeted_ Insult_Individual	Tamil
It's been like Ellam Avan seyal	Not_offensive	Tamil
Oru ambalayacham vanthare Mohan j	Not_offensive	Tamil
Remake of Pink awesome movie	Not_offensive	Tamil
Rakshitha baaari shoku apundu eeerna videos	Not_offensive	Tulu
very boar episode comedy ijji net	Not_offensive	Tulu
All English patherunu Dada mare	Offensive_Untargeted	Tulu
Tulu nank kannada p**k 😂 Bevarashi😂	Offensive_Untargeted	Tulu
dada umbe ajji g dayag satawoon	Not_offensive	Tulu
Next g kunne maya ayijinda yaVu	Offensive_Targeted_ Insult_Other	Tulu

Dravidian languages such as Tamil, Malayalam, Kannada, and Tulu are frequently mixed with English in online discourse, producing a unique set of challenges. Earlier studies of OLI in Dravidian languages [4, 5] have highlighted the scarcity of high-quality datasets and the limited effectiveness of transfer learning approaches in these low-resource settings. Table 1 presents few samples of user-generated content in Dravidian languages, highlighting the informal, noisy, and code-mixed nature of the text that poses challenges for automated OLI systems.

To address the challenges of OLI in code-mixed Dravidian languages, *Offensive Language Identification in Dravidian Code-Mixed Languages* [6] shared task at FIRE 2025 invited researchers to build automated systems that can distinguish user comments mainly as offensive or non-offensive in four Dravidian languages - Tamil, Malayalam, Kannada, and Tulu [6]. Similar shared tasks such as HASOC [7] and TRAC [8], in the past, demonstrated the value of creating benchmark datasets for abusive language detection. FIRE 2025 task continues this tradition by focusing on OLI in Dravidian code-mixed data, thus providing researchers with a test bed for evaluating learning systems for a low-resource code-mixed content.

We - team **MUCS**, participated in this shared task with two distinct pipelines, aiming to systematically compare traditional feature-based ML models with modern DL architectures. The first pipeline - ***Off_ML***, relies on TF–IDF features to train the ensemble of ML estimators (LR, SVM, RF, and NB) with hard and weighted voting. The second pipeline - ***Off_DL***, explores a hybrid CNN–LSTM model to capture semantic and contextual information more effectively. Our code is available on GitHub [1]. Our experimental results reveal a clear performance gap between the two approaches. For Tamil, the ML pipeline with hard voting achieved a macro-averaged F1-score of 0.452 (Rank 3), while the DL pipeline settled with that of 0.350 (Rank 8). For Malayalam, ML with weighted voting delivered macro-averaged F1-score 0.712

[1] https://github.com/rachanabn20/Offensive-Language-Identification-in-Dravidian-Code-Mixed-Languages---DravidianCodeMix-FIRE-2025

(Rank 4), compared to that of DL's 0.350 (Rank 8). For Kannada, ML model with hard voting secured 0.421 macro-averaged F1-score (Rank 5), while DL reached that of 0.344 (Rank 8). Finally, for Tulu, ML with weighted voting recorded its strongest result with 0.790 macro-averaged F1-score (Rank 2), whereas DL achieved its best overall performance with 0.610 macro-averaged F1-score (Rank 7). These findings emphasize that, despite the increasing popularity of DL approaches, traditional ML methods with careful feature engineering remain highly effective in low-resource and code-mixed language scenarios. Moreover, while DL models showed some promise in Tulu, their overall performance was inconsistent, highlighting the need for further research in adapting neural architectures for complex multilingual and code-mixed environments.

The subsequent sections of this paper details the related works (Section 2), methodology (Section 3), experiments, results, and implications of our approach (Section 4) followed by conclusion and future works (Section 5).

2. Related Work

The automatic detection of offensive and abusive language has been widely studied in recent years to advance research and develop robust systems for automatic OLI, particularly in Indian languages and code-mixed social media texts. The TRAC shared task [8] benchmarked aggression identification in social media. The HASOC track at FIRE 2019 [7] focused on hate speech and offensive content detection in Indo-European languages, while HASOC at FIRE 2020 [9] extended the focus to Hate Speech and OLI in Tamil, Malayalam, Hindi, English, and German. Furthermore, HASOC-Dravidian-CodeMix at FIRE 2021 [10] targeted code-mixed Tamil and Malayalam, whereas DravidianLangTech at EACL 2021 [11] addressed OLI in code-mixed Dravidian languages (Tamil-English, Malayalam-English, and Kannada-English). These tasks have significantly contributed to the creation of annotated datasets, evaluation metrics, and baseline models for offensive language detection in Indian and Dravidian languages, fostering inclusive and safer digital communication environments.

Early surveys by Schmidt and Wiegand [1] and Fortuna and Nunes [2], provided comprehensive overviews of hate speech detection techniques, ranging from lexicon-based approaches to more advanced ML models. These studies highlighted the limitations of keyword-based models and emphasized the importance of contextual features for reliable detection. In multilingual and code-mixed scenarios, the task becomes even more challenging. Jose and Choudhury [3], surveyed sentiment analysis and opinion mining in code-mixed text, noting difficulties such as transliteration, inconsistent grammar, and lack of standardized resources. Code-mixed datasets have also been explored in language pairs such as Hindi–English and Bengali–English, further illustrating the challenges of feature extraction and classification in noisy user-generated text [12].

Saumya et al. [13] compared traditional and neural approaches for OLI, reinforcing the importance of character- and word-level representations in noisy code-mixed data. Chakravarthi et al. [5] reported the first benchmark results for OLI in Tamil, Malayalam, and Kannada through a *Offensive Language Identification in Dravidian Languages-EACL 2021* shared task. The baseline systems included classical ML models such as LR, SVM, and RF with TF–IDF features, as well as neural models including CNN, BiLSTM, and multilingual transformers such as mBERT and XLM-R. Interestingly, classical ML models with TF–IDF often outperformed DL approaches in low-resource conditions, while transformer-based models achieved competitive performance but required careful tuning and were highly sensitive to noise in code-mixed data. The shared task's main contributions included the establishment of standardized datasets and evaluation protocols, demonstrating the continued strength of classical ML models for small and imbalanced datasets, and the potential of multilingual transformers. However, challenges such as limited data availability, transliteration inconsistencies, and poor cross-lingual transferability restricted model generalization.

Prajnashree et al. [14] proposed traditional ML models (SVM, RF, Passive Aggressive Classifier) and Siamese LSTM for OLI in low-resource Indian languages on HASOC 2023 datasets. The results illustrated that SVM achieved strong results for Sinhala (macro F1 = 0.78, Rank 11) and Siamese LSTM for

Gujarati (macro F1 = 0.72, Rank 12). Although effective in low-resource conditions, these models lagged behind transformer-based approaches due to limited contextual understanding. Fazlourrahma et al. [15] introduced COOLI, a system for code-mixed OLI in Tamil-English (Ta–En), Malayalam-English (Ma–En), and Kannada-English (Kn–En). They proposed two models: (i) COOLI-Ensemble (MLP, XGBoost, and LR in a voting setup) and (ii) COOLI-Keras (dense neural network). COOLI-Ensemble achieved the best performance, ranking 1st for Ma–En (F1 = 0.97), 4th for Ta–En (F1 = 0.75), and 6th for Kn–En (F1 = 0.69). Despite these strong results, dataset imbalance and inconsistent Romanization limited cross-lingual generalization.

Overall, prior work underscores the dual challenge of OLI and code-mixing, motivating the need for specialized approaches that balance classical feature-based ML approaches with DL models tailored for low-resource, code-mixed settings.

3. Methodology

Our approach to OLI in Dravidian code-mixed texts consists of classical ML and DL models. The methodology includes text pre-processing, feature extraction, model training, and evaluation. We developed two independent pipelines, namely ***Off_ML*** and ***Off_DL***, to systematically analyze performance trade-offs between traditional ML models and neural methods for OLI in four Dravidian languages. The steps involved in the methodology are given below:

3.1. Text Pre-processing

The dataset comprised user-generated code-mixed texts in four Dravidian languages: Kannada, Malayalam, Tulu, and Tamil. These texts contained multiple classes, including few types of offensive categories and a non-offensive category. Due to the noisy nature of user-generated content, the following pre-processing steps are applied to clean and prepare the text for further processing:

- **Normalization:** Repeated characters are reduced (e.g., *soooo* $\rightarrow$ *soo*) to handle exaggerated expressions.
- **Noise Removal:** URLs, user mentions, emojis, hashtags, numbers, and non-alphanumeric symbols are removed.
- **Stopwords:** Minimal stopword removal was performed to avoid loosing functional words that may contribute to offensive tone.
- **Transliteration Variants:** Transliterated forms (e.g., phonetic spellings in Roman script such as *maga* (Roman script) vs. *maga* (native Kannada script) are retained as it is without exlicit normalization.

3.2. Feature Engineering

Feature representations differed significantly across the two pipelines:

- **Features for *Off_ML* Models:** Word-level n-grams in the range (1–3) (unigrams, bigrams, and trigrams) are extracted to capture explicit lexical signals pertaining to offensive expressions and handle spelling variations. These features are then vectorized using the `TfidfVectorizer`[2] with the vocabulary size restricted to 15,000 features. Even though the feature set is high-dimensional and sparse, it is highly effective for linear classifiers. However, this work does not include the analysis of TF–IDF features contributing to classification performance, nor does it provide inspection of the most discriminative n-grams. Further, no explicit linguistic features are incorporated to address the challenges arising from code-mixed or morphologically rich text, which are common in multilingual social media data.
- **Features for *Off_DL* Models:** Each word token in the dataset is mapped to dense vector representation using an embedding layer with randomly initialized weights.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

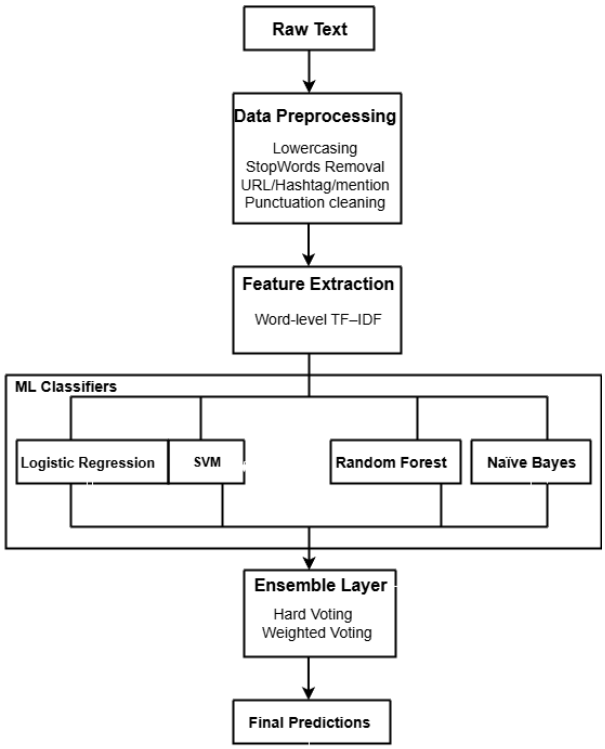


Figure 1: Proposed Machine Learning Framework

3.3. Model Training

The two pipelines are trained with different features and frameworks of these pipelines are shown in Figures 1 and 2.

- ***Off_ML* Model:** The following estimators are used in the ensemble model to enhance the performance of the classifier:
  - **Logistic Regression**[3]: is a linear classifier effective in high-dimensional sparse spaces.
  - **Support Vector Machine**[4]: is a margin-based classifier that maximizes class separation.
  - **Random Forest**[5]: is an ensemble of decision trees that improves robustness to feature noise.
  - **Naïve Bayes**[6]: is a probabilistic classifier well-suited for text data.

  These estimators are ensembled with *hard voting* (predictions from individual classifiers are combined based on majority voting) for Kannada and Tamil, and *weighted voting* (classifiers with higher validation performance are assigned larger weights) for Tulu and Malayalam.
- ***Off_DL* Model:** The Hybrid CNN–LSTM model combines CNN layers for local feature extraction with LSTM layers for sequential modeling, balancing short-span and long-range contexts:
  - **Convolutional Neural Network**[7]: learns phrase-level features directly from embeddings.
  - **Long Short-Term Memory**[8]: captures discourse-level and sequential patterns across tokens.

[3]https://en.wikipedia.org/wiki/Logistic_regression
[4]https://en.wikipedia.org/wiki/Support_vector_machine
[5]https://en.wikipedia.org/wiki/Random_forest
[6]https://en.wikipedia.org/wiki/Naive_Bayes_classifier
[7]https://en.wikipedia.org/wiki/Convolutional_neural_network
[8]https://en.wikipedia.org/wiki/Long_short-term_memory

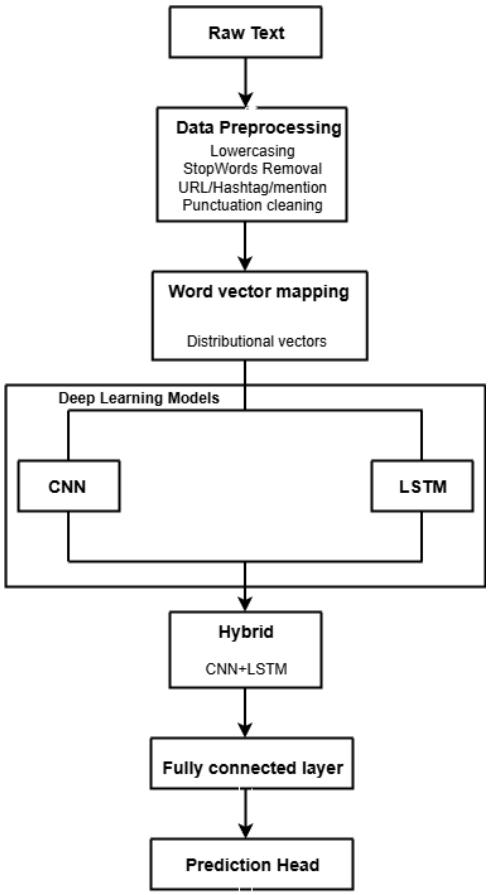


Figure 2: Proposed Deep Learning Framework

Table 2
Hyperparameters used in *Off_ML* and *Off_DL* Models

Pipeline	Model	Hyperparameters
Off_ML	LR	C = 1.0, Solver = liblinear
	SVM	Kernel = linear, C = 1.0
	RF	Trees = 200, Max depth = 20
	NB	Multinomial, Smoothing = 1.0
Off_DL	CNN	Filters = 128, Kernel size = 3, Dropout = 0.5
	LSTM	Hidden units = 128, Dropout = 0.5
	Hybrid CNN–LSTM	CNN filters = 64, LSTM units = 128, Dropout = 0.5
Optimization (DL)		Adam, Learning rate = 0.001, Batch size = 32, Epochs = 10

The final representations are fed to a fully connected layer to output probabilities across the offensive language categories.

Table 2 summarizes the hyperparameters used for both pipelines. While classical ML models are tuned via grid search[9], DL models used validation-based tuning.

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

Table 3
Language-wise Statistics of the Datasets

Language	Train	Val	Test	Label	Count
Kannada	6,217	777	778	Not_Offensive	3,544
				Not-Kannada	1,522
				Offensive_Targeted_Insult_Individual	487
				Offensive_Targeted_Insult_Group	329
				Offensive_Untargeted	212
				Offensive_Targeted_Insult_Other	123
Malayalam	16,010	1,999	2,001	Not_Offensive	14,153
				Not-Malayalam	1,287
				Offensive_Targeted_Insult_Individual	239
				Offensive_Untargeted	191
				Offensive_Targeted_Insult_Group	140
Tulu	2,692	577	576	Not Offensive	1,261
				Not Tulu	726
				Offensive Untargeted	462
				Offensive Targeted	243
Tamil	35,139	4,388	4,392	Not_Offensive	25,425
				Offensive_Untargeted	2,906
				Offensive_Targeted_Insult_Group	2,557
				Offensive_Targeted_Insult_Individual	2,343
				Not-Tamil	1,454
				Offensive_Targeted_Insult_Other	454

4. Experiments and Results

The proposed pipelines are built and evaluated using the Dravidian languages: Kannada, Malayalam, and Tamil datasets provided by the organizers of the shared task [5, 4]. Recent efforts have introduced OLI dataset in another Dravidian language Tulu [16], highlighting the growing importance of OLI in low-resource contexts. Each language dataset has Train, Validation and blind Test sets, and the distribution of these sets is shown in Table 3. Even though the datasets are highly imbalanced, we did not make any efforts to balance the datasets.

4.1. Results

The performance of the models is reported in terms of macro-averaged Precision (mPrecision), macro averaged Recall (mRecall), and macro averaged F1-score (mF1-score) across all the classes. This allows us to capture performance across all the classes in a better way rather than being dominated by the majority class. The participating teams are ranked based on the performances of the models in terms of mF1-score. Tables 4 and 5 present the results of our traditional ML ensemble approaches on Validation and Test sets respectively, while Tables 6 and 7 report the performance of the DL pipelines on Validation and Test sets respectively.

From the results, it is clear that the *Off_ML* models consistently outperformed the *Off_DL* models across all languages on the Test sets. For Kannada, *Off_ML* model achieved mF1-score of 0.42 with 5^{th} rank, compared to *Off_DL* model's mF1-score of 0.34 with 8^{th} rank. For Malayalam, *Off_ML* model obtained mF1-score of 0.71 (Rank 4), while *Off_DL* model reached only 0.35 mF1-score (Rank 8). Tulu showed the strongest *Off_ML* model performance with a mF1-score of 0.79 (Rank 2), substantially better than *Off_DL* model's 0.61 mF1-score (Rank 8). Tamil also followed this trend, where *Off_ML* model scored a mF1-score of 0.45 (Rank 3) but *Off_DL* model lagged at 0.35 mF1-score (Rank 8). Though *Off_DL* models captured contextual patterns, TF–IDF based *Off_ML* models proved significantly more competitive in the shared task.

Figures 3a, 3b, 3c, and 3d illustrate the leaderboard rankings based on mF1-score for Tulu, Tamil, Malayalam, and Kannada respectively, highlighting the relative performance of our systems against

other participating teams.

Table 4
Performances of *Off_ML* Models on Validation Set

Language	mPrecision	mRecall	mF1-score
Kannada	0.4537	0.4401	0.4415
Malayalam	0.5907	0.7115	0.6411
Tulu	0.7800	0.7590	0.7676
Tamil	0.4226	0.4757	0.4430

Table 5
Performances of *Off_ML* Models on Test Set

Language	mPrecision	mRecall	mF1-score	Rank
Kannada	0.42	0.42	0.42	5th
Malayalam	0.65	0.75	0.71	4th
Tulu	0.81	0.77	0.79	2nd
Tamil	0.44	0.48	0.45	3rd

Table 6
Performances of *Off_DL* Models on Validation Set

Language	mPrecision	mRecall	mF1-score
Kannada	0.31	0.37	0.33
Malayalam	0.35	0.32	0.33
Tulu	0.63	0.62	0.59
Tamil	0.35	0.33	0.33

Table 7
Performances of *Off_DL* Models on Test Set

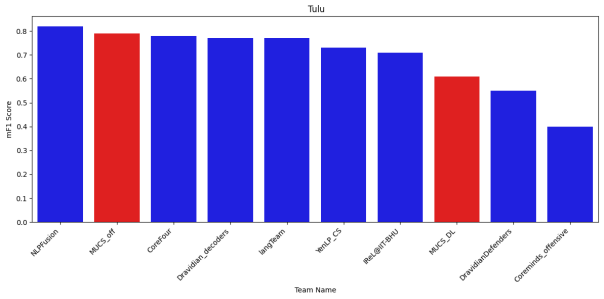
Language	mPrecision	mRecall	mF1-score	Rank
Kannada	0.31	0.37	0.34	8th
Malayalam	0.36	0.35	0.35	8th
Tulu	0.70	0.63	0.61	8th
Tamil	0.41	0.33	0.35	8th

4.2. Confusion Matrices

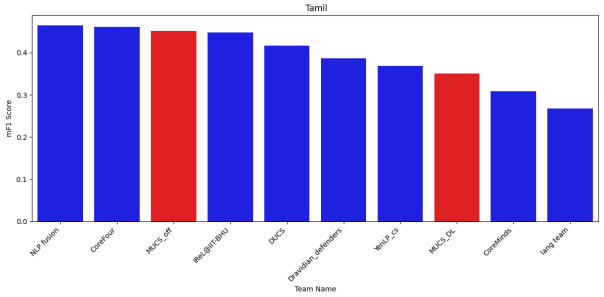
The confusion matrices provide further insight into classification performance by highlighting the misclassification. Figures 4a, 4b, 4c, and 4d illustrate the performances of *Off_ML* models across Kannada, Malayalam, Tamil, and Tulu datasets, respectively. Confusion matrices for Kannada, Tulu, Malayalam, and Tamil, obtained using *Off_DL* models are provided in Figures 5a, 5b , 5c and 5d, respectively, to show the contrast with ML pipelines. The low mF1-scores of *Off_DL* models for Kannada and Tamil, indicate that minority offensive categories are under-predicted. This mirrors the behavior observed in *Off_ML* models, though *Off_DL* pipelines are better at balancing recall across the major and mid-frequency classes.

4.3. Error Analysis

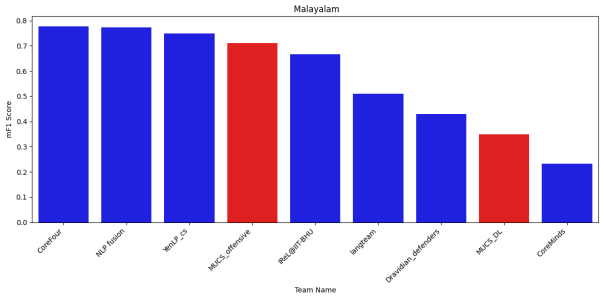
While the overall results demonstrate that both *Off_ML* and *Off_DL* models are capable of handling OLI in multiple langauges, a closer look at the performance across languages reveals important insights:



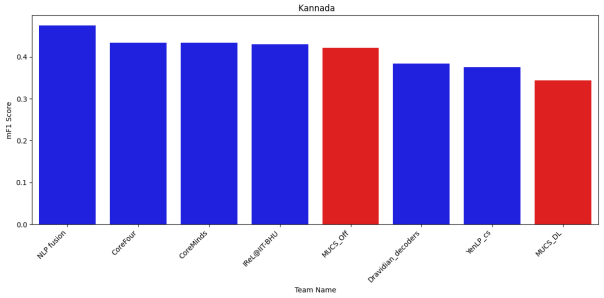
(a) Leaderboard Rankings for Tulu



(b) Leaderboard Rankings for Tamil



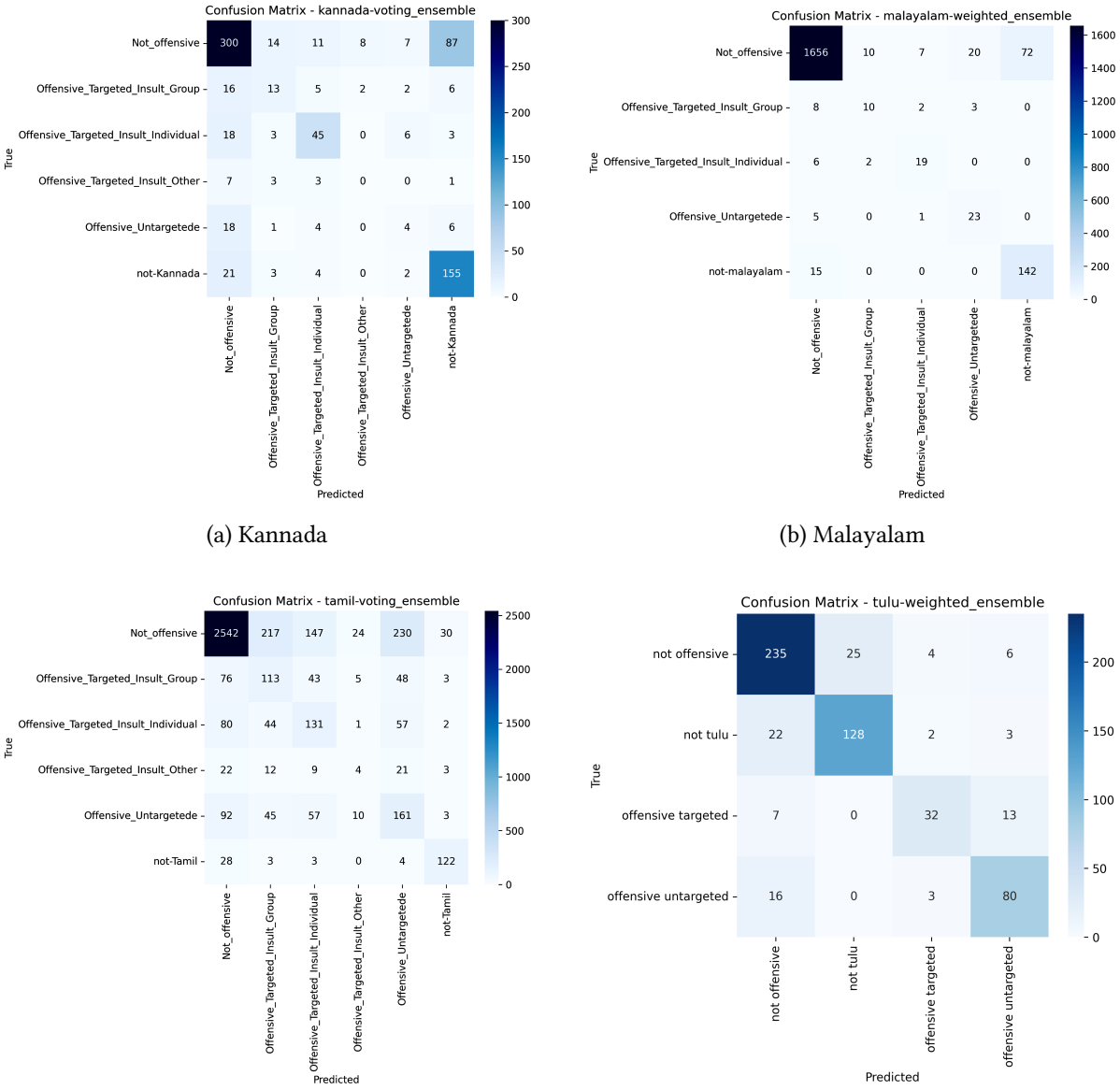
(c) Leaderboard Rankings for Malayalam



(d) Leaderboard Rankings for Kannada

Figure 3: Leaderboard Rankings across Languages

- Class imbalance has a significant impact in the performances of the classifiers. Minority categories such as *Offensive_Targeted_Insult_Other* or *Offensive_Untargeted* consistently showed very low recall, often close to zero in Kannada and Tamil. This suggests that both ML and DL systems leaned heavily toward predicting the more frequent *Not_Offensive* class, leading to missed detections of subtler offensive expressions.
- Linguistically, both *Offensive_Targeted_Group* and *Offensive_Targeted_Individual* categories rely on similar cues — such as insults or derogatory expressions — making it difficult for the models to distinguish whether the offense was aimed at a group or a single person.



(a) Kannada

(b) Malayalam

(c) Tamil

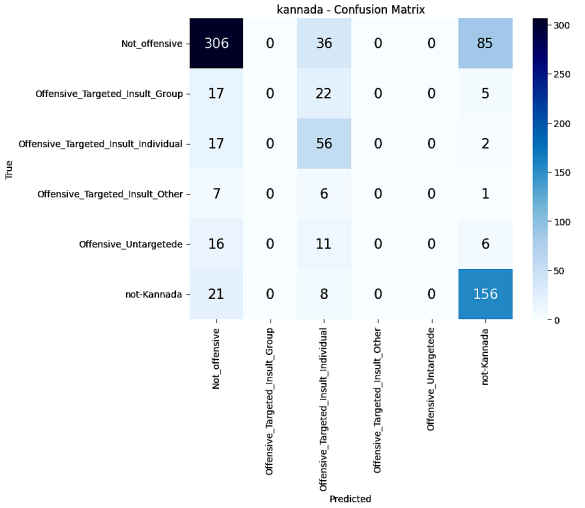
(d) Tulu

Figure 4: Confusion Matrices for Different Languages using *Off_ML* Model

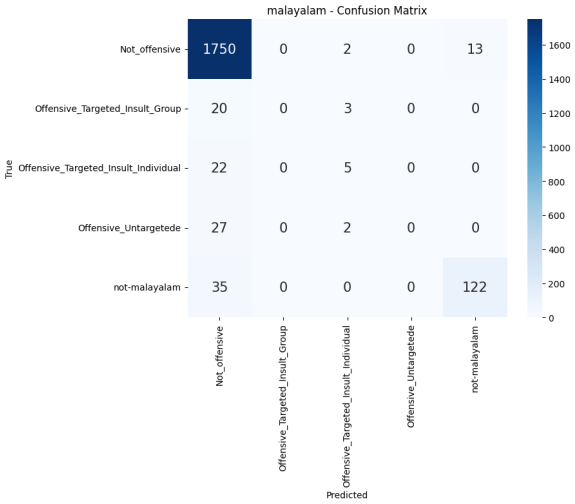
- Language-specific differences are striking. Malayalam and Tulu achieved higher weighted F1-scores, indicating relatively robust detection. These datasets appear cleaner, with fewer ambiguous cases, which may explain the performance boost. In contrast, Kannada and Tamil suffered from noisier data and heavier code-mixing, which made offensive intent harder to capture.
- Code-mixing itself remains one of the most challenging factors. Many social media posts combine English with native Dravidian scripts, producing hybrid structures that traditional ML and DL models struggled with. For example, a sentence might switch languages mid-phrase, obscuring both semantic and syntactic cues that are crucial for offensive language detection.

These observations highlight that while the models perform reasonably well for some languages, particularly Malayalam and Tulu, further work is needed to improve robustness in noisier and more code-mixed contexts. Addressing class imbalance and exploring advanced contextual embeddings or data augmentation techniques could help bridge this gap in the future.

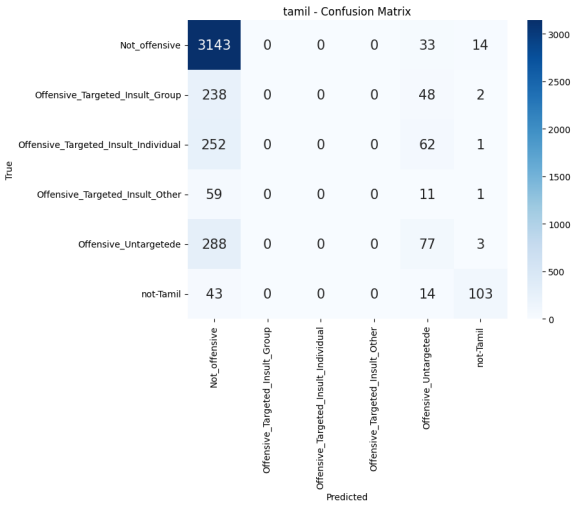
Table 8 presents representative misclassified samples across languages. Several factors contribute to the mis-classification of Test samples as given below:



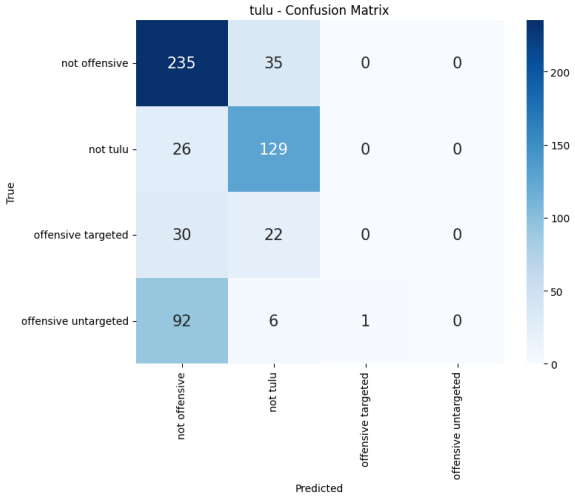
(a) Kannada



(b) Malayalam



(c) Tamil



(d) Tulu

Figure 5: Confusion Matrices for Different Languages using *Off_DL* Model

- **Code-switching and Mixed Scripts:** The classifier struggles when users mix English with regional languages or emojis. For example: *"Enjoyed too much. Superb"* was labeled as `Not_Language` but predicted as `Not_offensive`.
- **Sarcasm and Context Dependence:** Subtle insults or sarcastic remarks are difficult for the model to interpret. Example: *"so poor dialogue delivery From Mohanal..."* was a targeted insult but predicted as untargeted.
- **Ambiguity in Offense Type:** The model may detect offensive tone but fail to decide whether it is targeted or untargeted. Example: *"Rashmika ide iro avtava"* (targeted insult at an individual) was predicted as untargeted.
- **Dialect and Spelling Variations:** Regional slang and non-standard spellings confuse the classifier. Example: *"Bvc yrt comment manpva ha yavu"* (untargeted insult) was predicted as a targeted insult.
- **Named Entities and Cultural References:** The system fails to recognize organizations or cultural groups as insult targets. Example: *"RIP kannada film chamber. No proper plannings"* was an insult towards a group but predicted as non-language.
- **Polite vs. Non-language Confusion:** Positive or polite expressions are sometimes misclassified

Table 8
Misclassified Samples along with their Predicted and True Class Labels

Language	Text	True Label	Predicted Label
Tulu	Ganesh Anna na pukuli	Not_offensive	Offensive_Untargeted
Tulu	Enjoyed too much. Superb🔥 🔲🔲🔲	Not_Language	Not_offensive
Tulu	Pukuli Maya ..... Wa marl mare	Offensive_Targeted_ Insult_Individual	Offensive_Untargeted
Tulu	Bvc yrt comment manpva ha yavu	Offensive Untargeted	Offensive_Targeted_ Insult_Individual
Kannada	Sir dasara habbada shubashayagalu	Not_offensive	Not_Language
Kannada	Tik tak best app	Not_Language	Offensive_Targeted_ Insult_Group
Kannada	ಮೊದಲು ನಿನ್ನ ಗುಲಾಮು ಜಾಮ್ ಬಿದ್ದಿದ್ದ ನೋಡು	Offensive_Targeted_ Insult_Individual	Not_offensive
Kannada	RIP kannada film chamber. No proper plannings	Offensive_Targeted_ Insult _Group	Not_Language
Kannada	Shaata ivrnella borderg kalsbeku gotagutte	Offensive_Targeted_ Insult_Other	Offensive_Targeted_ Insult_Individual
Kannada	Rashimika ilde iro avtara	Offensive_Untargeted	Not_offensive
Malayalam	Fefka ee padam release cheyyan samadhicho?	Not_offensive	Offensive_Targeted_ Insult_Individual
Malayalam	Plese upload sunny Leone scene.	Not_Language	Not_offensive
Malayalam	Sangi lalappan.....ninte padam 8 nilayil pottumm!!!!	Offensive_Targeted_ Insult_Individual	Offensive_Targeted_ Insult_Group
Malayalam	Padam pottum... Charithram akanamenkil marakkar varanam	Offensive_Targeted_ Insult_Group	Not_offensive
Tamil	idhu 96 yara emathuravera lvl uh	Not_offensive	Offensive_Targeted_ Insult_Group
Tamil	so poor dialogue delivery From Mohanlal....	Offensive_Targeted_ Insult_Individual	Offensive_Untargeted
Tamil	Ana trailer pathona exaitment thanga mudyala	Offensive_Targeted_ Insult_Group	Offensive_Untargeted
Tamil	Dei bhairava copied photo echa pasangala	Offensive_Targeted_ Insult_Other	Offensive_Targeted_ Insult_Group
Tamil	Movie yachum nailla eruka nu papom	Offensive_Untargeted	Offensive_Targeted_ Insult_Other

as `Not_Language`. Example: *"Sir dasara habbada shubashayagalu"* was actually non-offensive but predicted as `Not_Language`.

- **Entity Type Confusion (Individual vs. Group vs. Other):** The classifier struggles to distinguish whether an insult is directed at an individual, a group, or an abstract concept. Example: *"Padam pottum... Charithram akanam kett marakkar varanam"* was a group insult but predicted as non-offensive.

The above mis-classification samples illustrate the challenges of noisy code-mixed content which includes inconsistent language mixing, informal spellings, and limited annotated data.

5. Conclusion and Future Work

In this work, we - team **MUCS** explored both ensemble of traditional ML models (*Off_ML*) and DL *Off_DL* pipelines for OLI in four Dravidian languages: Kannada, Malayalam, Tulu, and Tamil. The overall results showed that *Off_ML* models are more effective achieving strong leaderboard rankings — 2nd place for Tulu, 3rd place for Tamil, 4th place for Malayalam, and 5th place for Kannada. In contrast, *Off_DL* models struggled, consistently ranking 8th across all languages despite reasonable validation performance. Our analysis highlighted the key challenges: i) Minority offensive categories such as *Offensive_Targeted_Insult_Other* and *Offensive_Untargeted*, are severely under-predicted, leading to low mF1-scores (0.32–0.35 for Kannada and Tamil using *Off_DL* models), ii) Frequent confusions also occurred between related classes such as *Offensive_Targeted_Group* and *Offensive_Targeted_Individual*, and iii) Malayalam and Tulu benefited from relatively cleaner datasets, while Kannada and Tamil are impacted by noisy, code-mixed inputs that make offensive intent harder to capture. Looking ahead, there are several promising directions. Leveraging multilingual transformers (e.g., XLM-R, mBERT) or instruction-tuned Large Language Models (LLMs) may enhance the ability of the models to capture subtle cues of offensiveness. Text augmentation, re-sampling, and synthetic text generation, could

help alleviate class imbalance and improve recall for minority classes. Code-mixing aware models and transliteration-based pre-processing are also worth exploring to handle hybrid language scenarios more effectively. Finally, extending these experiments to additional Dravidian and low-resource languages will provide broader insights into the generalizability of offensive language detection systems. In summary, while we achieved competitive leaderboard performance through ML ensembles, further refinements are required for DL pipelines. Addressing class imbalance, code-mixing, and fine-grained class distinctions remains crucial for developing robust multilingual offensive language detection systems.

Declaration on Generative AI

Generative Artificial Intelligence (AI) tools are used in the preparation of this work for language refinement, grammar improvement, and formatting suggestions. The research content, analysis, results, and conclusions are developed independently, and AI tools are not used to generate original research findings.

References

[1] A. Schmidt, M. Wiegand, A Survey on Hate Speech Detection using Natural Language Processing, Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media (2017).

[2] P. Fortuna, S. Nunes, A Survey on Automatic Detection of Hate Speech in Text, ACM Computing Surveys (CSUR) 51 (2018) 1–30.

[3] J. Jose, M. Choudhury, A Survey on Sentiment Analysis and Opinion Mining in Code-Mixed Text, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020.

[4] B. R. Chakravarthi, R. Priyadharshini, et al., DravidianCodeMix: Sentiment Analysis and Offensive Language Identification Dataset for Dravidian Languages, in: Proceedings of the 12th Language Resources and Evaluation Conference, 2020.

[5] B. R. Chakravarthi, R. Priyadharshini, et al., Findings of the Shared Task on Offensive Language Identification in Dravidian Languages, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021.

[6] 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.

[7] T. Mandl, S. Modha, et al., Overview of the Hasoc Track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, 2019.

[8] R. Kumar, A. K. Ojha, S. Malmasi, M. Zampieri, Benchmarking Aggression Identification in Social Media, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-1), 2018.

[9] 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, FIRE '20, Association for Computing Machinery, New York, NY, USA, 2021, p. 29–32. URL: https://doi.org/10.1145/3441501.3441517. doi:10.1145/3441501.3441517.

[10] P. K. Kumaresan, Premjith, R. Sakuntharaj, S. Thavareesan, S. Navaneethakrishnan, A. K. Madasamy, B. R. Chakravarthi, J. P. McCrae, Findings of Shared Task on Offensive Language Identification in Tamil and Malayalam, in: Proceedings of the 13th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE '21, Association for Computing Machinery, New York, NY, USA, 2022, p. 16–18. URL: https://doi.org/10.1145/3503162.3503179. doi:10.1145/3503162.3503179.

[11] B. R. Chakravarthi, R. Priyadharshini, N. Jose, A. Kumar M, T. Mandl, P. K. Kumaresan, R. Ponnusamy, H. R L, J. P. McCrae, E. Sherly, Findings of the Shared Task on Offensive Language Identification in Tamil, Malayalam, and Kannada, in: B. R. Chakravarthi, R. Priyadharshini, A. Kumar M, P. Krishnamurthy, E. Sherly (Eds.), Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 133–145. URL: https://aclanthology.org/2021.dravidianlangtech-1.17/.

[12] P. Patwa, G. Aguilar, S. Kar, T. Solorio, SemEval-2020 Task 9: Overview of Sentiment Analysis of Code-Mixed Tweets, in: Proceedings of the 14th International Workshop on Semantic Evaluation, 2020.

[13] S. K. Saumya, D. Khurana, B. R. Chakravarthi, Offensive Language Identification in Dravidian Code-Mixed Text, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021.

[14] P. M, R. K, A. Hegde, K. G, S. Coelho, H. L. Shashirekha, Taming Toxicity: Learning Models for Hate Speech and Offensive Language Detection in Social Media Text, in: Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation, 2023. URL: https://ceur-ws.org/Vol-3681/T6-22.pdf.

[15] F. Balouchzahi, A. B. K, H. L. Shashirekha, MUCS@DravidianLangTech-EACL2021: Cooli - Code-Mixing Offensive Language Identification, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, 2021, pp. 323–329.

[16] 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.