MULTILINGUAL NLP MODEL FOR AUTOMATED IDENTIFICATION OF CYBERBULLYING IN KANNADA - ENGLISH CODE-MIXED STUDENT CHATS

Project Reference No.: *49S_MCA_0234*

College : Reva University, Bengaluru
Branch : Department Of Master of Computer Applications
Guide : Dr. Lokesh C K
Students : Ms. Shwetha P
Ms. Kirthana R

Keywords:

Cyberbullying Detection, Kannada–English Multilingual Text, XLM-RoBERTa, RoBERTa-Base, DeBERTa-v3-Base.

Introduction:

Cyberbullying has slowly crept into the everyday digital interactions of students, hitting hardest on messaging apps like WhatsApp where conversations happen constantly and largely outside the view of teachers or parents. Smartphones and cheap internet have made it ridiculously easy for harmful messages to travel fast, leaving victims dealing with anxiety, emotional breakdowns, depression, and a slow retreat from social life. In a country like India, the detection challenge gets worse because students rarely type in just one language — they casually mix English with Kannada in the same sentence, toss in slang, abbreviations, emojis, and misspelled words that carry meaning only within their circle. Informal grammar and transliterated text make it nearly impossible for standard tools to figure out whether a message is actually harmful or just playful banter. This project presents a Multilingual Cyberbullying Detection System built to classify chat messages as bullying or non-bullying using a combination of classical machine learning and modern natural language processing methods. Both traditional models and transformer-based deep learning architectures were employed to push classification accuracy as high as possible. The entire pipeline covers everything from data cleaning and model training to privacy-preserving mechanisms, a prediction API, an interactive teacher dashboard, and database-backed storage — and it handles English, Kannada, and English–Kannada code-mixed messages without skipping a beat.

Objectives:

1. Prepare a custom dataset of Kannada–English code-mixed student chat messages and train classical machine learning along with transformer models to sort each message into cyberbullying or not cyberbullying through binary classification.

2. Fine-tune XLM-RoBERTa, RoBERTa-Base, and DeBERTa-v3-Base on real student conversations loaded with slang, emojis, and transliterated words that existing keyword-based tools completely miss.
3. Combine all trained models using ensemble strategies like majority voting, weighted soft voting, and stacking to strengthen detection reliability beyond what any single model manages alone.
4. Develop a FastAPI-driven dashboard that lets teachers upload full WhatsApp group chat exports at once, scan every message through binary classification, review flagged content with severity scores, and take action through instant alerts without needing any technical skills.

Methodology:

The dataset was sourced and compiled manually across eight individual CSV files, bringing the total count to 59,000 records, spread across English, Kannada, and the messy code-mixed stuff students genuinely send on WhatsApp. File sizes ranged from 2,000 entries on the smaller end to 10,000 on the larger Kannada and mixed-language ones. Each message carried a straightforward binary label: cyberbullying or not cyberbullying. Raw text got cleaned up before anything touched a model — URLs yanked out, special characters dropped, everything lowercased, emojis replaced with word equivalents, and student shorthand expanded into proper form. TF-IDF with unigrams and bigrams handled feature extraction for the classical side, while transformer models broke input down through their built-in subword tokenizers, which is what lets them chew through misspelled and code-switched words without choking.

Logistic Regression (C=10, L2, lbfgs), SVM (linear kernel, Platt-calibrated probabilities), and Random Forest went through GridSearchCV with 5-fold cross-validation on identical TF-IDF vectors so no representation difference could muddy the comparison. RoBERTa-Base, DeBERTa-v3-Base, and XLM-RoBERTa were each fine-tuned for 2 epochs — AdamW optimizer, weight decay at 0.01, learning rates between 3e-5 and 5e-5, batch size 32, sequence cap of 48 tokens, FP16 mixed precision running on a Tesla T4 through PyTorch and Hugging Face. Predictions from all six models fed into three ensemble setups: Majority Voting, Weighted Soft Voting scaled by validation F1, and Stacking with a Logistic Regression meta-learner trained on 12 probability features. Sitting alongside these was an adaptive JSON-based message store — cosine similarity above 0.85 between an incoming message and a stored one triggered a 70-30 blend of the saved label with the fresh ensemble output. KNN-based severity scoring then rated every flagged message from 0 to 10 using ensemble confidence, cross-model agreement, and proximity to known harmful entries, pushing anything past 7 straight into a critical alert. Data was split 70-15-15, giving 37,797 training samples, 8,099 for validation, and 8,100 for testing. Everything ran through a FastAPI backend hooked to SQLite, with teachers accessing results on a browser dashboard.

Result and Conclusion: Random Forest and SVM achieved perfect 100% accuracy, precision, recall, and F1-score on 8,100 test messages. Logistic Regression got 99.99%. XLM-RoBERTa scored 99.99% from multilingual Kannada background, RoBERTa-Base reached 99.95%, and DeBERTa-v3-Base got 99.94%. Majority Voting, Weighted Soft Voting, and Stacking each hit 100%. The adaptive KNN store learns from educator input to find new slang and repeated bullying. The framework processes code-mixed Kannada-English text via FastAPI dashboard, sending instant notifications to teachers managing student group chats.

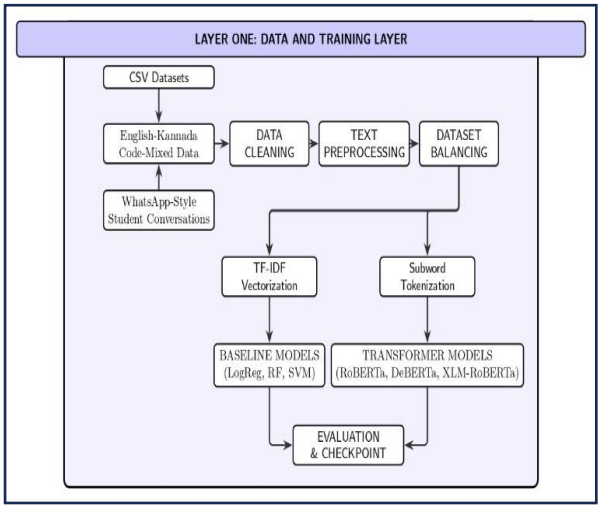

Figure 1: architecture overview

Figure 2: overview of the dashboard

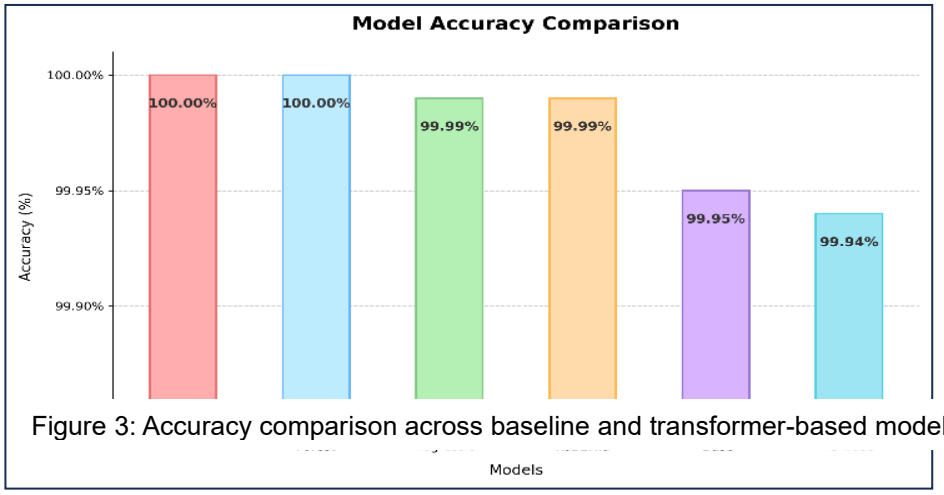

Figure 3: Accuracy comparison across baseline and transformer-based models.

Project Outcome & Industry Relevance:

Bringing together traditional classifiers and transformers under one roof turned out to be a reliable way to spot cyberbullying buried in the kind of mixed Kannada-English texts students actually type. Schools across Karnataka can now let the system scan group chats instead of asking teachers to sift through thousands of messages by hand. The adaptive store keeps things fresh — every time a teacher fixes a wrong call, the tool picks up that correction and gets sharper with new slang and insults it has never seen before. Outside classrooms, the same setup fits content moderation at regional startups, internal chat screening in HR departments, and child safety groups watching

over online spaces. Severity scores tell administrators not just what is harmful but how urgent it is, so the worst cases get attention first. On the research side, this work directly addresses a blind spot most NLP studies have skipped — handling real Indian code-mixed conversations instead of pretending everything arrives in neat English.

Project Outcomes and Learnings:

A Multilingual Cyberbullying Detection System was built using six models — RoBERTa-Base, DeBERTa-v3-Base, XLM-RoBERTa, SVM, Logistic Regression, and Random Forest — trained on 53,497 Kannada-English code-mixed student chat messages. The full setup runs on a FastAPI backend, an SQLite database, and a Streamlit dashboard that lets teachers upload WhatsApp chats, check messages on the spot, and get alerts based on severity. Every message receives a straight result — Cyberbullying or Not — along with a confidence score, a label such as insult or threat. On the 8,025-message test set, all models crossed 99.97% accuracy, with RoBERTa-Base and SVM hitting a perfect 100% on every metric, proving strong reliability for real-world multilingual detection.

Future Scope

1. Language stretch – Right now, the dataset handles Kannada-English only. Down the line, mixing in Tamil, Hindi, Telugu, and other regional tongues would make the whole thing useful far beyond Karnataka.
2. Sarcasm spotting – Some kids type stuff that reads nice but stings badly. A separate sarcasm-checking piece could catch that kind of hidden cruelty.
3. Picture and meme checks – Bullying doesn't always happen through typed words. Edited photos, memes, and screenshots need scanning too, which calls for an image analysis add-on.
4. Plugging into live chats – Nobody wants to upload messages by hand forever. Tying the tool straight into WhatsApp Business API or any school chat app fixes that problem fast.
5. Showing the reason behind flags – Teachers shouldn't have to guess why something got flagged. Something like Ollama could spit out a quick, readable reason pointing at the exact trigger words.
6. Emergency alerts board – Not every flagged message is equally bad. A separate panel highlighting the nastiest cases helps school heads act on those first without digging through everything else.