International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(9s) 2026 pp. 1327-1347
https://doi.org/10.70917/ijcisim-2026-3556
Article

A Novel XLMR-DualCL Approach for Hate Speech Detection in Hindi-English Code-Mixed Data on Social Media

Amit Sharma[1], Rajni Bhalla[2]
[1] School of Computer Application, Lovely Professional University, Punjab, India
[2] School of Computer Application, Lovely Professional University, Punjab, India
rajni.b27@gmail.com (Corresponding author)

Abstract: - Automated identification of hate speech in Hindi-English code-mixed content remains a significant open challenge in multilingual natural language processing, largely because users on Indian social media platforms routinely blend two languages within a single utterance, employ phonetic transliteration of Hindi in Roman script, and convey hostility through indirect or culturally situated expressions that resist straightforward lexical analysis. This study introduces XLMR-DualCL, a unified detection pipeline specifically designed to address these difficulties. The architecture integrates noise-aware text preprocessing, diversity-oriented data augmentation, deep multilingual encoding via XLM-RoBERTa, a dual contrastive learning objective that simultaneously exploits self-supervised and supervised signals, and focal loss to compensate for the class imbalance endemic to real-world hate speech corpora. Comprehensive evaluation is conducted across five publicly available benchmarks spanning HASOC, TRAC-1, ICHCL, THAR, and Kaggle Hinglish, supplemented by a self-curated corpus of Instagram and YouTube comments and a merged multi-source dataset. Across all evaluation settings the model delivers consistent and competitive results, recording F1-scores of 88.89, 89.83, 91.17, 88.49, 90.29, 91.62, and 87.92 on the seven corpora respectively, together with a peak classification accuracy of 91.67 on the self-collected data. Precision and recall values remain closely aligned throughout, confirming that the system avoids trading one error type against the other. Taken together, these outcomes demonstrate that coupling cross-lingual contextual encoding with dual contrastive representation learning substantially advances the state of hate speech detection in noisy, informal Hinglish social media environments.
Keywords: - Hate speech detection, Hindi-English code-mixed text, XLM-RoBERTa, Dual contrastive learning, NLP Text Classification

1. Introduction

Social media platforms have revolutionized the way individuals communicate, exchange opinions, and engage in public discourse in local, national, and global contexts. Platforms like Facebook, Twitter, Instagram, WhatsApp, and YouTube have created highly dynamic digital arenas where users can produce, share, and amplify material in real time, accelerating and scaling information transmission. These platforms provide great opportunities for communication, awareness, education, and civic engagement but they have also contributed to the rapid proliferation of destructive online behaviors such as hate speech, abusive language, cyberbullying, misinformation, and harassment. Among such challenges, hate speech has emerged as a particularly serious concern. Hate speech can attack individuals or communities based on religion, caste, ethnicity, gender, language or any other social identity and thereby intensifying social division and causing psychological, cultural and even physical harm[1].

The problem is especially acute in the Indian digital ecosystem, where social media use has exploded and online conversations typically revolve on sensitive themes such as religion, politics, caste and regional identity. In a multilingual and socially diverse environment such hateful expressions are not always expressed in a straightforward or explicit manner but often in the form of sarcasm, coded phrases, offensive humor, symbolic references and context-dependent statements that are hard to interpret automatically. The problem gets a bit more complicated in Hindi-English code-mixed communication popularly termed as Hinglish which is very extensively used by Indian social media users in their day-to-day digital interactions. Hinglish text is a mixture of Hindi and English in the same sentence, Hindi words in Roman script ignoring the grammatical rules, abbreviations and slang, and highly informal terms that represent spoken language more than normalized written language. Thus, most existing text-processing techniques developed for monolingual and grammatically regular data typically fail to grasp the true content, purpose

and emotional tone of such posts.

Thus, hate issues require action in Hindi-English code-mixed text is a challenging Natural Language Processing issue requiring models to recognize social context, cultural references, semantic shifts and implicit animosity along with linguistic structure. A single post can include mixed scripts, transliterated words, regional idioms, irony, and confusing expressions whose meaning can only be inferred from contextual information, not from surface vocabulary alone. Additionally, the absence of large, standardized and balanced annotated datasets for Hinglish hate speech identification hampers the performance of traditional machine learning methods and affects the generalizability of taught systems. Simple keyword-based filtering or rule-based moderation is sometimes not enough, as hostile intent can hide beneath seemingly common conversational patterns[2].

To solve these constraints, researchers have increasingly used Artificial Intelligence, Machine Learning, Deep Learning, and advanced Natural Language Processing approaches to automate hate speech identification. These techniques enable the processing of vast amounts of social media data and enable more advanced analysis through feature extraction, context representation, sentiment interpretation and classification[3]. Specifically, deep learning and transformer-based models have demonstrated more potential than traditional approaches since they can capture semantic dependencies, contextual relationships in multilingual and code-mixed text much better. Thus, it is important to design robust and context-aware models for detecting hate speech in Hindi-English code-mixed social media data to foster safer digital spaces, enhance automated moderation, and encourage socially responsible communication in multilingual online environments[4].

The major aim of this research effort is to provide a robust and efficient framework for automatic hate speech detection in Hindi-English code-mixed social media content by utilizing XLM-RoBERTa based contextual encoding, dual contrastive learning and focal loss optimization

- To design a robust framework for hate speech detection in Hindi-English code-mixed social media text collected from Instagram and YouTube.
- To manage linguistic diversity and informal expressions more effectively by preprocessing and enriching the noisy code-mixed data by cleaning, normalization, tokenization and augmentation.
- To learn strong contextual representations of Hinglish text by combining XLM-RoBERTa with dual contrastive learning for better separation of hate and non-hate instances.
- To improve classification performance on imbalanced data using focal loss and to evaluate the proposed model with standard measures such as accuracy, precision, recall, and F1-score.

This study proposes XLMR-DualCL, a framework that combines XLM-RoBERTa for multilingual contextual encoding, data augmentation for representation diversity, dual contrastive learning for improved class separability, and focal loss for imbalance-aware optimization[5]. The proposed approach is designed to classify Hindi-English code-mixed social media comments into hate and non-hate categories more accurately across both public benchmark datasets and self-collected Instagram and YouTube data.

2. Literature Review

Social media and the internet have made it possible to spread hate speech. YouTube, Instagram, Twitter, and Facebook are all places where you can view this. This material is available in the form of text, pictures, and films. Using social media, individuals are able to swiftly express their ideas and views from any location. Conversely, social media has become a platform where impolite and harmful comments are often exchanged. Individuals can publish hateful things on social networking platforms without fear of being traced, which is a significant factor contributing to the surge in hate speech. Because users are able to conceal their true identities, they do not need to be concerned about being discovered posting remarks that are cruel or impolite. [6] The ease with which individuals can pick on, attack, or disseminate hatred toward other individuals or groups is facilitated by this element.

Numerous studies have employed a variety of terminology to characterize hate speech. Employing language that is impolite or unpleasant is required to insult an individual or a group of individuals based on their religion, gender, race, language, or background. Such speech is done with the intention of causing pain, humiliation, or spreading hatred, and it frequently results in misconceptions and divides individuals into distinct groups.

2.1 Studies in Hindi-English Code-Mixed Data

Classify hateful text data by using traditional machine learning, there is some literature review that is related to advance techniques for HS detection.

Mridha M. et al. [7]create a model that can effectively identify offensive text in SM posts written in Bangla and Bangla-English code-mixed language. The study involved gathering data from various social media platforms, followed by data preprocessing and cleansing. To extract meaningful features, four techniques were employed: TF-IDF, Word2vec, FastText, and BERT feature extraction. For classification purposes, a novel model called L-Boost was developed, which combines modified AdaBoost with BERT and LSTM models. The performance of the L-Boost model was compared to that of traditional ML models, resulting in an impressive accuracy of 95.11% and an F1-Score of 91.85%.

El-Alami and Quatik [8] proposed a cutting-edge approach was developed for detecting offensive language in multiple languages, known as Multilingual Offensive Language Detection (MOLD). This model leveraged the power of transfer learning and fine-tuning techniques. The study utilized two datasets, one in English and the other in Arabic. The initial step involved collecting tweets in both languages from Twitter, followed by preprocessing the text data and creating separate corpora for English and Arabic. The text data was then tokenized and fine-tuned using BERT, a powerful language representation model. The next phase involved classifying the text data using multiple deep learning approaches specifically designed for identifying hate speech. In the first MOLD dataset, mBert achieved an impressive accuracy of 91.36% and an f1-score of 91%. For the second Arabic dataset, AraBERT performed exceptionally well, achieving 90.7% accuracy and an f1-score of 93%.

Khanday A. et al. [9] proposed a groundbreaking approach for the identification of HS on Twitter during the COVID-19 pandemic, utilizing an ensemble learning-based model. The data collection process involved leveraging the Twitter API and targeting popular hashtags related to the pandemic. To create a reliable training set, tweets were manually categorized into two distinct groups, taking various factors into consideration. For feature extraction, TF/IDF, Bag of Words, and Tweet Length were employed. The study revealed that the Decision Tree classifier demonstrated notable effectiveness, exhibiting Accuracy at 97%, precision at 98, Recall at 97, and F1-S is 97. Nevertheless, the Stochastic GB classifier surpassed all other traditional ML classifiers, attaining exceptional performance metrics of 98.04% accuracy, 99 precision, 97 recall, and 98 F1-Score.

Chakavarthi B. et al. [10]Tamil and Tamil-English code-mixed comments were collected from YouTube by means of web scraping. These comments were subsequently annotated for binary classification (BACD) and fine-grained comment-based classification (FGACD). From the dataset, features will be extracted, a careful selection process was conducted. Traditional ML and deep learning techniques were applied to classify the datasets, allowing for a comprehensive evaluation of their performance. Additionally, the t-test was utilized to determine the statistical significance of the applied methods.

Sharma, Kabra, and Jain [11] Proposed a model for detecting Hindi-English mixed HS on social media, with the aim of developing HS detectors that can recognize such content on a regional level. To conduct the study, three datasets were used, namely track-I, HS, and HOT, and the data was preprocessed. In the experiment, the performance of traditional machine learning models was compared using simple surface features. The models were run on both raw data and data mapped using the proposed MoH method. Baseline models were also used for comparison, and the proposed technique was compared with other data transliteration techniques. These data transliterations were trained using the same Bert-based models that were used to train the MoH-mapped data. The results showed that MoH + M-Bert and MoH + MuRIL outperformed the other models.

Madhu et al. [12] Developed an ML-based model for detecting HS in code mixed data, and a benchmark dataset was created to conduct the experiment. For the first dataset, a conversational-based approach was used to classify HS in Hindi-English code-mixed data. Long conversations were collected to extract contextual information, and the two categories were manually added to the dataset: HOF (hate speech and offensive) and NOT (non-hateful speech and offensive). Text representation was done by extracting features, and ML was applied to classify the text data. The proposed system, which utilized SentBERT, LSTM, and Seq, achieved an F1-macro score of 89.2.

Sreelakshmi, Premjith, and Soman [13] Developed a model to investigate hateful speech in combined Hindi-English text, with the aim of classifying HS tweets in both languages. The proposed framework utilized three datasets, one taken from individuals who speak both English and Hindi, another taken from individuals who speak multiple languages, and a third taken from HASOK. After preprocessing the data using word embedding and classifying it

using SVM and RF, the proposed framework achieved its highest F1-Score of 80.85% using SVM+RBF.

2.2 Studies on cyberbullying and hate speech Text Data

Uma Maheswari and Priya [14] developed a novel foul language detection model for social media utilising Tree CNN combined with Adversarial Bi-LSTM networks. To deal with semantic understanding and unbalanced datasets issues, the study applied SMOTE and feature extraction approaches based on Word2Vec. Attention mechanisms were used to increase interpretability and to identify recurring objectionable tendencies in social media chats. Experimental results on benchmark datasets showed the effectiveness of the proposed model with higher accuracy and better interpretability than existing deep learning-based systems for offensive language identification.

Alanazi and Lin [15] proposed a machine learning method for severity identification of cyberbullying in tweets in Saudi-dialect. The authors constructed a dataset of 5819 tweets categorised into four classes: non-cyberbullying, low severity, medium severity and high severity. In this study Support Vector Machine (SVM), Naïve Bayes (NB) classifiers were used using Bag-of-Words (BoW) and TF-IDF feature extraction methods. To increase classification performance, different preprocessing and balancing procedures (random oversampling, synonym substitution, etc. The suggested BoW+SVM model had the best accuracy (92.23%), which shows the efficiency of balanced data and machine learning in detecting cyberbullying severity.

Pushpavalli et al. [16] presented an optimised sentiment aware deep learning framework, DSO-CAM-CDL for cyber threat detection on social media platforms. Dolphin-Sparrow Optimisation for feature selection, Convoluted Auto-Encoding Memory (CAM) and Customised Deep Learner (CDL) for sentiment prediction and threat analysis were incorporated into the model. The methodology was based on the detection of fake news, phishing, cyberbullying, identity theft and dangerous online actions utilising Twitter datasets. Experimental results revealed that the suggested model attained 99.1% accuracy with enhanced precision, recall and F1-score over the standard machine learning and deep learning techniques.

Imthiyas Banu and Thiyagarajan [17] Proposed the Nonlinear Evolutionary Seagull Optimised Bidirectional Gated Neural Network (NESO-BGNN) for online social networks English hate speech detection. The model was divided into three phases: preprocessing, keyword extraction and classification. Robust Scaling Normalisation was employed to eliminate noise and outliers, and the Nonlinear Evolutionary Seagull Optimisation algorithm was used for extracting the ideal keywords for hate speech classification. Finally, a Bidirectional Gated Recurrent Neural Network (BGRNN) was used in order to increase the classification accuracy and to lower the misclassification rates. The experimental investigation, utilising the Hate Speech and Offensive Language Dataset, showed that the suggested framework obtained greater accuracy with decreased detection time.

Imthiyaz Banu et al. [18] has designed a Censored Regressive Canonical Optimised Convolutional Deep Belief Classifier (CRCOCDBC) for multi-class hate speech detection in online social networks. The model used a Gestalt pattern recognition for stop word removal and a censored regression function for relevant keyword extraction from social media material. In this paper, Deep Belief Convolutional Neural Networks and Canonical Correlation approaches were combined to increase the accuracy of hate speech classification with low error rates and processing time. Experimental results demonstrated that the suggested model achieved greater authentication accuracy and precision compared to various existing hate speech detection methods.

Anand Raj et al. [19] introduced the Multi-Layer Perceptron and Feature Extraction for Detecting Bullying in Multimodal Content (MLPFE-DBMMC) framework for multimodal cyberbullying detection. The program used text, image and audio data obtained from social media platforms to detect bullying behaviour. The picture feature extraction is performed using CLIP, the text analysis using GPT-2 and the audio feature extraction using VGGish. The retrieved multimodal characteristics were fused and categorised using a CORAL-based multi-layer perceptron model using Improved Artificial Rabbit Optimisation (IARO) for hyperparameter tuning. Our suggested strategy achieved 94.44% accuracy on the MultiBully dataset, showing the usefulness of multimodal deep learning techniques for cyberbullying detection.

Almazroi et al. [20] established a hate behaviour detection methodology to detect the Islamophobic content on social media sites. The study classified the Islamophobic remarks based on severity and context by Natural Language Processing (NLP), Artificial Intelligence (AI), keyword recognition, tone analysis and machine learning approaches. The system was developed to detect poisonous comments and hate speech against Muslim communities on social media. The proposed method emphasised the significance of automated hate speech monitoring systems to decrease online toxicity and tackle the rapid spread of harmful content across digital channels.

Existing hate speech detection studies have shown good performance on several social media datasets, but many of them are developed for monolingual text, cleaner language settings, or general multilingual corpora rather than noisy Hindi-English code-mixed content used on Indian social platforms. In Hinglish data, users frequently mix Hindi and English words, write Hindi in Roman script, use slang, abbreviations, spelling variations, sarcasm, and context-dependent expressions, which makes straightforward lexical matching and conventional classifiers less reliable.

Hate in Indian online discourse is often tied to religion, caste, gender, politics, and nationality, and that such hostility may appear in implicit or indirect forms rather than in clearly offensive words alone. At the same time, available datasets are fragmented across tasks and platforms, class imbalance remains a major issue, and many systems do not jointly address contextual representation learning, data augmentation, and imbalance-aware

3. Proposed Methodology

In this chapter, we present the proposed XLMR-DualCL framework for automatic detection of hate speech in Hindi–English code-mixed social media data from platforms such as Instagram and YouTube. Due to the cohabitation of many languages, informal written styles, and dimension-sensitive terms (including hate speech among numerous other features), detecting hateful speech in such data is a next-to-impossible task. Often earlier terms cannot be interpreted in isolation because the same term can have very different meanings depending on the context. With many non-hate occurrences compared to hate discourse, the problem of class imbalance tends to make learning difficult for traditional models.

The new strategy represents one unified framework of a combination of these and other sophisticated techniques to help solve these challenges. It uses a multilingual transformer model called XLM-RoBERTa as the backbone encoder to extract semantic and contextual embeddings from code-mixed text proficiently. Additionally, the system combines self-supervised and supervised contrastive learning with a dual contrastive objective. This trains the model to learn a better representation in such a way that clusters together similar samples while separating others as far as possible in the embedding space.

Focal loss is used as the training criterion, designed to mitigate hard and class-imbalanced classes' effects. The pipeline starts with raw social media comments, then has preprocessing and contrastive learning-specific data augmentation. The text is tokenized and passed into the encoder of XLM-RoBERTa to create contextual embeddings. These embeddings are further refined through a projection head and then contrasted and classified, the final predictions being hate or no-hate.

3.1 Overview of the Proposed Framework

We propose a well-structured multi-stage pipeline called XLMR-DualCL to convert the raw Hindi–English code-mixed social media text into correct hate speech predictions. The framework, based on a sequence of certain defined steps, is meant to process and learn meaningful representations of textual data while accounting for the noisy, unstructured, and linguistically latent aspects of the nature of social media data.

First, databases of Instagram and YouTube are obtained, as they are the most used social networking platforms by people, so that they consist only of the actual utilization of Hindi–English code-mixed language (which may be abusive or hateful). The data collected is then preprocessed by removing any noise such as URLs, emojis, hashtags, and redundant punctuations. Absence of normalization techniques for spelling variations or code-mixed inconsistencies is applied, and duplicate entries are removed.

Data augmentation techniques are employed to improve the diversity and robustness of the data set. These are synonym replacement, back-translation, and controlled insertion or deletion of words. These transformations produce semantically similar paraphrases of the original text, which are valuable for contrastive learning.

The processed text is tokenized with the XLM-RoBERTa subword tokenizer for efficient handling of multilingual & code-mixed inputs. The tokenized sequences are fed to the XLM-RoBERTa encoder, which generates deep contextual representations by capturing both semantic and syntactic relations across languages.

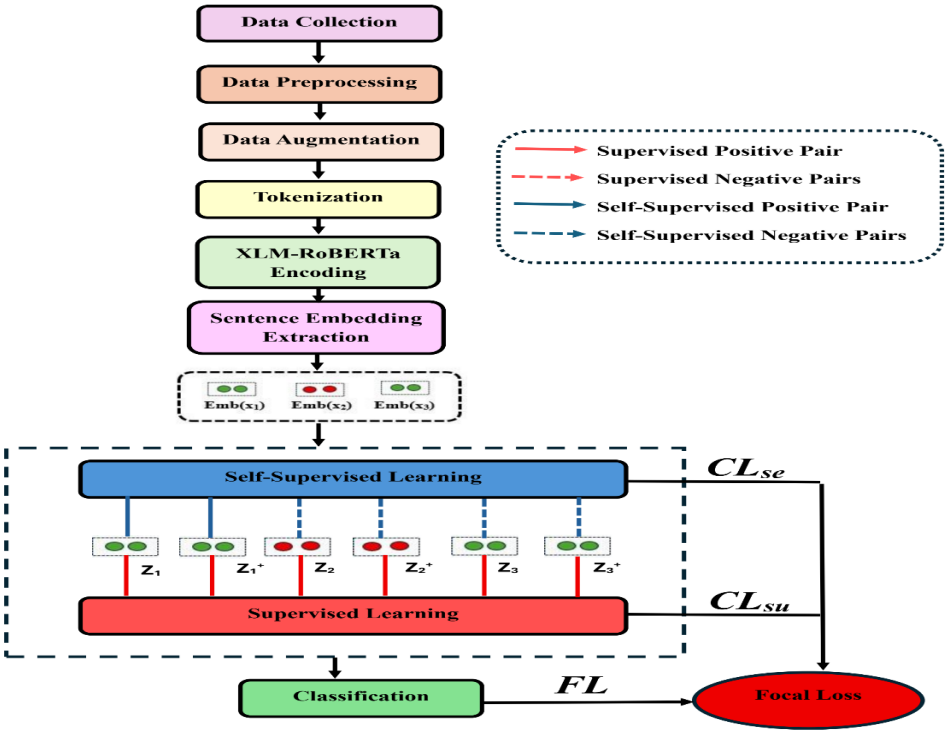


Fig 1. Framework for novel Approach for hate speech detection in Hindi-English code-mixed data

Dual contrastive learning, which unifies self-supervised and supervised contrastive losses, is the main building block of the framework. Such a strategy like this improves the class separability through enhancing the representation learning by attracting semantically similar samples and repulsing dissimilar ones to be placed farther away in the embedding space.

We apply a classification layer containing a softmax to classify if the comment is hateful or non-hateful. Focal loss is used during training for natural class imbalance, allowing the model to pay more attention to the difficult and minority classes.

Finally, the proposed framework can successfully leverage multilingual contextual encoding with two contrastive representation learning techniques and imbalance-aware optimization based on generative background data, leading to robust performance in detecting hate speech from Hindi–English code-mixed data.

3.2 Data Collection

Data Collection: Data collection is the first and foremost step in the process of building a reliable dataset for the detection of hate speech. In this research, data is collected from popular social media platforms, Instagram and YouTube, which are rich in Hindi–English code-mixed user-generated content. As seen in the framework diagram, initially we begin by selecting social media platforms, then we select domains or topics. These areas have domains in which hate speech is more likely to be active, such as discussions on religion, caste, gender politics, and nationality. This allows us to gather a wide range and representative set of comments.

Once the domain is chosen, data scraping is carried out to extract comments and replies from posts and videos. This involves either APIs or web scraping tools in accordance with platform policies and guidelines. In addition to the textual data, some additional context data related to times and engagement may also be stored, but user identities are anonymized for privacy purposes. The data you collect is now filtered to improve the quality. We keep only those comments that have both Hindi and English words so that the dataset is truly code-mixed. Examples: very short comments, comments containing only emojis, URLs, links, and too much irrelevant content.

After filtering the data, it goes to annotation and labeling. During this step, each comment is manually examined and annotated as hate speech or non-hate speech based on gold standard guidelines. "Hate speech" refers to any language that is directed at individuals or groups based on certain characteristics like religion, caste, gender, and

nationality. To ensure consistency and reliability, the labeling is performed by trained annotators, with checks on agreement between them.

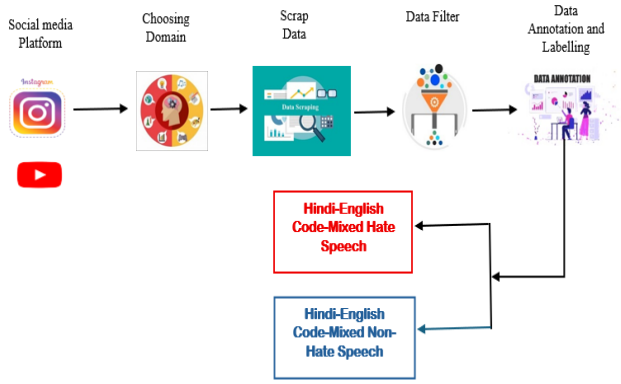


Fig 2. Data collection and labelling Pipeline

Lastly, the processed dataset is separated into Hindi–English code-mixed hate speech and Hindi–English code-mixed non-hate speech illustrated through the diagram. We make use of this labeled dataset to train and assess the performance of the proposed model. Also we are taking publicly available datasets which details given below

3.2.1 Public Datasets

TRAC-1 Dataset

TRAC-1 is the Colloquial Dataset, a benchmark for several aggression tasks proposed as part of the Shared Task on Aggression Identification COLING 2018. The dataset collects 15,000 Facebook posts and comments in English and Hindi (both Roman and Devanagari) formats with three-class classification: Overtly Aggressive (OAG), Covertly Aggressive (CAG), and Non-Aggressive (NAG). Since social media text is often subjective and context-dependent, the value of this dataset lies in its ability to capture explicit as well as implicit forms of aggression. On the other hand, TRAC-1 offers an established testbed for building and validating automatic systems for detecting hate speech and aggressive content[6].

HASOC Dataset

HASOC is a series of FIRE shared tasks on Hate Speech and Offensive Content Identification since 2019, featuring multilingual datasets including Hindi-English code-mixed tweets and posts. Subtask A performs binary classification (HOF: hateful/offensive vs NOT), while Subtask B provides fine-grained labeling (HATE, OFFN: offensive, PRFN: profane), with examples like Hindi train sets of 4,665 posts[21].

CHCL Dataset

Identification of conversational hate speech in code-mixed languages (ICHCL) dataset is a contextual social media corpus introduced for the task of hate speech detection from conversational dialogue. Previous datasets have viewed each post in isolation; ICHCL, however, maintains the structure of Twitter conversations with parent tweets and replies included alongside comments. This dataset covers Hindi-English code-mixed text and has been manually annotated into hateful/offensive and non-hateful classes. Due to its contextual nature, it is particularly appropriate for studying shifts in meaning across dialogue and for implementing more effective systems for detecting hate speech[12].

THAR Dataset

Targeted Hate Speech Against Religion dataset is a Hindi-English code-mixed corpus of 11,549 YouTube comments that have been manually annotated for detecting religious hate speech. THAR, however, is a dataset specifically for finding anti-religion content and the religion targeted (rather than general hate speech datasets). The dataset contains two subtasks: binary classification for anti-religion vs non-anti-religion and multi-class classification for Islam, Hinduism, Christianity, and 'none'. THAR is a corpus that accommodates both code-mixed data and a multi-tier fine-grained annotation scheme, consequently making it suitable for targeted hate speech research in Indian low-resource language settings[22].

3.3 Text Preprocessing

The following standardized pipeline cleans and normalizes annotated tweets, preserving semantic integrity while eliminating extraneous elements:

- Punctuation and Special Character Removal: Strip non-alphabetic symbols like |, ;, &, !, and? to reduce noise without altering word boundaries or intent.
- Hashtag Normalization: Convert hashtags to plain words (e.g., #jihad → jihad) to retain lexical content and prevent tokenization fragmentation.
- Lowercasing: Transform all text to lowercase for case-insensitive matching, crucial for code-mixed data with inconsistent capitalization across scripts.
- URL, Numeric, and Emoji Removal: Eliminate hyperlinks, digits, and emojis (e.g., · → empty) to focus on textual semantics, as these rarely contribute to hate/offensive signals in code-mixed contexts.
- Stopword Elimination: Remove common stop words from Hindi (e.g., है, हैं) and English (e.g., the, is) lists tailored for code-mixed corpora, minimizing sparsity while retaining domain-specific terms.

This pipeline yields clean uniform input ready for embedding or modeling, typically implemented via libraries like NLTK, regex, and Indic NLP tools.[23].

3.4 Data Augmentation

To improve model generalization and better enable contrastive learning, the proposed method includes a data augmentation stage such that alternatives of each input comment are generated in which annotations remain unchanged. This is especially crucial in the context of hate speech detection since there are few such instances and they can be expressed wildly, in the correct style across social media channels.

More formally using x for a normalized comment, the augmentation function generates:

$$A(x) = \{x_1, x_2, \dots, x_k\} \qquad (1)$$

In equation each $x_i$is an augmentation of x with similar semantics and label.

For every normalized comment, several augmented variants are generated with controlled transformation methods. For example, one of which is synonym replacement where a few non-stopwords are replaced with synonyms using lexicon resources or transformer based masked language models. The second method is back translation which translates a sentence to an intermediate language to generate different paraphrased alternatives that maintain the same meaning of a source text (English or Hindi). The first strategy is random insertion and deletion of minor words (in our case intensifier) that do not change the meaning of the whole sentence, in a controlled way. This produces a varied collection of augmented sentences for each input sentence. The augmented data is applied in two major ways. First, it is for additional training data which helps the model to generalize better over different ways of expressing the same content. Second, it plays a role in contrastive learning by treating two different augmented views of the same comment as semantically as each other which helps the model generalize better and create more context-aware representations[24], [25].

Additionally, the framework applies stochastic dropout at the representation level to produce implicit instances of a single input. This technique (like that of dual contrastive learning) adds random noise on representations to assist the model in understanding invariant semantic characteristics for hate speech detection.

3.5 Tokenization

We use XLM-RoBERTa tokenizer to convert each of the preprocessed and data augmented input comments into a sequence of tokens, Tokenization, which is the process of turning raw text into a format that can be fed to a model. XLM-RoBERTa has a Sentence Piece based subword tokenization method trained on 100 languages with large monolingual corpora (~2.5 TB of Common Crawl per language). This technique leaves rare words handled well by the model, also helps spell variation and separates languages.

This property is especially helpful for Hindi–English code-mixed data that contain sentences which consist of a mixture of English words and Romanized spellings of the Hindi words (whose spellings may also vary from sentence to sentence.) The tokenizer can segment words in smaller subword units rather than treating each word as a separate unit which allows to better handle unknown or uncommon terms.

The tokenizer generates a sequence of tokens for each input sentence:

$$(x_1, x_2, \dots, x_n) \quad (2)$$

In the equation n represents the length of the sequence after special tokens like classification and separator markers are added. Afterwards, each token is mapped to an integer-based index based on the predefined vocabulary of XLM-RoBERTa creating a sequence of token ids[25].

Native sequence lengths vary between samples, so to make the input vector for each sample the same size you need to apply pad/ truncate. So, again short sequences padded and longer ones truncated to the by default fixed length maximum. XLM-RoBERTa model can easily perform processes quickly as there is standard representation of input data,

3.6 XLM-RoBERTa Encoder and Contextual Representation

XLM-RoBERTa Based Framework The proposed framework adopts XLM-RoBERTa (XLM-R) as its base encoder to obtain contextual representations of the input text. XLM-R is a large-scale transformer-based Multilingual Language Model based on Common Crawl data of 100 languages. Its coverage of extensive multilingual training makes it an excellent choice for cross-lingual understanding or code-switched text as well, which uses multiple languages in the same sentence.

XLM-RoBERTa processes the tokens from a tokenized input sequence sequentially through many layers of transformer blocks, which consist of self-attention and feed-forward networks. These layers allow the model to find contextual relationships between terms in the same sentence. Consequently, all tokens are conveying their intrinsic meaning and in terms of the surrounding tokens in the sentence.

Formally, let the input token sequence be represented as embeddings X. The encoder produces contextualized representations H, which can be expressed as:

$$H = XLM - R(X) \quad (3)$$

Im the equation 3 $H = (h_1, h_2, \dots, h_n)$, and each $h_i \in R^d$ represents the contextual embedding of the i-th token, with $d$being the hidden dimension.

The self-attention mechanism provides the ability for any token to come visit all of the other tokens in the sequence, allowing our model to retain word order as well as has relationships across languages and long-distance dependencies. Hindi–English code-mixed text is such that meaning can only be determined by the co-occurrence of words from more than one language; The ability to put hate speech in context, however, is very important for capturing indirect expressions of hate via linguistic nuance[26].

3.7 Sentence Embedding Extraction

In sentence-level tasks such as hate speech detection, we need to pool a sequence of token-level representations into one vector that has length independent of the input data (sentence). This is done using the special [CLS] token representation from the last layer of XLM-RoBERTa. The [CLS] token is trained to predict the overall meaning of the input sequence and common in transformer-based classification tasks.

We denote hidden state corresponding to the [CLS] token as $h_{[CLS]}$ classification. The embedding for the input to this sentence is defined in equation 4:

$$s = h_{[CLS]} \quad (4)$$

The model converts each sentence to a vector, used as input of the following layer (like projection or classification), which is compact in that it summarizes the entire contents of the sentence. However, other alternatives may also be used like mean-pooling or max-pooling over all token embeddings; as such, we chose to work with the [CLS] representation for its simplicity and proven efficiency in previous hate speech detection studies[27].

3.8 Projection Head Architecture

An additional projection head is utilized for transforming the extracted sentence representation into a new space for contrastive learning. Simply put, the projection head is an additional layer in the neural network that tries to polish the sentence embedding before similarity-based learning. It is primarily used to assist the model in contrastive learning of how different sentence representations are compared.

We implement the projection head as a small multi-layer perception (MLP) with two fully connected layers that are separated by non-linear activation function. Such design enables the model to extract more informative and discriminative representations for subsequent contrastive learning. Instead of employing contrastive loss directly on the base sentence embedding, the framework first utilizes the projection head to transform representation in a more appropriate space for measuring semantic similarity and dissimilarity between samples.

One major benefit of the projection head is that it decouples the contrastive representation space from the classification representation space. Consequently, the model will be able to optimize one representation space that represents the strong relational structure among samples for learning while keeping our original sentence embedding for the final hate speech classification task. This enhances the quality of learnt features and helps in improved generalization. Thus, the projection head becomes a significant intermediate module capable of improving the overall framework in learning strong contextualized representations.

3.9 Dual Contrastive Learning Framework

The main interaction with the proposed methodology is that a tandem contrastive learning framework is added to the hate speech detection pipeline. It leverages both the structure of unlabeled data and explicit label information by using the framework of self-supervised contrastive learning (CLse) and supervised contrastive learning (CLsu).

Intuitively, self-supervised contrastive learning pulls different augmented or noisy views of the same sentence close in the embedding space and pushes separate views of different sentences away. This allows the model to learn representations that are agnostic to superficial differences like slight changes in text or dropout noise.

In contrast, supervised contrastive learning groups together the representations of sentences that share a label (hate/non-hate) and separates those with different labels, essentially directly constraining the decision boundary of the classification task.

$B = \{(s_i, y_i)\}_{i=1}^{M}$ be a mini batch of sentence embedding and their labels. At training time, one or more contrastive views are created for each sentence embedding s_i, either by means of text augmentation (Section 3.4) or through independent dropout masks in the projection head to generate representations $z_i^{(1)}, z_i^{(2)}, \ldots$

With this implication in mind, the dual contrastive objectives are defined as

$$L_{CL} = L_{se} + L_{su} \qquad (5)$$

In the equation 5 L_{se} are self-supervised contrastive loss and L_{se} Lsu is supervised contrastive loss. The subsections below describe these components in more detail[28].

3.9.1 Self-Supervised Contrastive Learning

Without the use of class labels, self-supervised contrastive learning is fine-tuned. The idea here is to bring two different views of a same sentence close together in the embedding space, and push representations of different sentences farther apart. This is important as social media comments have varying spellings, informal writing, code-mixing and additional syntactic variations. Such changes mustn't affect good representation.

Given a batch of sentence embeddings s_i, two stochastic views $z_i^{(1)}$ and $z_i^{(2)}$ are produced by applying the projection head with independently sampled dropout masks. Both these views belong to the same positive pair as they both come from one underlying sentence[29].

We take all the representations obtained from other sentences in the batch as negative samples.

Let $Z = \{z_1, z_2, \ldots, z_{2M}\}$ denote the set of all contrastive representations when each sentence produces two views. For a given anchor representation z_j, let z_j^+ denote its corresponding positive view.

The similarity between two representations z_a and z_b is measured using cosine similarity shown in equation 6:

$$sim(z_a, z_b) = \frac{z_a^\top z_b}{\|z_a\| \, \|z_b\|}. \qquad (6)$$

The self-supervised contrastive loss is then defined as an InfoNCE-style objective over the batch:

$$L_{se} = -\frac{1}{2M}\sum_{j=1}^{2M} \quad log \frac{exp(sim(z_j, z_j^+)/T_{se})}{\sum_{k=1}^{2M} \quad 1[k \neq j] exp(sim(z_j, z_k)/T_{se})}, \qquad (7)$$

In equation 7, where $T_{se} > 0$ is a temperature hyperparameter and $1[\cdot]$ is the indicator function.

This loss pushes the model to increase similarity of positive pairs while decreasing the similarity with remaining samples in the same batch. In the case of Hindi–English code-mixed hate speech detection, this allows the encoder to become resistant to small lexical and/or syntactic variations in the expression of a hate/non-hate message, such as those introduced due to code-mixing or informal spelling.

4.9.2 Supervised Contrastive Learning (CLsu)

Self-supervised contrastive learning can learn useful invariances from data but does so without directly using label information. Supervised contrastive learning generalizes it by considering all samples with the same label to be positive for each other and samples from different labels to be negative.

For a batch $B = \{(z_i, y_i)\}_{i=1}^{M}$, the supervised contrastive loss is defined as

$$L_{su} = -\sum_{i=1}^{M} \frac{1}{N_{y_i}-1} \sum_{j=1}^{M} 1[i \neq j] 1[y_i = y_j] log \frac{exp(sim(z_i,z_j)/T_{su})}{\sum_{k=1}^{M} 1[k \neq i] exp(sim(z_i,z_k)/T_{su})} \quad (8)$$

where N_{y_i} is the number of samples in the batch with label y_i, and $T_{su} > 0$ is a temperature parameter for the supervised loss.

This goal gathers all hate remarks within the batch and all non-hate feedback in the batch into two separate but distinct clusters in the contrastive space such that compared to each non-hate comment, any hate-language feedback has a lot better notion of embedding than any random comparison to any one other feedback (non-dedicated) for a selected point at a ratio these days defined. Nonetheless, this structure is more complex because the comments change depending on the target group, topic or style, and as such it has all freedom to lay out so that loss minimizes[30].

Since with supervised contrastive learning, same-ground truth label comments are attracted while different labels repelled, it is specific useful for Hindi–English code-mixed data in separating non-hate containing abusive terms (e.g., usage of familiarity among friends) from genuine hate speech.

The overall dual contrastive loss is shown in equation 9:

$$L_{CL} = L_{se} + L_{su}. \quad (9)$$

During training, this loss is optimized jointly with the focal loss from the classification layer, as described later.

3.10 Classification Layer

Classification layer that classifies each input comment either to hate or non-hate classes. Once the sentence has been passed through the encoder, you now have a sentence-level representation of the passage, and this goes to the classification layer that will decide. This layer transforms the features learned by the previous layers into scores specific to each class.

The encoder generates a sentence embedding, and this embedding can then be fed to a linear layer, which outputs scores for the two classes in this framework. These scores are then routed through a softmax layer to turn them into probabilities. Which we can simply write as:

$$\hat{y} = softmax(Ws + b) \quad (10)$$

In equation 10 s is the sentence representation, W is the weight matrix, and b is the bias term.

The output is a probability of seeing the comment from each class[31].

There are two uses of the predicted probabilities. They are then used to calculate the loss function and learn accurate decision boundaries that the model can use during training. At test or inference time, the class with the highest probability is chosen as the final prediction. In this way, the classification layer is a key component that converts contextual sentence representations to the final output of hate speech detection.

3.11 Focal Loss Optimization

The proposed framework follows with focal loss for weakly supervised training to overcome the challenges of

class imbalance for hate speech detection. Non-hate comments predominantly outnumber hate comments in real-world datasets, particularly those collected from platforms like Instagram and YouTube. This imbalance may result in standard loss functions such as cross-entropy giving disproportionately higher weight to the majority class during training. This could lead the model to be biased towards predicting not hate and increasing false negatives for minority hate.

To lessen this issue, focal loss alters the usual classification loss by assigning more weight to difficult and misclassified examples while decreasing the impact from simple examples that are already correctly predicted by the model. Write simple version of focal loss:

$$FL = -\alpha(1 - p_t)^{\gamma} log(p_t) \qquad (11)$$

In equation 4.11 p_t is the predicted probability of true class, α is a weighting factor and γ is focusing parameter. This factor $(1 - p_t)^{\gamma}$ reduces the contribution of loss by easy samples, encouraging the model to use less importance for these easy cases in favor of focusing harder on harder cases. The second is the weighting factor, which aims to balance the influence of each class on the loss function, so that the hate class will receive more attention during model training[32].

Focal loss is useful in the context of Hindi–English code-mixed hate speech detection since it can help to address not only the imbalance in class proportions, due to lower number of hateful comments, but also because hateful comments are often more subtle, implicit or dependent on context. Those examples are less difficult to classify than non-abusive comments. Focal loss encourages the model to learn more discriminative decision boundaries by focusing on such hard instances. Thus, its incorporation increases the network's capacity to not only detect rare and hard instances of hate speech but also boosts the overall performance in imbalanced classification scenarios.

3.12 Final Prediction Mechanism

Then, in the final prediction stage, we employ a joint optimization of the dual contrastive learning framework and focal loss to provide accurate predictions with trained model parameters. When doing inference, this mode works in the forward-pass only (the encoder + projection head (if any) + classification layer). At this stage, it does not utilize the contrastive and focal loss functions.

This is how the prediction process goes for an unseen input comment:

Preprocessing: Your data is preprocessed with predefined preprocessing functions on text having inputs. Here noises like special characters, multiple spaces and mixed formatting are cleared to get the same-2 input form.

Tokenization: Processing the text and tokenizing it with XLM-RoBERTa tokenizer. The right input sequence is prepared using different special tokens.

Encoding: It passes the tokenized sequence through XLM-RoBERTa encoder to get contextual hidden representations. The last token contains the sentence-level representation, which encodes the overall semantic meaning of the input.

Classification: The obtained representation is input into a classification layer to further generate the output logits, which are converted to predictions via softmax.

While we do not explicitly apply the contrastive learning components during inference, they still have an imprint through parameters learned by the model. The embeddings learned by dual contrastive loss are more well-structured and separable in the embedding space. Thus, it can easily differentiate hate vs non-hate comments even when the same word is used in a different contextual meaning.

3.13 Evaluation Matrix

A classification model must be evaluated using transparent and robust evaluation metrics. Model performance: In the present study, the metrics used to evaluate model performance are accuracy, precision, recall and F1-score. These metrics are used to know how good models differentiate between hate and non-hate comments.

All these evaluation measures are based upon confusion matrix. Confusion matrix is a table that compares the actual class labels with predicted class labels from the model. This indicates the number of comments correctly and incorrectly classified in binary hate speech detection into two categories, hate and non-hate. There are four values that reside in the confusion matrix, they are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). True Positive: The hate comments which are accurately labeled as Hate. For example, True Negative is non-

hate comments which are classified as non-hate. False Positive = non-hate comment is incorrectly classified as hate and vice versa False Negative = Hate comment is incorrectly classified as non-hate

These four values are crucial because they illustrate the own behavior of the model. Not only do they show whether the prediction was correct or not, but they distinguish what kind of error the classifier made. This is especially important when it comes to hate speech detection, since both kinds of mistakes lead to issues. False positives on benign comments increase fairness losses, while false negatives on hateful comments decrease system safety and utility[33].

The basic representation of a confusion matrix for binary classification is:

Actual Class	Predicted Hate	Predicted non-hate
Hate	TP	FN
Non-Hate	FP	TN

Table 1: Confusion Matrix of Actual and Predicted Labels

3.13.1 Accuracy

Accuracy indicates the overall correctness of the model. This is the ratio between all correctly identified positive comments and negative comments divided by all comments found in the dataset.

The formula for accuracy is shown in equation 12:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

If the accuracy value is high, then it means that most predictions made by the model are correct. But accuracy is not all the time enough, especially for unbalanced dataset. In these scenarios, a model can still get a very high accuracy percentage but really suck at the hate class.

3.13.2 Precision

Precision measures how many comments predicted as hate are hate. It is based on how good the positive predictions are from the model.

The formula for precision is in equation 13:

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

High precision means fewer false positives by the model. This is key because this avoids falsely labelling non hate comments as hateful. As such, precision describes how reliable the model is when it predicts the hate class.

3.13.3 Recall

This means how many of the actual hate comments are detected by the model. The score indicates how well the model captures hateful content from the data set.

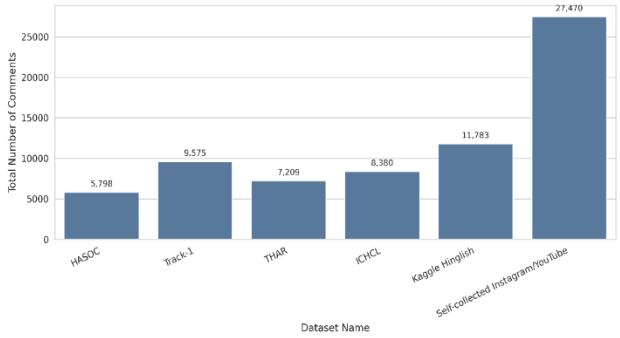
The formula for recall is in equation 14:

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

A model with high recall value, misses fewer hate comments. This is critical in hate speech detection since the hidden hateful content could continue propagating throughout online platforms. Hence, Recall explains the model's sensitivity towards Hate class.

3.13.4 F1-Score

F1-score is the so-called harmonic mean between precision and recall. It is used when false positives and false negatives are equally important.



The formula for F1-score is:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (15)$$

F1-score provides an overall perspective of how well the model performed. In cases where precision is high, but recall is low (or vice versa), the F1-score will also be lower. This is why F1-score can be a handy metric to assess the performances of models when it comes to hate speech detection tasks.

For the classification models, the evaluation metric used by this study was accuracy, precision, recall and F1-score. These metrics are calculated from the confusion matrix, which enables a more in-depth evaluation of model performance. Accuracy is how right you are on a large scale, precision checks how reliable your hate predictions are, recall measures how able a model can be in detecting real comments that contain hate speech, and finally the F1 score gives us a balance between precision and recall. Thus, these evaluation measures are appropriate for evaluating the performance of Hindi-English code-mixed hate speech detection models in a scientific and academic way.

4. Results and Discussion

The experimental setup is based on a combination of data collected by the authors and publicly available datasets. Standardized benchmark data from public datasets are ideal for comparison with previous studies, while our self-collected Instagram and YouTube datasets represent more recent, noisy, and diverse social media content. Hate speech detection is particularly affected by this aspect because real-world comments usually contain informal spellings, code-mixing, repeated characters, slang and context-dependent expressions.

Table 2: Overview of Datasets Used in the Study

Dataset Name	Total Number of Comments
HASOC	5798
TRAC-1	9575
THAR	7209
ICHCL	8380
Kaggle Hinglish	11783
Self-collected Instagram/YouTube	27470

Fig. 3 shown, the six Hindi–English code-mixed datasets used in this work are characterized by their distribution through comments. Among the three self-collected Instagram/YouTube datasets, HASOC has the fewest comments.

This variation demonstrates the diversity in models

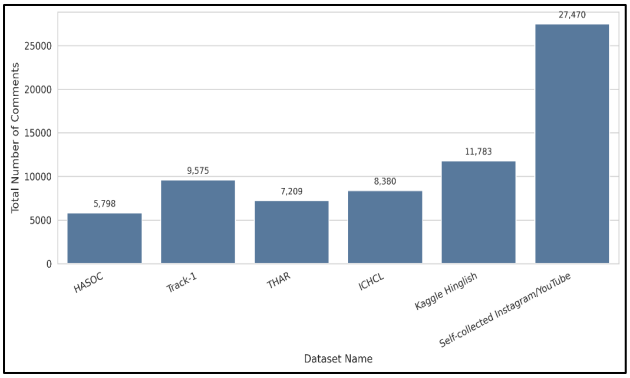


Fig 3. Bar Graph for Overview dataset

based on different sources and sizes of datasets which help assess their performance across real-world applications.

4.1 Experimental Setup and Implementation

This work proposes a systematic experimental protocol to experimentally evaluate conventional machine learning, deep learning and transformer-based approaches for hate speech detection from Hindi–English code-mixed social media text. All the experiments were conducted in Python, a language that has strong support for natural language processing and machine learning & deep learning research. The implementation was carried out using Google Colab for training the model and experimenting on a large scale, while a local system, an Intel Core i7 processor with 16 GB of RAM and NVIDIA graphics support, was used to develop the code, preprocess data, and test it at an intermediate level. This established a cohesive, uniform environment that could work with both lighter and heavier tasks without sacrificing convenience.

4.2 proposed model Results

The model achieves high results on all datasets, with final F1-scores varying from 87.92% on the merged dataset up to 91.62% on the self-collected Instagram and YouTube dataset. Similarly, it also shows a strong performance on TRAC-1 and Kaggle Hinglish, where its F1-scores are 91.17% and 90.29%, respectively. The experimental results demonstrate the effectiveness of the proposed approach on various benchmark datasets and real noisy social media text.

Similarly, the accuracy values also continue to be high and stable across datasets at 91.67% on the self-collected corpus. Precision and recall are comparable, which implies that the model can detect hate speech accurately while minimizing blinded hateful comments. It is necessary to strike this balance in the detection of hate speech, where a false positive and false negative will both limit the value of such a system.

The combination of multilingual contextual encoding and dual contrastive learning results in strong performance on both XLMR-DualCL. The deep semantic representations provided by XLM-RoBERTa can suit code-mixed and noisy social media text better; dual contrastive learning can help the model compacting images of similar and dissimilar samples in the embedding space more efficiently. These enhance the learned representations to be more discriminative and robust against spelling variation, transliteration, slang use or contextual ambiguity.

Table 3. Performance of XLMR-DualCL for Classifying Hate Speech Dataset

Dataset	No. of Comments	Accuracy	Precision	Recall	F1-Score
HASOC	5798	88.94	89.70	88.10	88.89
ICHCL	8380	89.88	90.62	89.05	89.83
TRAC-1	9575	91.26	91.95	90.40	91.17
THAR	7209	88.52	89.10	87.90	88.49
Kaggle Hinglish	11783	90.34	91.00	89.60	90.29

Self-Collected (Insta + YT)	27470	91.67	92.30	90.95	91.62
Merged Data	70215	88.92	89.55	87.30	87.92

Table 3 model has also achieved excellent results on the merged dataset (larger and more interspersed), with an accuracy of 88.92% and an F1-score of 87.92%. This shows that the proposed framework generalizes for various datasets and is not restricted to a single corpus. Overall, the overall results confirm that XLMR-DualCL has better performance compared with the baseline models and can achieve competitive classification performance in this work.

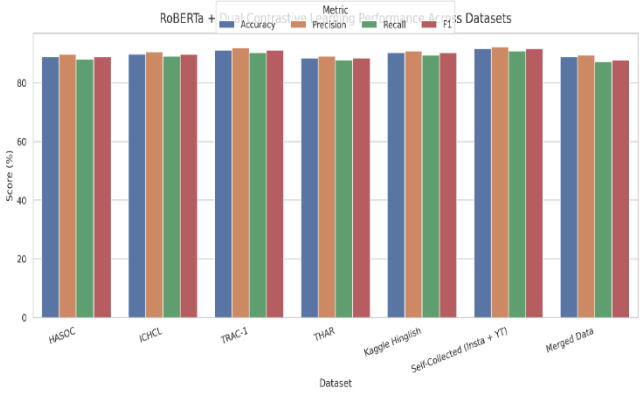


Fig 4. XLMR-DualCL Performance Across Datasets

Fig. 4 compares accuracy, precision, recall and F1-score across all datasets, showing that the proposed method maintains strong and stable results. Since the model is a classification problem, the figures in the confusion matrix are conditional frequencies of class prediction behavior, with diagonal cells providing correct predictions while off-diagonal cells indicate errors.

4.3 Dataset wise Comparative Analysis

comparative analysis of the proposed XLMR-DualCL model against recent literature and the baseline models report. The comparison focuses on the datasets used in this paper, namely HASOC, TRAC-1, ICHCL, THAR, Kaggle Hinglish, the self-collected Instagram/YouTube corpus, and the merged dataset. The main goal is to understand not only which model performs best, but also why particular model families behave differently under code-mixed, noisy, and socially generated text.

Table 4. Compares the HASOC-related studies with the proposed model

Proposed Model	**Method**	**Accuracy (%)**	**F1-Score (%)**
Ramiandrisoa et al. [34] proposed RoBERTa-based multitask learning	RoBERTa-based multitask learning	84	Macro-F1 0.79; Weighted-F1 0.84
Chopda et al. [35]	TabNet + PCA + MuRIL embeddin	85	86

	gs		
BERT baseline	BERT fine-tuning	81.56	80.00
CNN baseline	CNN	82.63	81.44
Proposed XLMR-DualCL	**XLM-RoBERTa + dual contrastive learning + focal loss**	**88.94**	**88.89**

Table 5. Compares the TRAC-1 related studies with the proposed model

Proposed Model	**Method**	**Accuracy (%)**	**F1-score (%)**
Sharma et al.[11]	MoH transliteration + MuRIL / mBERT	Reported improvement only	Average +4% F1 over best baselines
Putra et al. [36]	BERT + Advanced CNN	78	76.00
BERT baseline	BERT fine-tuning	79.24	80.63
CNN baseline	CNN	76.13	76.42
Proposed XLMR-DualCL	**XLM-RoBERTa + dual contrastive learning + focal loss**	**91.26**	**91.17**

Table 6. Compares the ICHCL-related studies with the proposed model

Proposed Model	**Method**	**Accuracy (%)**	**F1-score (%)**
Madhu et al. [12]	SentBERT + LSTM / weighted KNN pipeline	Not reported as a single accuracy value	89.2
BERT baseline	BERT fine-tuning	80.28	80.54

CNN baseline	CNN	74.29	73.70
Word2Vec + LR baseline	Word2Vec + Logistic Regression	60.26	61.26
Proposed XLMR-DualCL	**XLM-RoBERTa + dual contrastive learning + focal loss**	**89.88**	**89.83**

Table 7. Compares the THAR related studies with the proposed model

Paper title / source	Method	Accuracy (%)	F1-score (%)
Deepawali et al.	MuRIL	80	78
Word2Vec + Logistic Regression baseline	Traditional ML	63.66	62.94
CNN baseline	CNN	81.57	77.72
Chapter 5 BERT baseline	BERT fine-tuning	81.12	80.60
Proposed XLMR-DualCL	**XLM-RoBERTa + dual contrastive learning + focal loss**	**88.52**	**88.49**

4.4 F1-score comparison across datasets

The F1 comparison makes the thesis contribution very clear. The proposed method improves all datasets, and the gains are not random. The largest improvement appears on TRAC-1, followed closely by ICHCL and the self-collected corpus. These are precisely the datasets where context, spelling variation, or conversational ambiguity are strongest. That pattern suggests that XLMR-DualCL is not merely memorizing easier lexical regularities; it is solving the harder representation problem that the earlier models struggle with.

Table 8. F1-score comparison across datasets with Baseline Model

Dataset	baseline / source	Baseline F1 (%)	Proposed F1 (%)
HASOC	CNN baseline	81.44	88.89

ICHCL	BERT baseline	80.54	89.83
TRAC-1	BERT baseline	80.63	91.17
THAR	BERT baseline	80.60	88.49
Kaggle Hinglish	TF-IDF + Logistic Regression	83.70	90.29
Self-Collected (Insta + YT)	CNN baseline	83.78	91.62
Merged Data	BERT baseline trend	80.97	87.92

From an evaluation perspective, F1 is the most persuasive metric for this problem because it reflects the trade-off between precision and recall. A classifier that only improves accuracy by rejecting uncertain examples may still miss hateful content, which is undesirable in moderation systems. XLMR-DualCL raises F1 while keeping precision and recall close together, and that is a stronger result than accuracy alone. The table therefore supports the central claim of the thesis: better representation learning produces better practical hate detection.

5. Conclusion

This work has examined the problem of automatically recognising hate speech within Hindi-English code-mixed content shared on social media platforms, a task made particularly demanding by the interplay of transliteration, informal orthography, mixed-language syntax, and culturally embedded expressions of hostility that cannot be resolved through simple vocabulary lookup. In response, we developed XLMR-DualCL, an end-to-end pipeline that combines systematic preprocessing, label-preserving data augmentation, cross-lingual contextual encoding through XLM-RoBERTa, a dual contrastive objective that draws on both self-supervised and supervised learning signals, and focal loss to handle the pronounced class imbalance characteristic of hate speech datasets. Empirical evaluation spanning seven diverse corpora including HASOC, TRAC-1, ICHCL, THAR, Kaggle Hinglish, a self-curated Instagram-YouTube collection, and a merged multi-source dataset confirms that the framework achieves reliable and competitive results, with F1-scores ranging from 87.92 to 91.62 and a top accuracy of 91.67 recorded on the self-collected data, indicating strong generalisation across both established benchmarks and real-world noisy corpora. The experimental evidence further establishes that investing in richer contextual representation yields measurable gains in detection quality for code-mixed settings, particularly in cases where a model must discriminate between genuinely hateful remarks and surface-similar but benign conversational utterances that share vocabulary with hate speech. The near-symmetry of precision and recall figures observed across all datasets signals that XLMR-DualCL avoids the common failure mode of optimising one metric at the cost of the other, a property of direct practical importance given that both over-blocking of innocent content and under-detection of harmful content carry real social costs. The combined evidence therefore supports the conclusion that pairing multilingual transformer-based encoding with dual contrastive representation learning and imbalance-aware loss optimisation constitutes a sound and practically viable strategy for tackling hate speech detection in the linguistically complex, culturally specific, and stylistically diverse environment of Indian social media.

Several directions merit further investigation. Extending the current binary hate versus non-hate scheme to a multi-label taxonomy that captures the specific nature of harm, such as religious hostility, caste-based discrimination, gender-directed abuse, or implicit derogation, would significantly increase the diagnostic value of automated moderation systems because knowing what kind of harmful content is present matters as much as knowing that it is present. Integrating multi-modal signals, particularly images, audio fragments, and video context that frequently accompany social media posts, represents another promising avenue since hateful intent is not always conveyed

through text alone. Additionally, exploring lightweight or distilled variants of XLM-RoBERTa would facilitate deployment on resource-constrained platforms and enable real-time moderation at scale. Longitudinal evaluation on temporally evolving data streams is also warranted, as language norms, slang, and coded expressions of hate shift continuously on dynamic platforms like Instagram and YouTube, and robustness to such drift is an important property for any deployed system.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgement

The authors would like to express their sincere gratitude to their institution, mentors, colleagues, and all those who directly or indirectly supported this research work. Their guidance, encouragement, and valuable suggestions greatly helped in completing this study successfully.

Reference

1. A. K. Gautam and A. Bansal, "A Review on Cyberstalking Detection Using Machine Learning Techniques: Current Trends and Future Direction," Mar. 01, 2022, *Seventh Sense Research Group*. doi: 10.14445/22315381/IJETT-V70I3P211.
2. V. S. Pimprale and M. Deore, "A Systematic Survey of AI-Generated Text Detection, Humanization, and Grammar Correction Techniques," 2026, *Seventh Sense Research Group*. doi: 10.14445/22315381/IJETT-V74I3P104.
3. A. Schmidt and M. Wiegand, "A Survey on Hate Speech Detection using Natural Language Processing," Association for Computational Linguistics. [Online]. Available: https://en.wikipedia.org/wiki/List_
4. S. Dowlagar and R. Mamidi, "A Survey of Recent Neural Network Models on Code-Mixed Indian Hate Speech Data," in *Forum for Information Retrieval Evaluation*, New York, NY, USA: ACM, Dec. 2021, pp. 67–74. doi: 10.1145/3503162.3503168.
5. D. Iftimie and E. Zinn, "Evaluating Simple Debiasing Techniques in RoBERTa-based Hate Speech Detection Models," Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.15430
6. R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking Aggression Identification in Social Media," 2018. [Online]. Available: https://competitions.codalab.org/
7. M. F. Mridha, M. A. H. Wadud, M. A. Hamid, M. M. Monowar, M. Abdullah-Al-Wadud, and A. Alamri, "L-Boost: Identifying Offensive Texts from Social Media Post in Bengali," *IEEE Access*, vol. 9, pp. 164681–164699, 2021, doi: 10.1109/ACCESS.2021.3134154.
8. F. zahra El-Alami, S. Ouatik El Alaoui, and N. En Nahnahi, "A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 6048–6056, Sep. 2022, doi: 10.1016/j.jksuci.2021.07.013.
9. A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, and S. H. Malik, "Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques," *International Journal of Information Management Data Insights*, vol. 2, no. 2, Nov. 2022, doi: 10.1016/j.jjimei.2022.100120.
10. B. R. Chakravarthi *et al.*, "Detecting abusive comments at a fine-grained level in a low-resource language," *Natural Language Processing Journal*, vol. 3, p. 100006, Jun. 2023, doi: 10.1016/j.nlp.2023.100006.
11. A. Sharma, A. Kabra, and M. Jain, "Ceasing hate with MoH: Hate Speech Detection in Hindi–English code-switched language," *Inf. Process. Manag.*, vol. 59, no. 1, Jan. 2022, doi: 10.1016/j.ipm.2021.102760.
12. H. Madhu, S. Satapara, S. Modha, T. Mandl, and P. Majumder, "Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments," *Expert Syst. Appl.*, vol. 215, Apr. 2023, doi: 10.1016/j.eswa.2022.119342.
13. K. Sreelakshmi, B. Premjith, and K. P. Soman, "Detection of Hate Speech Text in Hindi-English Code-mixed Data," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 737–744. doi: 10.1016/j.procs.2020.04.080.
14. V. U. Maheswari and R. Priya, "Novel Approach to Offensive Language Detection on Social Media: Tree CNN Integration with Adversarial Bi-LSTM," *International Journal of Engineering Trends and Technology*, vol. 72, no. 7, pp. 187–197, Jul. 2024, doi: 10.14445/22315381/IJETT-V72I7P120.
15. B. A. Alanazi and C. T. Lin, "Severity Detection of Cyberbullying in Saudi-DialecTweets: A Machine-Learning Approach," *International Journal of Engineering Trends and Technology*, vol. 74, no. 3, pp. 54–74, 2026, doi: 10.14445/22315381/IJETT-V74I3P105.
16. R. Pushpavalli, P. Rengaramanujam, I. E. Berna, D. Suseela, B. M. Dilli, and R. S. Vignesh, "Detection of Cyber Threats on Social Media using Optimized Sentiment-Aware Deep Learning Models," *International Journal of Engineering Trends and Technology*, vol. 74, no. 3, pp. 336–352, 2026, doi: 10.14445/22315381/IJETT-V74I3P123.
17. I. I. Banu and V. Thiyagarajan, "Nonlinear Seagull Optimized Bidirectional Recurrent Network for English Hate Speech Detection in Online Social Network," *International Journal of Engineering Trends and Technology*, vol. 74, no. 1, pp. 25–38, 2026, doi: 10.14445/22315381/IJETT-V74I1P102.
18. I. I. Banu, V. Thiyagarajan, V. J. Arputharaj, and P. Thenmozhi, "Censored Regressive Canonical Optimized Convolutional Deep Belief Classifier For Hate Speech Detection in Online Social Network," *International Journal of*

Engineering Trends and Technology, vol. 74, no. 2, pp. 223–234, 2026, doi: 10.14445/22315381/IJETT-V74I2P116.
19. I. Anand Raj, R. Vidya, and A. Martin, "Optimizing Bullying News Detection on Social Media: An Ensemble Deep Learning Approach with Enhanced Optimization Algorithm," *International Journal of Engineering Trends and Technology*, vol. 74, no. 2, pp. 266–279, 2026, doi: 10.14445/22315381/IJETT-V74I2P119.
20. A. A. Almazroi, A. A. Shah, and F. Mohammed, "Social Media and Online Islamophobia: A Hate Behavior Detection Model," *International Journal of Engineering Trends and Technology*, vol. 71, no. 11, pp. 27–32, 2023, doi: 10.14445/22315381/IJETT-V71I11P203.
21. T. Mandl *et al.*, "Overview of the HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages under Creative Commons License Attribution 4.0 International (CC BY 4.0)," 2020, [Online]. Available: http://ceur-ws.org
22. D. Sharma, A. Singh, and V. K. Singh, "THAR- Targeted Hate Speech Against Religion: A high-quality Hindi-English code-mixed Dataset with the Application of Deep Learning Models for Automatic Detection," *ACM Transactions on Asian and Low-Resource Language Information Processing*, Mar. 2024, doi: 10.1145/3653017.
23. A. Chhabra and D. K. Vishwakarma, "A literature survey on multimodal and multilingual automatic hate speech identification," *Multimed. Syst.*, vol. 29, no. 3, pp. 1203–1230, Jun. 2023, doi: 10.1007/s00530-023-01051-8.
24. C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text Data Augmentation for Deep Learning," *J. Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00492-0.
25. S. Masud, M. A. Khan, Md. S. Akhtar, and T. Chakraborty, "Overview of the HASOC Subtrack at FIRE 2023: Identification of Tokens Contributing to Explicit Hate in English by Span Detection," Nov. 2023, [Online]. Available: http://arxiv.org/abs/2311.09834
26. T. Leburu-Dingalo, K. Johannes Ntwaagae, N. Peace Motlogelwa, E. Thuma, and M. Mudongo, "Application of XLM-RoBERTa for Multi-Class Classification of Conversational Hate Speech," 2022. [Online]. Available: https://hasocfire.github.io/hasoc/2019/call_for_participation.html
27. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Aug. 2019, [Online]. Available: http://arxiv.org/abs/1908.10084
28. J. Lu *et al.*, "Hate Speech Detection via Dual Contrastive Learning," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2787–2795, 2023, doi: 10.1109/TASLP.2023.3294715.
29. Q. Chen, R. Zhang, Y. Zheng, and Y. Mao, "Dual Contrastive Learning: Text Classification via Label-Aware Data Augmentation," Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.08702
30. A. Sharma and R. Bhalla, "Detecting Hate Speech for Hindi-English Code-Mix Text Data Using Dual Contrastive Learning," in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 35–43. doi: 10.1016/j.procs.2025.03.304.
31. A. Apicella, F. Isgrò, and R. Prevete, "Hidden classification layers: Enhancing linear separability between classes in neural networks layers," *Pattern Recognit. Lett.*, vol. 177, pp. 69–74, Jan. 2024, doi: 10.1016/j.patrec.2023.11.016.
32. X. Li, C. Lv, W. Wang, G. Li, L. Yang, and J. Yang, "Generalized Focal Loss: Towards Efficient Representation Learning for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–14, 2022, doi: 10.1109/TPAMI.2022.3180392.
33. S. Sathyanarayanan, "Confusion Matrix-Based Performance Evaluation Metrics," *African Journal of Biomedical Research*, pp. 4023–4031, Nov. 2024, doi: 10.53555/ajbr.v27i4s.4345.
34. F. Ramiandrisoa, "Multi-task Learning for Hate Speech and Aggression Detection," 2022. [Online]. Available: http://ceur-ws.org
35. A. Chopra, D. K. Sharma, A. Jha, and U. Ghosh, "A Framework for Online Hate Speech Detection on Code Mixed Hindi-English Text and Hindi Text in Devanagari," *ACM Transactions on Asian and Low-Resource Language Information Processing*, Oct. 2022, doi: 10.1145/3568673.
36. C. D. Putra and H. C. Wang, "Advanced BERT-CNN for Hate Speech Detection," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 239–246. doi: 10.1016/j.procs.2024.02.170.
37. T. Khosla, S. Singh, and A. Shrivastava, "Hinglish Hate: Leveraging Code-Mixed Embeddings for Offensive Language Detection on Indian Social Media," in *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval)*, Association for Computational Linguistics, 2023, pp. 1102–1112. doi: 10.18653/v1/2023.semeval-1.154.
38. P. Ranasinghe and M. Zampieri, "Multilingual Offensive Language Identification with Cross-lingual Embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2021, pp. 5838–5844. doi: 10.18653/v1/2021.emnlp-main.459.
39. P. Velankar, H. Patil, A. Gore, S. Salunke, and R. Joshi, "MuRIL and BERT-Based Hate Speech Detection for Hindi-English Code-Mixed Text," *arXiv preprint*, arXiv:2112.09301, Dec. 2021. [Online]. Available: https://arxiv.org/abs/2112.09301
40. A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzman, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 8440–8451. doi: 10.18653/v1/2020.acl-main.747