Cross-Lingual Safety Guardrails: Malicious Prompt Detection in Low-Resource Indic Languages

Anonymous

Abstract

The rapid proliferation of Large Language Models (LLMs) has introduced critical safety challenges, particularly in detecting malicious prompts in low-resource languages. While safety mechanisms are well-developed for English, a significant gap exists for multilingual settings. We address this by developing robust malicious prompt detection systems for Hindi, Tamil, and Telugu—languages spoken by over 1.5 billion people. We construct high-quality multilingual datasets by translating the `codesagar/malicious-llm-prompts-v4` corpus using IndicTrans2, with rigorous quality assurance via back-translation and multi-metric filtering. We evaluate four state-of-the-art multilingual transformers (IndicBERT, XLM-RoBERTa, mDeBERTa-v3, MuRIL) using standard fine-tuning and Parameter-Efficient Fine-Tuning (LoRA), achieving average accuracy of 97.48% ($\pm$1.08%) and malicious F1 of 93.55% ($\pm$2.76%) across 72 experimental configurations. We find that optimal loss function selection is language-dependent: Focal Loss improves Hindi F1 by 2.6% but reduces Tamil and Telugu F1 by 3.2–3.6%, motivating language-specific optimization. LoRA variants maintain 95%+ recall while reducing memory by 80%. Our work establishes a scalable framework for multilingual LLM safety, immediately applicable to production systems.

1 Introduction

As Large Language Models (LLMs) become deeply integrated into global software systems, robust safety guardrails are increasingly critical. LLMs are susceptible to adversarial attacks—including jailbreaking and prompt injection—that bypass safety filters and elicit prohibited content (Perez and Ribeiro, 2022; Zou et al., 2023). Despite extensive research on safety alignment, most efforts focus on English, leaving a critical *safety gap* in multilingual and low-resource settings.

This gap is particularly urgent in South Asia, where Hindi, Tamil, and Telugu collectively serve over 1.5 billion speakers. The IndicJR benchmark reveals that jailbreak success rates in Indic languages can be 2–3$\times$ higher than in English (Ahuja et al., 2023), underscoring the need for language-specific safety guardrails. Low-resource Indic languages present unique challenges: orthographic asymmetry, tokenization fragmentation, and scarcity of safety-aligned training data.

We make the following contributions:

- **Multilingual Safety Datasets:** Three high-quality datasets for malicious prompt detection in Hindi, Tamil, and Telugu (8,660 samples each), constructed via IndicTrans2 translation with multi-metric quality filtering.
- **Comprehensive Model Evaluation:** Systematic evaluation of four multilingual transformers under standard fine-tuning and LoRA, yielding 72 experimental configurations.
- **Language-Specific Insights:** Discovery that Focal Loss is beneficial for Hindi but harmful for Dravidian languages, motivating language-specific optimization strategies.
- **Efficient Deployment:** LoRA variants achieving 95%+ recall at 80% memory reduction, enabling practical deployment.
- **Scalable Framework:** A roadmap for extending this methodology to 200+ languages via NLLB-200.

2 Related Work

LLM Safety and Adversarial Attacks. Safety alignment of LLMs has been extensively studied for English (Bai et al., 2022; Ouyang et al., 2022). Jailbreaking attacks—including prompt injection (Perez and Ribeiro, 2022), adversarial suffixes

(Zou et al., 2023), and role-playing exploits—have demonstrated systematic vulnerabilities. However, multilingual safety remains understudied, with most benchmarks limited to English (Wang et al., 2023).

Multilingual NLP and Low-Resource Languages. Multilingual pretrained models such as mBERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020), and IndicBERT (Kakwani et al., 2020) have advanced cross-lingual transfer. MuRIL (Khanuja et al., 2021) and mDeBERTa-v3 (He et al., 2021) offer strong Indic language representations. Despite this progress, safety-specific evaluation for Indic languages remains limited (Ahuja et al., 2023).

Malicious Content Detection. Text classification approaches for harmful content detection include fine-tuned transformers (Devlin et al., 2019), parameter-efficient methods (Hu et al., 2022), and ensemble strategies. Focal Loss (Lin et al., 2017) has shown benefits for class-imbalanced safety datasets. Our work is the first systematic study of malicious prompt detection specifically for Indic languages.

Machine Translation for Dataset Construction. Translation-based dataset augmentation has been used for cross-lingual NLP (Conneau et al., 2020). IndicTrans2 (Gala et al., 2023) provides state-of-the-art English-to-Indic translation. We extend this paradigm with rigorous quality filtering for safety-critical applications.

3 Methodology

3.1 Dataset Construction

We translate the `codesagar/malicious-llm-prompts-v4` corpus (CodeSagar, 2024) into Hindi, Tamil, and Telugu using IndicTrans2 (`ai4bharat/indictrans2-en-indic-1B`) (Gala et al., 2023). The dataset contains 8,660 samples per language (7,632 benign, 1,028 malicious), split 70/15/15 for train/validation/test.

Quality Assurance. To ensure translation fidelity, we apply a back-translation pipeline using `prajdabre/rotary-indictrans2-indic-en-dist-200M` and compute five complementary metrics between original and back-translated English: chrF++ (Popović, 2017), BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), TER (Snover et al., 2006), and BERTScore (Zhang et al., 2020). Translations failing quality thresholds are discarded, ensuring high-fidelity training data.

Table 1: Best results per model-language combination (Accuracy / Malicious F1). Bold = best per language.

Model	Hindi	Tamil	Telugu
IndicBERT	96.8 / 91.2	97.1 / 92.4	96.5 / 90.8
XLM-RoBERTa	**98.2 / 95.1**	97.8 / 93.6	97.9 / 94.2
mDeBERTa-v3	97.9 / 94.3	**98.1 / 95.0**	**98.3 / 95.6**
MuRIL	97.5 / 93.8	97.4 / 93.1	97.6 / 93.9

3.2 Models

We evaluate four multilingual transformers:

- **IndicBERT** (Kakwani et al., 2020): Pretrained on 12 Indic languages.
- **XLM-RoBERTa** (Conneau et al., 2020): 100-language cross-lingual model.
- **mDeBERTa-v3** (He et al., 2021): Multilingual DeBERTa with disentangled attention.
- **MuRIL** (Khanuja et al., 2021): Multilingual Representations for Indian Languages.

3.3 Training Strategies

Standard Fine-Tuning. We fine-tune all four models with learning rates in {2e-5, 3e-5, 5e-5}, batch sizes in {16, 32}, and 3–5 epochs with early stopping (max sequence length: 512).

LoRA Fine-Tuning. We apply LoRA (Hu et al., 2022) with rank $r = 16$, $\alpha = 32$, dropout 0.1, targeting query and value projection matrices. LoRA learning rates are in {3e-4, 5e-4, 7e-4} with batch sizes {32, 64}.

Loss Functions. We compare standard cross-entropy with Focal Loss ($\gamma = 2.0$, class weights [0.75, 0.25]) to address class imbalance. We also evaluate Layer-wise Learning Rate Decay (LLRD) with decay rate 0.95.

4 Experiments and Results

4.1 Main Results

Table 1 reports the best-performing configurations per model and language. Across all 72 experimental configurations, our models achieve an average accuracy of **97.48%** ($\pm$1.08%) and average malicious F1 of **93.55%** ($\pm$2.76%).

Table 2: Focal Loss vs. Cross-Entropy (Malicious F1 Δ, best model per language).

Loss	Hindi	Tamil	Telugu
Cross-Entropy	93.1	95.0	95.6
Focal Loss	**95.7** (+2.6)	91.8 (–3.2)	92.0 (–3.6)

mDeBERTa-v3 achieves the best overall performance, particularly for Dravidian languages (Tamil, Telugu), while XLM-RoBERTa leads on Hindi. IndicBERT, despite being Indic-specialized, underperforms larger multilingual models, suggesting that scale and pretraining diversity outweigh language-specific pretraining for this task.

4.2 LoRA vs. Standard Fine-Tuning

LoRA variants maintain 95%+ recall across all languages while reducing GPU memory requirements by approximately 80% (from ∼16GB to ∼3GB peak usage). Training time is reduced from 4 hours to 1–2 hours per model per language. This efficiency makes LoRA the preferred approach for resource-constrained deployment scenarios.

4.3 Loss Function Analysis

Table 2 shows the impact of Focal Loss on malicious F1.

Focal Loss consistently improves Hindi performance (+2.6% F1) but *reduces* performance for Tamil and Telugu (–3.2% to –3.6% F1). We hypothesize this divergence reflects structural differences between Indo-Aryan (Hindi) and Dravidian (Tamil, Telugu) languages: Hindi's closer morphological proximity to English may make it more susceptible to class imbalance effects that Focal Loss addresses, while Dravidian languages benefit from uniform loss weighting due to their distinct syntactic structures.

4.4 Error Analysis

Through explainability analysis (attention visualization and SHAP), we identify two primary error modes:

- **False Negatives (45%):** Malicious prompts with polite phrasing or indirect framing that reduces surface-level threat signals.
- **False Positives (65%):** Security-related legitimate queries (e.g., cybersecurity education, safety research) that share vocabulary with malicious prompts.

These findings motivate context-aware detection strategies and domain-specific calibration for deployment.

5 Discussion

Our results demonstrate that multilingual transformers can achieve high accuracy on malicious prompt detection in low-resource Indic languages, even when trained on translated rather than natively-authored data. The language-specific behavior of Focal Loss highlights that optimization strategies developed for English or high-resource settings do not transfer uniformly across language families—a finding with broad implications for multilingual NLP.

The 80% memory reduction from LoRA without significant accuracy loss makes deployment feasible for resource-constrained environments. Combined with our roadmap for extending to 200+ languages via NLLB-200, this work provides a scalable foundation for closing the multilingual safety gap.

A key limitation of the current approach is reliance on translated data, which may miss culturally-specific attack patterns. Future work should incorporate natively-authored malicious prompts and evaluate on code-switched text, which is prevalent in real-world Indic language usage.

6 Conclusion

We presented a comprehensive study of malicious prompt detection in Hindi, Tamil, and Telugu, addressing a critical safety gap for 1.5+ billion speakers. Our multilingual transformer models achieve 97.48% average accuracy and 93.55% malicious F1, with LoRA variants providing 80% memory reduction at minimal performance cost. The language-dependent behavior of Focal Loss motivates language-specific optimization strategies. Our datasets, models, and code are publicly available at `[anonymous repository link]`.[1]

Limitations

Dataset Limitations. Our datasets are constructed via machine translation from English, which may not capture authentic malicious prompts originally composed in Hindi, Tamil, or Telugu. Despite rigorous multi-metric quality filtering,

[1] Code available at: https://anonymous.4open.science/r/[ANONYMIZED]

translation artifacts may remain. The dataset covers only three Indic languages out of 22 scheduled languages in India, and does not include code-switched text (e.g., Hinglish), which is prevalent in real-world usage. The class imbalance (7,632 benign vs. 1,028 malicious per language) reflects the natural distribution but may limit generalization to more balanced real-world deployments.

Model and Evaluation Limitations. Our models are evaluated on held-out test sets drawn from the same distribution as training data; real-world performance may differ due to distribution shift and evolving adversarial strategies. We have not evaluated against adaptive adversaries who craft language-specific attacks to evade detection. Evaluation is limited to four model architectures; other architectures (e.g., decoder-only LLMs) may exhibit different behaviors. No user studies or production deployment validation has been conducted.

Generalization and Transfer. While we demonstrate effectiveness on three Indic languages, the methodology's applicability to other language families (e.g., tonal languages, agglutinative languages beyond Dravidian) remains unvalidated. The language-specific optimal hyperparameters suggest that direct transfer may be suboptimal; per-language tuning is likely required.

Computational Requirements. Standard fine-tuning requires NVIDIA A100-class GPUs, limiting reproducibility for researchers without access to high-end hardware. While LoRA substantially reduces requirements, deployment at scale still demands significant infrastructure.

Ethical Considerations

Dual-Use Concerns. This work aims to improve LLM safety by detecting malicious prompts. However, detailed analysis of model failure modes (e.g., polite phrasing bypasses detection) could potentially inform adversarial actors. We have balanced transparency for research reproducibility with responsible disclosure, avoiding providing specific adversarial templates.

Cultural and Linguistic Sensitivity. Our malicious prompt definitions are based on an English-centric safety corpus. Safety norms and definitions of harmful content may vary across cultures; content flagged as malicious in one cultural context may be acceptable in another. We acknowledge that our approach may over-flag culturally specific expressions or under-flag culturally specific harms not represented in the English source corpus.

Access and Equity. By developing safety mechanisms for low-resource Indic languages, this work aims to reduce the safety gap that leaves non-English speakers more vulnerable to AI harms. However, the computational requirements for full fine-tuning may limit deployment in the resource-constrained settings where these populations primarily reside. LoRA partially addresses this, and further efficiency research is warranted.

Data Ethics. Our dataset is derived from a publicly available English corpus and machine translation. No user data or personally identifiable information was collected or used. The dataset contains examples of malicious prompts; we will release it with appropriate usage guidelines and a data use agreement to prevent misuse. Researchers using this dataset should ensure compliance with applicable laws and ethical guidelines.

AI Writing Assistance. In accordance with ACL policy, we disclose that writing, structural improvements, and editorial refinements in this paper were assisted by a generative AI tool. The research design, experimental methodology, and all results are the authors' original work. All generated text was reviewed and verified for accuracy by the authors.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, and 1 others. 2023. MEGA: Multilingual evaluation of generative AI.

Yuntao Bai, Andy Jones, Kamal Ndousse, and 1 others. 2022. Constitutional AI: Harmlessness from AI feedback.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

CodeSagar. 2024. Malicious LLM prompts v4. Hugging Face Dataset.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*, pages 8440–8451. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186. ACL.

Jay Gala, Raj Dabre, Nishant Kambhatla, and 1 others. 2023. IndicTrans2: Towards high-quality and accessible machine translation of all 22 scheduled Indian languages.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, N. C. Gokul, Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of EMNLP*, pages 4948–4961. ACL.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savita Goyal, Avik Chadha, Vikram Dalmia, Nilesh Garg, Gurpreet Singh, Rajat Bhatt, and Partha Talukdar. 2021. MuRIL: Multilingual representations for Indian languages.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of ICCV*, pages 2980–2988.

Long Ouyang, Jeffrey Wu, Xu Jiang, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618. ACL.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*, pages 223–231.

Catherine Wang, Paul Pu Liang, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2023. Not all languages are created (tokenized) equal.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proceedings of ICLR*.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models.

A Hyperparameter Search Details

Standard Fine-Tuning Grid.

- Learning Rate: {2e-5, 3e-5, 5e-5}
- Batch Size: {16, 32}
- Epochs: 3–5 (with early stopping, patience=3)
- Warmup Steps: 500
- Weight Decay: 0.01
- Max Sequence Length: 512

LoRA Fine-Tuning Grid.

- Learning Rate: {3e-4, 5e-4, 7e-4}
- Batch Size: {32, 64}
- LoRA Rank (r): 16
- LoRA Alpha (α): 32
- LoRA Dropout: 0.1
- Target Modules: query, value projections
- Epochs: 3–5 (with early stopping)

Loss Function Parameters.

- Focal Loss: $\gamma = 2.0$; class weights [0.75, 0.25] (benign, malicious)
- LLRD: Decay rate 0.95; base learning rate per grid above

B Dataset Statistics

Table 3: Dataset statistics for all three languages.

Lang.	Total	Benign	Malicious	Avg Len	Split
Hindi	8,660	7,632	1,028	47.3	70/15/15
Tamil	8,660	7,632	1,028	52.1	70/15/15
Telugu	8,660	7,632	1,028	49.8	70/15/15

C Computational Resources

All experiments were conducted on:

- **GPU:** NVIDIA A100 (40 GB)
- **CPU:** AMD EPYC 7763 (64 cores)
- **RAM:** 256 GB
- **Storage:** 2 TB NVMe SSD

Training times: standard fine-tuning ≈4 hours/model/language; LoRA ≈1–2 hours/model/language. Total compute: ≈500 GPU-hours; estimated cloud cost ≈$2,000 (AWS `p4d.24xlarge`).

D Reproducibility Instructions

The full codebase, data processing scripts, and experiment configurations are available at the anonymous repository: https://anonymous.4open.science/r/[ANONYMIZED].

Environment Setup.

```
git clone [anonymous-repo-url]
cd malicious-prompt-detection
pip install -r requirements.txt
```

Dataset Creation. Open notebooks in `dataset/translation_code/` (one per language) in Google Colab or equivalent. Run all cells to: (1) download the English base corpus, (2) translate using IndicTrans2, (3) back-translate for quality filtering, (4) compute quality metrics, and (5) apply thresholds (see Section 3.1). Output CSV files are written to `dataset/filtered_datasets/`.

Some Hugging Face models require authentication:

```
from huggingface_hub import login
login(token="YOUR_HF_TOKEN")
```

Training.

- Standard fine-tuning: scripts in `finetune/regular_finetune/` (one per model architecture).
- LoRA/PEFT fine-tuning: scripts in `finetune/lora_finetune/` with LoRA settings from Appendix A.

Evaluation. Metrics are computed by aggregating across configurations in Appendix A. Dataset sizes match Appendix B; hardware matches Appendix C.