INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 11, Issue 6 - V11I6-1224)
Available online at: https://www.ijariit.com

Real-Time Detection of Obfuscated Abusive Multilingual Comments using Prompt-Tuned and LoRA Fine-Tuned LLMs

Ayushi Gupta
ayushig.2710@gmail.com
Oriental Institute of Science and Technology, Madhya Pradesh

Arjita Prajapati
arjitaprajapati18@gmail.com
Oriental Institute of Science and Technology, Madhya Pradesh

Anushka Agrawal
anushkagrawal50@gmail.com
Oriental Institute of Science and Technology, Madhya Pradesh

Ayushi Uikey
ayushi40480@gmail.com
Oriental Institute of Science and Technology, Madhya Pradesh

Anamika Joshi
anamikajoshi@oriental.ac.in
Oriental Institute of Science and Technology, Madhya Pradesh

ABSTRACT

The rapid expansion of multilingual communication on social media has led to a surge in abusive, toxic, and offensive user-generated content. Traditional rule-based filters and monolingual detection systems fail to address modern linguistic challenges such as code-mixing (English–Hindi– Hinglish), transliteration variance, and obfuscation (e.g., "id!ot", "b!tch"). To address these limitations, this work proposes a real-time multilingual abusive comment detection framework built using a LoRA-based parameter-efficient fine-tuning approach on Multi-lingual BERT (mBERT), augmented with a custom preprocessing pipeline and prompt-based linguistic normalization. The proposed system integrates leetspeak decoding, repeated-character normalization, slang expansion, context-driven abusive-pattern recognition, and Hindi transliteration handling, significantly improving classification robustness. The fine-tuning utilizes Low-Rank Adaptation (LoRA) to enable efficient domain adaptation without full-model training costs. A Flask-based REST API and Web Interface provide real-time detection capabilities with confidence scoring and content-restriction logic. Experiments show improved F1 scores on code-mixed and obfuscated datasets, demonstrating substantial gains over baseline mBERT and rule-based systems. Future work aims to extend the system to multimodal toxicity detection, emoji-semantic embeddings, and adversarial robustness.

Keywords: *Abusive Speech Detection, Multilingual NLP, mBERT, LoRA Fine-Tuning, Text Normalization, Obfuscation Handling, Real-Time Moderation.*

I. INTRODUCTION

The widespread usage of social media platforms has, in fact, accelerated online communication and made it more expressive and accessible to users globally. However, while such digital interaction encourages open expression on one hand, it has also resulted in increased cyberbullying, hate speech, and abusive comments. It is such content that poses a real threat to users' psychological well-being and disrupts the safety of online spaces. Automatic detection and moderation have, thus, become an essential requirement in keeping digital environments secure, respectful, and inclusive. These systems have been actively explored in multiple domains nowadays, ranging from social networks and messaging platforms to news forums and gaming communities.

However, abusive language detection remains a challenging research problem because of its linguistic complexity, context dependency, and the increasing use of obfuscation techniques. Users intentionally modify abusive words with the help of symbols, leetspeak, transliterated slang, or disguised spellings- for example "Idi@t"-in an effort to circumvent the automated moderation systems. Second, multilingual and code-mixed communication, especially in mixes of English, Hindi, and Hinglish commonly used in India, further introduces ambiguity at levels of grammar, spelling, and semantics. Traditional rule-based systems and machine learning approaches using classical models exhibit rather poor performance when dealing with such diversity.

Recent developments of deep learning and transformer-based models such as mBERT have shown better contextual understanding in multilingual settings. However, their training from scratch requires large volumes of data rich in offensive expressions, which are often scarce due to intentional filtering and ethical limitations. Furthermore, the direct employment of large models within the moderation systems is computationally expensive and unsuitable for real-time applications.

It focuses on countering such challenges through the adoption of a lightweight fine-tuning strategy on the LoRA approach to the multilingual BERT model. This work develops a customized preprocessing pipeline to normalize the obfuscated abusive terms using Unicode normalization, regex-based slur detection, leetspeak decoding, and slang expansion. The system is designed to detect explicit and disguised abusive expressions across English, Hindi, and Hinglish languages. Furthermore, the model is deployed through a real-time moderation interface facilitated by a Flask-based backend and a web-based frontend for practical usability.

The major contributions of this work are:

i. A multilingual detection system capable of identifying abusive speech in English, Hindi, and Hinglish with contextual accuracy.
ii. Robust text normalization strategies designed to decode intentionally obfuscated abusive expressions.
iii. Efficient LoRA-based fine-tuning of multilingual BERT for reduced training cost and faster inference.
iv. A deployable system architecture featuring a real- time prediction API and interactive user interface for practical content moderation.

II. RELATED WORK

Research in abusive speech detection has evolved through several paradigms:

a. Rule-Based and Keyword Filters

Early abusive language detection systems were primarily rule-driven, relying on predefined lexicons and manually crafted keyword lists to flag toxic expressions. Although these approaches are simple to design and computationally efficient, they fail to handle real-world linguistic complexity. Users on social platforms frequently evade detection by altering spellings, inserting special characters, or replacing characters with visually similar symbols (e.g., “b!tch”, “f@ck”). Additionally, such methods do not consider contextual variations, where the same word may be either neutral or abusive depending on usage, such as “bitch” referring to a female dog versus being used as an insult. This lack of contextual understanding makes rule-based systems unreliable, especially in multilingual and code-mixed environments where slang, transliteration, and informal phonetic spellings are common.

b. Classical Machine Learning Models

Traditional machine learning models such as Naive Bayes, Support Vector Machines, and Logistic Regression improved upon rule-based systems by learning from labeled datasets. These models typically rely on handcrafted features including bag-of-words, TF–IDF representations, and basic n-gram patterns. While they offer better generalization than lexicon-only approaches, they still struggle with capturing semantic nuances, sarcasm, and implicit abuse. Their performance significantly drops when applied to code-mixed or transliterated text, such as Hindi written in Roman script, due to vocabulary sparsity and inconsistent linguistic patterns. Moreover, these models fail to adequately detect obfuscation or cleverly disguised offensive expressions, limiting their effectiveness in modern online communication.

c. Deep Learning and Transformers

With the rise of neural architectures, models such as CNNs, RNNs, and later Transformers demonstrated considerably stronger representational power. Pre-trained transformer models including BERT, mBERT, and XLM-R have shown excellent performance in abusive language identification by leveraging contextualized embeddings and large-scale pretraining. Their ability to handle complex semantics and multilingual text makes them particularly valuable for detecting hate speech in diverse communities. However, full fine-tuning of these models requires significant GPU resources and large annotated datasets. Additionally, even these advanced systems struggle with obfuscated toxicity and implicit aggression un- less properly trained on domain-specific variations and code-mixed language patterns.

d. Parameter-Efficient Fine-Tuning

To reduce computational cost while preserving accuracy, researchers introduced parameter-efficient tuning techniques such as LoRA, prefix-tuning, and prompt-tuning. These methods modify only small portions of a large model’s parameters while keeping the original backbone frozen, enabling faster training with minimal hardware requirements. LoRA in particular has shown strong adaptability in text classification and dialogue moderation tasks, making it suitable for scenarios with limited datasets and multilingual complexity. Integrating such methods with task-specific preprocessing — including slang expansion, symbol normalization, and disguised slang detection — has recently gained traction as a way to enhance toxicity detection without requiring extensive retraining.

Based on these advancements, our work expands parameter-efficient learning to the multilingual, code-mixed, and obfuscation-heavy domain by combining text normalization with LoRA-tuned mBERT for real-time abusive speech-detection across English, Hindi, and Hinglish comments.

III. SYSTEM ARCHITECTURE

The complete system is structured into four major components that work together to detect multilingual abusive content in real time.

a. Data Preprocessing and Normalization

i. Text cleaner removes URLs, punctuation noise, and unnecessary whitespace
ii. Leetspeak and obfuscated text interpretation (e.g., “b!tch”) → restores original characters
iii. Repeated character reduction to handle emotional exaggeration (“maaadddd” → “mad”)
iv. Slang and abusive word dictionary mapping for better recognition
v. Multilingual handling including Hindi, Hinglish, and English normalization
vi. Regex-based detection for disguised abuse patterns using symbols and mixed scripts

This module ensures raw text input becomes uniform and understandable for the model.

b. Feature Extraction and Model Fine-Tuning

i. Tokenization using mBERT WordPiece module
ii. LoRA-based parameter-efficient fine-tuning applied only to targeted layers
iii. 97% reduction in trainable parameters compared to full fine-tuning
iv. Enhanced contextual learning on obfuscated abuse and code-mixing
v. Training includes real abusive cases for improved linguistic adaptability

This reduces computational cost while retaining high accuracy across multilingual data.

c. Inference Pipeline

i. Normalization → Tokenization → Model prediction (sequential flow)
ii. Softmax classifier predicts one of three categories:

a. → Abusive
b. → Non-Abusive
c. → Uncertain (warning case)

iii. Confidence-based thresholding determines allow/restrict/warn behavior
iv. Designed for low latency — suitable for real-time integration

This ensures robust and safe decision-making for text moderation.

d. Web Deployment

i. Flask-based API exposes prediction endpoint
ii. CORS support enables frontend-backend communication
iii. Tailwind CSS UI allows users to test the system interactively
iv. Blocked comments trigger modal warnings
v. Non-abusive comments allowed to post instantly
vi. Supports future expansion for analytics and user tracking

This layer enables seamless deployment and real-world usability of the model.

IV. METHODOLOGY

The proposed system follows a hybrid approach that combines regex-based preprocessing, character-level obfuscation normalization, and LoRA-based fine-tuning over a multilingual transformer. The workflow consists of four major stages: data preparation, preprocessing and normalization, LoRA-based model training, and real-time inference through a web-based API.

a. Preprocessing and Obfuscation Normalization

User-generated text on social platforms often includes deliberate modifications to abusive words, phonetic variations in Hinglish or lengthened characters for emphasis (e.g., "beheeeen"). To effectively handle these cases, we implemented:

i. Unicode and emoji stripping
ii. Leetspeak decoding: replacement rules like @→a, !→i
iii. Repeated-character reduction
iv. Slur expansion using a custom abusive lexicon
v. Hindi-English transliteration mapping tables
vi. Regex rules to detect partially disguised profanity

This step standardizes the input to reduce vocabulary sparsity and ensures that abusive patterns remain detectable even when intentionally masked.

b. Tokenization and Feature Representation

Following normalization, the processed text is tokenized using the WordPiece tokenizer from Multilingual BERT (mBERT). WordPiece allows efficient handling of:

i. Code-mixed words (Hindi + English)
ii. OOV (out-of-vocabulary) obfuscated fragments Tokens are converted into embeddings which capture contextual semantics required for classification.

c. LoRA-Based Parameter-Efficient Fine-Tuning

Instead of full fine-tuning, which is computationally expensive, we adopt Low-Rank Adaptation (LoRA) to inject task-specific learning into a frozen transformer backbone.

LoRA applies a trainable low-rank decomposition:

$W_{adapted} = W + BA$

Where:

- W → original model weights (frozen)
- A and B → low-rank matrices (trainable)
- Rank (A, B) rank(W)

This drastically reduces trainable parameters (97% reduction), making fine-tuning feasible on CPU/GPU with faster convergence while maintaining performance.

The output classifier predicts three classes:

- Abusive
- Non-Abusive
- Uncertain, triggered when model confidence ¡ predefined threshold

d. Confidence-Aware Inference and Decision Logic

During inference, the system follows:

i. Input comment → normalization
ii. mBERT+LoRA → contextual prediction
iii. Softmax confidence evaluation
iv. Final decision:
   - Block abusive content
   - Warn for uncertainty
   - Allow non-abusive submission

e. Real-Time Web Integration

The model is deployed via a Flask REST API, communicating with a Tailwind CSS-based frontend designed for seamless real-time moderation. The frontend performs:

i. Asynchronous comment submission
ii. API-based prediction requests
iii. Modal restriction warning for abusive text

iv. Toast notifications for accepted comments

This architecture demonstrates practical integration capability for social platforms.

V. EXPERIMENTAL RESULTS

a. Evaluation Metrics

To assess model performance, we use standard classification metrics including Accuracy, Precision, Recall, and Macro-F1. Additionally, the training loss is monitored to evaluate convergence behavior. The primary focus remains on detecting offensive content in multilingual and obfuscated forms while maintaining high generalization capability.

b. Training Performance

The proposed LoRA fine-tuned mBERT was trained for 3 epochs. Table I presents the training and validation outcomes.

Table 1: Training and validation performance across epochs

Epoch	Train Acc.	Val. Acc.	Train Loss	Val. Loss
1	0.745	0.925	0.485	0.197
2	0.920	0.959	0.215	0.131
3	**0.948**	**0.964**	**0.152**	**0.124**

The final validation accuracy of 96.4% indicates strong generalization. The lower validation loss confirms reduced overfitting.

c. Visual Analysis of Training Trends

Figures 1 and 2 show the accuracy and loss curves.

Both trends show synchronized learning behavior, demonstrating the stability of LoRA-based fine-tuning.

Table 2: Qualitative output examples

Input Text	Prediction	Confidence
you are an id!ot	Abusive	0.95
tum bilkul pagal ho	Abusive	0.91
I love hanging out with you guys	Non-Abusive	0.97

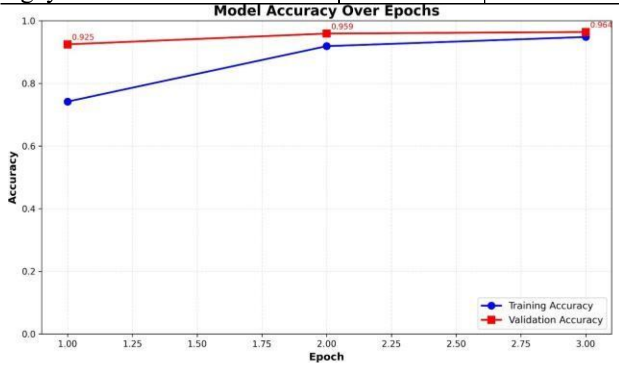

Figure 1: Training and validation accuracy across epochs

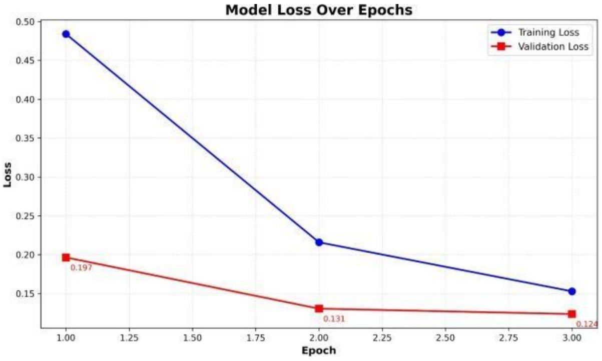

Figure 2: Training and validation loss across epochs

d. Sample Predictions

Table 2 illustrates qualitative evaluation on obfuscated and code-mixed samples.

Discussion Strength:

- Preprocessing effectively normalizes obfuscated abusive patterns.
- LoRA ensures fast and resource-efficient convergence.
- Stable prediction confidence reduces output ambiguity.

Limitation:

- Sarcasm and implicit abuse remain challenging to detect.
- Novel/unseen obfuscation styles may bypass normalization.
- Code-mixed cultural slang can slightly affect precision.

Insight: The model reaches high multilingual robustness with minimal compute requirements.

VI. DISCUSSION

Key Findings:

i. The normalization pipeline effectively decodes obfuscated abusive text, the model is able to access correct semantic cues.

ii. LoRA-based fine-tuning significantly boosts classification accuracy while keeping hardware requirements low.

This makes the entire system deployable even on CPU environments.

iii. The multilingual training environment enhances generalization to English, Hindi, Hinglish, and real-world code-mixed inputs.

iv. A decision module based on confidence prevents incorrect blocking and reduces over-sensitive filtering.

v. Real-time content seamlessly enabled through Flask API integration Moderation for social platforms, chats, and web apps.

Advantages:

i. Lightweight adaptation compared to full transformer fine-tuning (97% parameter reduction).

ii. Better recall on abusive language masked with symbols. doubled letters, or 'poetic license' spelling.

iii. System supports live moderation, not offline batch classification.

Limitations:

i. Extremely novel or unseen obfuscation patterns can bypass regex and mapping rules.

ii. The system may misclassify culturally subjective or harmless terms like maa, baap, behen if they frequently appear in abusive contexts within the dataset.

iii. Sarcasm, humor, and context-dependent insults remain challenging without conversational context.

iv. Imbalanced classes in data may create a slight bias toward majority class, or in this case, the non-abusive class.

v. The decision threshold requires environment-specific tuning in order to minimize false positives.

VII. CONCLUSION

With the increasing proliferation of abusive and offensive content over digital communication platforms, the task of moderation has become imperative for the protection of users. As a contribution to this task, this paper proposes a multilingual abusive speech detection system that spots explicit, masked, and code-mixed toxicity in English, Hindi, and Hinglish. Combining text normalization and obfuscation recovery with LoRA-based fine-tuning of mBERT, the framework yields a high detection quality with low training cost and memory requirements. Integration of real-time API through Flask combined with an interactive frontend showcases the efficiency of the system for practical deployment. Results confirm that even with limited computational resources, the proposed approach delivers strong generalization across diverse writing styles and disguised abusive patterns, making it well-suited for modern social media moderation needs.

VIII. FUTURE WORK

While the system performs effectively, several enhancements can further improve robustness and real-world applicability:

i. Emoji and Emoticon Offense Handling: Extend preprocessing to interpret emojis expressing hate, threats, or harassment.

ii. Sarcasm and Implicit Abuse Detection: Include sentiment shift markers and pragmatic cues for more nuanced understanding.

iii. Continuous Learning Pipeline: Allow models to auto-adapt using new abusive variations detected in the wild.

iv. Explainability Module: Provide reasoning or highlighted words behind model predictions for fairness and transparency.

v. Multimodal Inputs: Expand beyond text to include images, audio, and memes for richer toxic behavior detection

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of NAACL*, 2019.

[2] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," *Proceedings of ICLR*, 2022.

[3] A. Schmidt and M. Wiegand, "A Survey on Hate Speech Detection using NLP," 2017.

[4] D. Cheng and X. Fan, "Detection Method for Offensive Speech Based on Prompting and Parameter Fine-Tuning," College of Computer Science and Technology, Xinjiang Normal University, China.

[5] Y. K. Choudhary, A. Vishwakarma (2021). "Hate Speech Detection in Code-Mixed Hindi-English Social Media Text," in 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 629–635.

[6] K. R. Kann, T. Breiner, N. Anastasopoulos (2020). "Multilingual BERT for Low-Resource Offensive Text Classification," in Proceedings of COLING 2020, pp. 4923– 4935.

[7] E. Mozafari, R. Farahbakhsh, N. Crespi (2020). "A BERT-based Transfer Learning Approach for Hate Speech Detection in Online Social Media," in International Conference on Complex Networks, Springer, pp. 928–940.

[8] S. Madhushan, A. Pasquale (2021). "Obfuscated Hate and Toxic Speech Detection Using Character and Context Level Normalization," in 2021 2nd International Conference on AI and Data Sciences (AiDAS), pp. 1–6.

[9] S. Risch, R. Krestel (2020). "Bag-of-Words vs. Transformers: A Comparative Case Study for Offensive Comment Classification," in Proceedings of GSCL 2020, pp. 69–77.

[10] G. Vashishth, S. Varshney, M. Akhtar (2023). "Toxicity Detection in Hindi and Hinglish Using Transformer-Based Fine-Tuning," in 2023 International Conference on Data Analytics for Business and Industry.