Performance Comparison of State-of-the-Art NLP Models for Urdu Hate Speech Detection

Maimoona Khilji[1], Haris Iqbal[1], Afsheen Khalid[1], Dilawar Khan[2], Muhammad Shahan Ibad*[1]
[1]Department of Computer Science, Institute of Management Sciences, Peshawar, Pakistan
[2]Department of Computer Science, University of Engineering and Technology, Peshawar, Pakistan
*Correspondence: bsds.225301943@imsciences.edu.pk

Citation | Khilji. M, Iqbal. H, Khalid. A, Khan. D, Ibad. M. S, "Performance Comparison of State-of-the-Art NLP Models for Urdu Hate Speech Detection", IJIST, Vol. 8 Issue. 9 pp 3877-3893, October 2026

Received | Oct 08, 2026 **Revised |** Oct 22, 2026 **Accepted |** Oct 26, 2026 **Published |** Oct 30, 2026.

The growth of social media has significantly increased the spread of hate speech in the Urdu language. Urdu is a low-resource language, and building accurate natural language processing models for Urdu remains a challenging task. The main reasons are linguistic complexity and limited labelled data. Manual content moderation is slow, subjective, and inconsistent. To address this issue, automated hate speech detection using natural language processing has become a technical necessity for maintaining safe digital platforms. This research presents a comparative evaluation of multilingual and Urdu-specific transformer-based models for binary hate speech detection in Urdu text. Four transformer models were compared: XLM-RoBERTa (multilingual baseline), urdu-bert-ner and roberta-urdu-small (Urdu-specific), and mmBERT-base (a recent multilingual encoder). All models were fine-tuned under one protocol on a public corpus of 26,174 usable Urdu texts, using a stratified, duplicate-aware train/validation/test split, learning-rate selection on validation macro-F1, and three random seeds. On the test set, mean accuracy ranged from 78.43% to 79.83%, macro-precision from 78.47% to 79.90%, macro-recall from 78.41% to 79.82%, and macro-F1 from 78.41% to 79.82%. XLM-RoBERTa and mmBERT-base performed best (79.8% macro-F1) and were significantly better than urdu-bert-ner by about 1.4 points (Holm-adjusted $p \leq 0.005$); mmBERT-base also exceeded roberta-urdu-small by 1.1 points ($p = 0.028$). XLM-RoBERTa had the lowest evaluation loss (0.460 versus 0.511 to 0.536) and the lowest median inference latency on a Tesla T4 GPU (12.4 ms versus 13.5 to 29.3 ms), whereas mmBERT-base was about 2.4 times slower without an accuracy gain. A dataset audit found 1,419 texts labelled both hate and no-hate, which limits attainable accuracy; on texts with a single label, accuracy rose to 81.9% to 83.4%. The results show no advantage for Urdu-specific pretraining in this setting, and XLM-RoBERTa gave the best balance of accuracy, calibration and speed among the models tested.

Keywords: Urdu Hate Speech Detection, Low-Resource NLP, Transformer Models, Multilingual Models, Urdu Language Processing

Introduction:

The development of social media platforms and user-generated content providers has transformed how people globally interact with each other. The current digital revolution lets anyone access information about anyone else and allows people from different countries to communicate in real time. This connectedness also allows for an expedient spread of hate speech, which divides people in Urdu-speaking communities. Millions of South Asians and others are using sites like Facebook, X (formerly Twitter), and WhatsApp daily, and therefore, the amount of unregulated text online is at an unprecedented level. The increase in internet use has both created the possibility of people finding ways to connect with one another but has also led to an increase in the spread of harmful ideas. Removing hate speech from the internet is not just a technical solution anymore; it is now an ethical requirement if we hope to create a safe and respectful digital space for Urdu speakers worldwide. Hate crimes often lead to physical harm to others, socially isolating and/or creating division, creating a sense of enmity or intense rivalry between ethnic groups, and/or resulting in physical attacks against groups of people based on the ways one is different from others, like religion, ethnicity, gender, etc. Recently, some advances have been achieved in the field of automatic detection of hate speech in some languages with the necessary resources to financially pay for an engineering team to build models such as English. However, those models cannot be easily translated directly into Urdu due to the unique script, grammar and culture that exists there. In addition, there is a large void of automated moderation applications for the Urdu web. Additionally, despite being the second-most spoken language in India, there are also few digital resources available to support this language. A large part of it is due to many years of Natural Language Processing (NLP) research on languages like English, as well as access to large English corpora (280 billion words). These challenges make Urdu hate speech detection a distinct classification problem in which linguistic variation, contextual interpretation, and the availability of reliable labelled data can influence model performance. Therefore, evaluating pretrained language models on a common dataset and under a clearly documented experimental protocol is important for obtaining a meaningful comparison.

Conversely, models developed for Latin-based languages cannot adequately learn to function with Persian-Arabic scripts. This poses numerous complicated issues for regularly tokenizing and feature extraction functions in Latin languages. However, developing accurate models for Urdu is particularly difficult for three main reasons; First and foremost, there is not enough high-quality data available, meaning that there are too few large, high-quality datasets specifically developed for the Urdu language. Datasets that can be found in libraries are generally not clearly recognized as having proper annotation or are too limited in size for effective deep learning training on such data sets. Second, there is an issue of context; throughout the world, there is no unified interpretation of hate or what constitutes hate due to differences in social norms and different sets of sensitizing circumstances that have happened in Urdu speaking parts of the world. For example, a word used every day in Western Hindi may carry an unrecognized level of religious/historical toxicity when used in Urdu. Lastly, language differences create additional barriers. For example, current models do not process the large amount of code-switching that occurs both between English and Urdu and also as there are now both Roman Urdu and Urdu script.

To address this gap, this study investigates binary hate speech detection in Urdu, classifying text into two categories: hate speech and non-hate speech. Four pretrained transformer models are compared: XLM-RoBERTa, Urdu-BERT-NER, RoBERTa-Urdu-Small, and mmBERT-base. These models represent multilingual and Urdu-oriented pretrained approaches and provide a basis for examining their relative performance on the same classification task.

Although pretrained language models have been applied to hate speech and related text classification tasks, comparing their reported results across studies is difficult because datasets, label definitions, language varieties, and evaluation procedures may differ. Consequently, performance reported on one dataset or task does not necessarily establish that a model is superior for binary Urdu hate speech detection. A focused comparison of multilingual and Urdu-oriented pretrained transformers on a common dataset can provide more directly comparable evidence. In addition to classification performance, examining class-specific metrics, prediction errors, and computational efficiency can help clarify the practical trade-offs between the evaluated models.

Accordingly, this study evaluates four pretrained transformer models for binary Urdu hate speech detection using a common dataset and a documented evaluation protocol. This study provides a comparative evaluation of existing models, examining their classification performance and computational characteristics. The findings are intended to inform future research on Urdu hate speech detection and to provide evidence for selecting models for further investigation in practical text classification settings. The study's results should be interpreted in the context of the selected dataset, experimental configuration, and evaluation conditions.

Objectives:

The following objectives guided this research:

To compare XLM-RoBERTa, Urdu-BERT-NER, RoBERTa-Urdu-Small, and mmBERT-base for binary hate speech detection in Urdu.

To evaluate model performance using multiple metrics, including accuracy, precision, recall, F1-score, evaluation loss, and inference time, where applicable.

To examine the trade-off between classification performance and computational efficiency across the evaluated models.

To analyse model prediction errors and assess the uncertainty of performance differences, as supported by the experimental results.

To establish a documented comparative baseline that can support future research on Urdu hate speech detection and related text classification tasks.

Novelty Statement:

The contribution of this study is an empirical comparison of multilingual and Urdu-oriented pretrained transformer models for binary Urdu hate speech detection using a common dataset. In addition to comparing classification metrics, the study considers computational efficiency to provide a broader view of model performance. Where supported by the final experimental results, error analysis and uncertainty estimates are used to examine the reliability and practical significance of observed differences. The study does not propose a new transformer architecture; instead, it provides a comparative and diagnostic evaluation of existing models under the documented experimental conditions.

Literature Review:

Automatic hate speech detection for Urdu has developed along several lines: fine-tuning pretrained transformers, classical machine learning and deep learning with Urdu-specific processing, adaptation of large language models, and ensembles. This section reviews these lines critically and identifies the gaps addressed by the present study. Table 1 summarizes the studies.

[1] fine-tuned Urdu-RoBERTa and Urdu-DistilBERT, with grid-search tuning, on a Nastaliq Urdu corpus of Facebook posts for binary hate speech detection and target-community identification. [2] combined differential transfer learning with adaptive loss functions (DAmBERT) on 18,058 manually annotated Nastaliq YouTube comments and obtained a weighted F1 of 91.49%, outperforming traditional machine learning and deep learning baselines. Other comparisons point the opposite way. On 4,808 Urdu tweets, [3]

found that a 1D-CNN with unigram features outperformed Urdu-BERT and Urdu-RoBERTa in accuracy and macro-F1, whereas [4], using about 3,600 tweets for threat detection, found fine-tuned RoBERTa best among machine learning, deep learning and transformer models. These conflicting conclusions on small corpora indicate that the ranking of model families depends on the dataset, and we did not find significance tests or repeated runs reported for these comparisons.

Several studies include Urdu within multilingual settings. [5] studied English, Urdu and Sindhi and reported a 91% F-score for binary classification and an 86% weighted F-score for five fine-grained categories using XLM-RoBERTa. [6] compared translation-based and joint multilingual strategies for Arabic and Urdu on the UA-HSD-2025 dataset, and XLM-R performed best. [7] fine-tuned mBERT and XLM-RoBERTa for multi-class hate speech detection across high- and low-resource languages and reported a 31% improvement over the baseline for Urdu. [8] built a trilingual framework for English, Urdu and Spanish on 10,193 tweets, combining attention layers with transformers and large language models (GPT-3.5 Turbo, Qwen 2.5 72B), and outperformed traditional baselines such as SVM. These multilingual studies mix languages and label schemes, so the specific effect of Urdu-focused versus multilingual pretraining is rarely isolated.

[9] addressed threat detection in Urdu, where limited NLP resources, inconsistent vocabulary and weak preprocessing tools are obstacles. They introduced an Urdu-specific vocabulary, stopword list and stemming dictionary and trained an LSTM-based sequential model that reached 82% accuracy, showing the value of language-specific preprocessing. [10] created a dataset of 12,020 Urdu tweets following the OLID taxonomy and compared seven machine learning classifiers, of which logistic regression and SVM performed best. [11] introduced the Urdu Toxicity Corpus, a multilabel Nastaliq dataset, and evaluated TF-IDF, bag-of-words and n-gram features with traditional and deep models; Random Forest with SMOTE performed best. [12] proposed an attention-based bidirectional GRU (BiGRU-A) for inappropriate content detection, and the attention mechanism improved results over LSTM, BiLSTM, GRU and TCN baselines. The competitiveness of simple models on some corpora makes transformer gains harder to demonstrate without careful, repeated evaluation.

A large share of recent work targets Roman Urdu. [13] added a Profanity Check Technique layer (ReLU activation with a logistic regression classifier) to a fine-tuned multilingual BERT and reported 93.4% accuracy. [14] trained a Roman Urdu BERT (BERT-RU) from scratch on 173,714 messages and reported 96.70% accuracy. [15] showed that data augmentation improves a fine-tuned mBERT on Roman Urdu tweets. [16] evaluated machine learning, deep learning and transformer approaches on Roman Urdu YouTube comments: SVM with TF-IDF and n-grams was best among traditional models, CNN outperformed BiLSTM, and LLaMA 2 fine-tuned with LoRA achieved the highest F1 (96.58). [17] applied QLoRA fine-tuning to Roman Urdu-English code-mixed text, with LLaMA 3 reaching an F1 of 91.45. [18] released 46,045 Roman Urdu comments for insult detection, where ensemble methods reached an F1 of 97.79%. [19] built a Roman Urdu Instagram corpus on transgender-related content with a slur lexicon and emoji taxonomy, and their custom BERT model reached an F1 of 77.34% for hate speech. [20] compared zero-shot inference with LoRA fine-tuning of Mistral, LLaMA, Falcon and multilingual BERT on the PURUTT corpus of more than 72,000 comments. Because Roman Urdu uses the Latin script, these results do not transfer directly to Nastaliq text.

[21] collected 36,000 Urdu tweets labelled hate, offensive or neutral and combined FastText, XLM-RoBERTa, ULMFiT and XGBoost in an ensemble that reached a macro-F1 of 0.94 under stratified 5-fold cross-validation. [22] used a stacked ensemble of ELECTRA, mBERT and XLM-RoBERTa for Urdu fake news detection (accuracy 0.914, F1 0.904). Ensembles improve accuracy but increase computational cost. Work on neighboring tasks

informs architecture choice but is not directly comparable: [23] used a hybrid CNN-LSTM with FastText embeddings for Urdu sentiment analysis (96.9% accuracy), [24] proposed an explainable framework for threatening-tweet identification on an English corpus (73.15% accuracy), and [25] used concept graphs to analyses Urdu news.

Evaluation quality depends on the data. Overlap between training and test sets can inflate results, because models are rewarded for memorizing rather than generalizing [26]. Duplicated text in datasets can also cause researchers to overestimate accuracy and bias model and hyperparameter selection towards overfitting [27]. In hate speech, annotator disagreement is a recognized source of label noise [28]. In the Urdu studies reviewed here, we did not find reports of audits for duplicate texts or contradictory labels.

The reviewed studies differ widely in corpus size (about 3,600 to more than 72,000 texts), script (Nastaliq or Roman), label scheme (binary, three-class, multilabel) and validation design, and their reported scores (roughly 77% to 98%) mostly reflect these differences. They cannot be ranked against one another. Three gaps follow.

The highest reported scores come from ensembles, large language models and models trained from scratch, while compact Urdu-specific encoders and newer multilingual encoders have not been compared under one controlled protocol. mmBERT [29], whose training schedule aims to improve low-resource languages, has to our knowledge not been evaluated on Urdu hate speech.

Parameter-efficient fine-tuning papers discuss memory savings [16][17][20], but in the studies reviewed we did not find inference latency or throughput reported for Urdu hate speech detection, even though moderation systems need fast inference.

Nastaliq-script corpora receive less systematic comparison than Roman Urdu, preprocessing resources are limited [9], and corpora are seldom audited for duplicates and label consistency.

The present study addresses these gaps by comparing four transformer models (XLM-RoBERTa, urdu-bert-ner, roberta-urdu-small and mmBERT-base) on one public Nastaliq corpus of 26,198 records (26,174 usable, 22,319 unique after text normalisation), with a duplicate-aware split, repeated seeds, paired significance tests, and measured inference latency.

Table 1. Summary of Related Work on Urdu / Roman Urdu Text Classification.

Ref	Authors (Year)	Script / language	Dataset / size	Model / approach	Key finding
[1]	Malik et al. (2024)	Nastaliq Urdu	Facebook posts (HSTC)	Urdu-RoBERTa, Urdu-DistilBERT	Binary hate and target-community detection
[2]	Hussain et al. (2025)	Nastaliq Urdu	18,058 YouTube comments	DAmBERT (transfer learning)	91.49% weighted F1
[3]	Khan et al. (2024)	Urdu	4,808 tweets	1D-CNN (unigrams)	Beat Urdu-BERT and Urdu-RoBERTa
[4]	Raza et al. (2025)	Urdu	about 3,600 tweets	Fine-tuned RoBERTa	Best of ML, DL, transformers
[5]	Siddiqui et al. (2024)	English, Urdu, Sindhi	Curated multilingual set	XLM-RoBERTa	91% F-score (binary); 86% weighted F (fine-grained)

[6]	Ahmad et al. (2025)	Arabic, Urdu	UA-HSD-2025	XLM-R, translation and joint models	XLM-R best
[7]	Srikissoon and Marivate (2023)	Multilingual incl. Urdu	Multi-class	mBERT, XLM-R	31% lift over baseline for Urdu
[8]	Usman et al. (2025)	English, Urdu, Spanish	10,193 tweets	Transformers and LLMs	Outperformed SVM
[9]	Ullah et al. (2024)	Urdu	Threat data	LSTM with Urdu preprocessing	82% accuracy
[10]	Ud Din et al. (2025)	Urdu	12,020 tweets	LR, SVM and others	LR and SVM best
[11]	Rashid et al. (2025)	Nastaliq Urdu	UTC corpus	Random Forest + SMOTE	RF best (0.99 accuracy as listed in the earlier table)
[12]	Shoukat et al. (2025)	Urdu	Inappropriate content	BiGRU-A	Attention improved results
[13]	Ali et al. (2025)	Roman Urdu	not stated	mBERT + PCT	93.4% accuracy
[14]	Bilal et al. (2023)	Roman Urdu	173,714 messages	BERT-RU (from scratch)	96.70% accuracy
[15]	Maqbool et al. (2024)	Roman Urdu	RUHSOLD++	mBERT + augmentation	Augmentation improved results
[16]	Zain et al. (2025)	Roman Urdu	YouTube comments	LLaMA 2 + LoRA	96.58 F1; beat CNN and SVM
[17]	Hussain et al. (2025)	Roman Urdu-English	Code-mixed	LLaMA 3 + QLoRA	91.45 F1
[18]	Khan et al. (2025)	Roman Urdu	46,045 comments	Ensembles (CNN, LSTM)	97.79 F1
[19]	Raza et al. (2025)	Roman Urdu	Instagram comments	Hate-RU-BERT	77.34% F1 (hate)
[20]	Zubair et al. (2026)	Roman Urdu	PURUTT, 72,000+ comments	LoRA vs zero-shot LLMs	LoRA compared with zero-shot
[21]	Ullah et al. (2025)	Urdu	36,000 tweets	Ensemble (FastText, XLM-R, ULMFiT, XGBoost)	0.94 macro-F1
[22]	Harris et al. (2025)	Urdu	Fake news	Stacked ELECTRA, mBERT, XLM-R	0.914 accuracy

[23]	Ahmad et al. (2025)	Urdu	Sentiment	CNN-LSTM + FastText	96.9% accuracy
[24]	Malik et al. (2025)	English	Twitter corpus	AMR graph + pattern structure	73.15% accuracy
[25]	Mehmood et al. (2025)	Urdu	News articles	Concept graphs	Relationship visualisation

Methodology:

This study compares four transformer-based models for binary Urdu hate speech detection (hate versus no-hate) under one controlled protocol. Figure 1 shows the workflow in the order it was executed. The public corpus was loaded, cleaned, and audited for repeated texts and contradictory labels. It was then divided once into stratified, duplicate-aware training, validation, and test sets. Each model's own tokenizer converted the text into token IDs, and each model was fine-tuned at three candidate learning rates on one random seed. The learning rate with the highest validation macro-F1 was selected, and the selected configuration was repeated with two further seeds. The best-validation model of every run was evaluated once on the test set. Statistical comparison, a conflict-free sensitivity analysis, and a latency measurement followed.

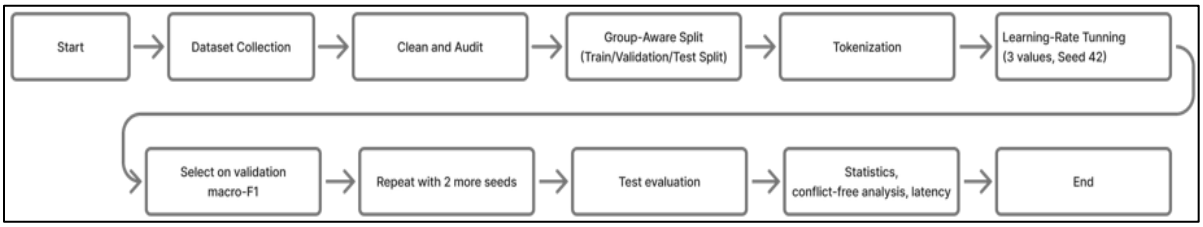

Figure 1. Experimental workflow.

Dataset Description:

A publicly available dataset, "Urdu Hate Speech," obtained from Hugging Face [30], is used for comparative analysis. This dataset contains 26,198 short-form Urdu texts and their corresponding binary labels indicating "hate" and "no-hate" speech. The origin of the data was public tweets collected via the Twitter API using Urdu keywords, hate-speech lexicon terms, and topic-specific queries. The first five rows are shown in Figure 2.

Label	Text
1	یہ بھی دیکھیں آپا اب یہ چھمک چھلو ٹک ٹاک سے بھی کمائی کرے گی بیلی ڈانس کر کے
0	آپ کو میں فالو نہیں کرتی پر انٹرایکشن ہو جاتا ہے کبھی کبھی پہلے پہل کچھ خاص پسند نہیں آتے تھے آپ کے ریلائی پر اب رائے کچھ مثبت ہونے لگی ہے
1	جھوٹا ہے یہ انجینئر، یہ صاحبوں کی جھوٹی باتیں بتا کر مسلمانوں میں اختلاف پھیلا رہا ہے۔ لانت ہو
1	تمہارے بزرگ تو کباڑی ہوتے تھے تم صحافی لفافی بن گے
1	اب یہ لنڈے لڑے آ گئے ہیں۔ انہیں جوازت کرنا

Figure 2. Depression Student Dataset.

The data were manually annotated by domain-aware annotators following written guidelines. There are two label sets:

not_hate (0): Neutral, benign, counterspeech, or non-hateful opinions.

Hate (1): Hateful speech like dehumanising, insulting, or advocating violence or discrimination against protected groups or individuals based on protected characteristics.

Additionally, after removing 24 records with empty text, 26,174 records remain: 12,993 no-hate (49.62%) and 13,181 hate (50.38%). The classes are therefore balanced, as shown in Figure 3, and no resampling or class weighting was used.

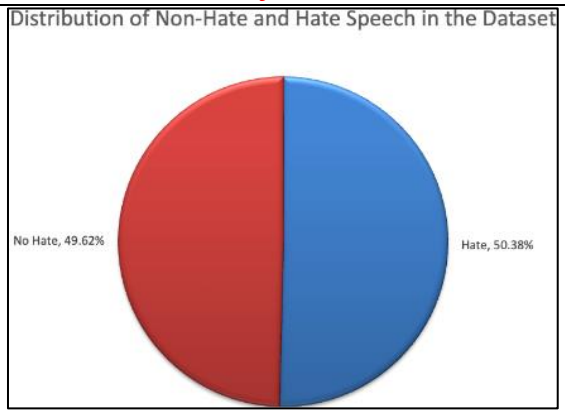

Figure 3. Class distribution of the corpus after removing empty texts

According to the dataset documentation, the dataset creators filtered the data as follows: retweets and quote-tweets were removed where applicable; duplicate and near-duplicate messages were removed; images and other media were removed, keeping only text; heuristic script filtering retained Urdu content; and user handles and URLs were normalised to remove personal data. The creators also removed diacritics and tatweel (kashida) when present, collapsed multiple spaces, normalised punctuation spacing, replaced user mentions with @user and URLs with <url>, stripped leading retweet markers, and removed non-textual artefacts such as leftover HTML entities. Urdu is caseless, and the original letter forms are preserved. Despite the stated removal of duplicates, our audit (below) found repeated and contradictory records.

Data Audit:

To detect repeated texts, each text was normalized using Unicode NFKC, lower-casing, and whitespace collapsing and trimming. This normalised form was used only to group texts; the models receive the original text. The audit found 22,319 unique normalised texts among the 26,174 records, so 3,855 records repeat an earlier text. In addition, 1,419 normalised texts occur with both labels, covering 2,867 records (10.95% of the corpus). Each such text has at least one copy whose label disagrees with another copy, so any classifier makes at least 1,419 errors on the full corpus, and the maximum attainable accuracy is about 94.6%. These records were kept in the main experiments to leave the public data unaltered, and a conflict-free sensitivity analysis is reported separately.

Data Preprocessing:

No further cleaning was applied, since transformer-based models can handle raw text effectively. Each model used its own pre-trained tokenizer to convert Urdu text into token IDs. Texts were truncated at a maximum length of 128 tokens, and padding was applied dynamically to the longest sequence in each batch. Tokenization was performed in batches to improve efficiency.

Table 2. Data partitions.

Partition	Records	Share	Unique texts	Hate share	Records from conflicting texts
Train	18,696	71.4%	15,936	50.36%	2,055
Validation	3,739	14.3%	3,192	50.36%	408
Test	3,739	14.3%	3,191	50.36%	404

The corpus was divided once, using a fixed random seed of 42, with scikit-learn's StratifiedGroupKFold (seven folds, shuffled). One-fold was used as the test set, one as the validation set, and the remaining five as the training set. Stratification keeps the label proportions equal in all partitions, and grouping by normalized text keeps all copies of a text in the same partition. This prevents text-level data leakage, and we verified that no normalized

text appears in more than one partition. The split was saved and reused for every model and seed. The test set was never used for tuning or early stopping.

Models:

This research evaluated four transformer-based encoders to compare their effectiveness for Urdu hate speech detection (Table 3).

Table 3. Model categories for Urdu hate speech detection.

Category	Model	Checkpoint	Parameters (M)
Multilingual baseline	XLM-RoBERTa	FacebookAI/xlm-roberta-base	278.0
Urdu-specific	urdu-bert-ner	mirfan899/urdu-bert-ner	177.9
Urdu-specific	roberta-urdu-small	urduhack/roberta-urdu-small	126.0
Recent multilingual	mmBERT-base	jhu-clsp/mmBERT-base	307.5

XLM-RoBERTa is a strong multilingual transformer trained on 100 languages [31]. urdu-bert-ner is a domain-specific Urdu BERT model originally trained for named entity recognition [32]. roberta-urdu-small is a lightweight RoBERTa model optimized for Urdu [33]. mmBERT-base is a recent multilingual encoder trained on more than 3 trillion tokens across more than 1,800 languages, introducing novel techniques for learning low-resource languages during the decay phase [29]. Each model receives a newly initialized two-class classification head and is fine-tuned in full for binary classification (hate vs no-hate). Parameter counts include the classification head.

Training Configuration:

All models were fine-tuned with the Hugging Face Trainer API using identical settings (Table 4).

Table 4. Training configuration.

Setting	Value
Optimizer	AdamW (adamw_torch)
Learning rate	Selected from 1e-5, 2e-5, and 3e-5 (validation macro-F1, seed 42)
Schedule	Linear decay with 10% warm-up
Weight decay	0.01
Batch size	16 (training), 64 (evaluation); no gradient accumulation
Maximum epochs	5, with evaluation after every epoch
Early stopping	Patience of 2 epochs on validation macro-F1
Checkpoint used	Best validation macro-F1, reloaded at the end of training
Maximum sequence length	128 tokens, dynamic padding
Precision	fp16 mixed precision
Random seeds	42 (tuning and first run); 123 and 2026 (repeat runs)
Hardware	NVIDIA Tesla T4 GPU (Google Colab)
Software	Python 3.13.16, PyTorch 2.11.0 (CUDA 13.0), Transformers 4.57.6

The learning rate was selected separately for each model on seed 42, using validation macro-F1 only. The selected values were 1e-5 for XLM-RoBERTa, 2e-5 for mmBERT-base, and 3e-5 for urdu-bert-ner and roberta-urdu-small. Each selected configuration was then repeated with seeds 123 and 2026, giving three runs per model. Every run evaluated its best-validation model once on the test set. Test predictions were also produced for the tuning runs but were not used for any selection.

A fair comparison was ensured by using one shared split, the same learning-rate grid, batch size, epoch budget, and stopping rule for every model, selection on validation data only, and repeated seeds. Reproducibility is supported by fixed seeds, a saved split, and released code and prediction files. Fixed seeds do not make GPU training bit-for-bit identical.

Evaluation Metrics:

Model performance is evaluated on the test set using the following metrics. A confusion matrix compares actual and predicted values, with hate as the positive class:

Table 5. Confusion Matrix

	Predicted Negative	**Predicted Positive**
Actual Negative	TN	FP
Actual Positive	FN	TP

where TP (true positive) is a hate text correctly predicted as hate, TN (true negative) is a no-hate text correctly predicted as no-hate, FP (false positive) is a no-hate text incorrectly predicted as hate, and FN (false negative) is a hate text incorrectly predicted as no-hate. The following metrics are derived from it:

Accuracy, the overall proportion of correct predictions:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

Precision, the share of predicted positives that are actually positive:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (2)$$

Recall, the share of actual positives that are correctly predicted:

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

F1 score, the balance between precision and recall:

$$\text{F1} = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (4)$$

Precision, recall, and F1 are computed for each class and **macro-averaged**, meaning the unweighted mean of the hate and no-hate values. The F1 of the hate class is also reported.

Evaluation loss is the mean cross-entropy (log loss) of the predicted probabilities on the test set:

$$\text{Loss} = -(1/N)\, \Sigma\, [\, y_i\, log(p_i) + (1 - y_i)\, log(1 - p_i)\,] \quad (5)$$

where y_i is the true label, p_i is the predicted probability of hate, and N is the number of test texts. It measures calibration as well as correctness: a model with high accuracy can still have a high loss if it makes confident errors.

Repeated Runs: For each model, the mean and standard deviation over the three seeds is reported, with a 95% confidence interval from the t-distribution (two degrees of freedom).

Significance Testing: Models were compared with a paired bootstrap over test texts. In each of 2,000 resamples, groups of identical normalised texts were drawn with replacement, so that repeated texts stay together. Each model's metric was averaged over its three seeds, and the difference between models was recorded. The 95% interval is the 2.5th to 97.5th percentile of the differences, and the two-sided p-value is twice the smaller tail proportion. The six pairwise p-values were adjusted with the Holm procedure. This test captures variability from test-set sampling; variability from training randomness is shown by the seed standard deviations.

Conflict-Free Sensitivity Analysis: All metrics were also recomputed on the 3,335 test records whose text carries a single label.

Inference Cost: Latency is the time to tokenize and classify one text (batch size 1), measured over 200 requests after 20 warm-up requests on the Tesla T4 with fp16 autocast, and reported as the median and 95th percentile. Throughput is the number of texts classified per second at batch size 32. Because speed does not depend on the learned weights, these measurements used each pretrained checkpoint with an untrained classification head. They describe this GPU only.

Hyperparameters and checkpoints were selected on validation macro-F1, and the test set was used once per run for final reporting.

Results and Discussion:

This section reports the test-set performance of the four models, the learning-rate selection, the statistical comparison, a sensitivity analysis on texts with a single label, and the

inference cost. All models were evaluated on the same 3,739-record test set, and all values are means over three random seeds unless stated otherwise.

Predictive Performance:

Table 6 shows the test performance, and Figure 4 shows it graphically. All four models fall in a narrow band of 78.4% to 79.8% accuracy. XLM-RoBERTa and mmBERT-base are tied at the top, with a mean accuracy of 79.83% and a mean macro-F1 of 79.8%. They are followed by roberta-urdu-small (78.73% accuracy, 78.69% macro-F1) and urdu-bert-ner (78.43% accuracy, 78.41% macro-F1). Precision and recall are balanced for every model, and the hate-class F1 follows the same ordering, from 80.4% (XLM-RoBERTa) to 78.9% (urdu-bert-ner). Seed-to-seed variation is small for XLM-RoBERTa and roberta-urdu-small (standard deviation about 0.3 points) and larger for mmBERT-base (0.83 points). With three seeds, the 95% confidence intervals are wide: ±0.72 points for XLM-RoBERTa, ±2.07 for mmBERT-base, ±0.74 for roberta-urdu-small, and ±1.39 for urdu-bert-ner.

Table 6. Test-set performance, mean ± SD over three seeds (n = 3,739; hate is the positive class; precision, recall, and F1 are macro-averaged unless stated).

Model	LR	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Hate-class F1 (%)	Log loss
XLM-RoBERTa	1e-5	79.83 ± 0.29	79.90 ± 0.23	79.81 ± 0.30	79.81 ± 0.30	80.38 ± 0.01	0.460 ± 0.015
mmBERT-base	2e-5	79.83 ± 0.83	79.88 ± 0.87	79.82 ± 0.82	79.82 ± 0.82	80.27 ± 1.05	0.536 ± 0.209
roberta-urdu-small	3e-5	78.73 ± 0.30	78.92 ± 0.55	78.71 ± 0.28	78.69 ± 0.26	79.44 ± 0.90	0.511 ± 0.085
urdu-bert-ner	3e-5	78.43 ± 0.56	78.47 ± 0.60	78.41 ± 0.55	78.41 ± 0.55	78.86 ± 0.88	0.515 ± 0.056

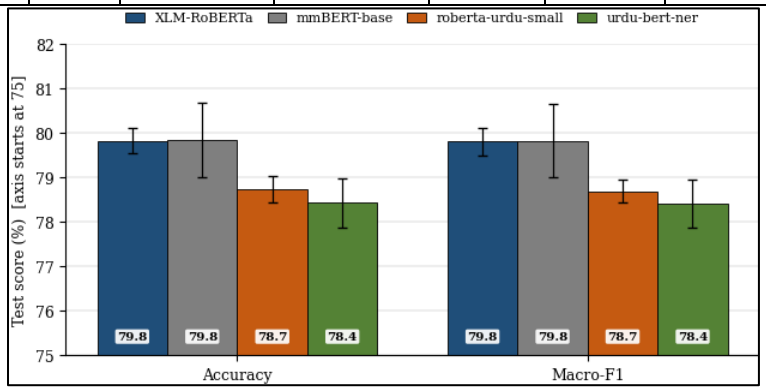

Figure 4. Test accuracy and macro-F1 of the four models.

Hyperparameters were chosen on the validation set. Table 7 shows the validation macro-F1 for each learning rate on seed 42. The differences between learning rates are about one point or less, and three models selected a value at the edge of the grid (1e-5 for XLM-RoBERTa, 3e-5 for urdu-bert-ner and roberta-urdu-small).

Table 7. Validation macro-F1 (%) by learning rate (seed 42). The selected value is in bold.

Model	1e-5	2e-5	3e-5
XLM-RoBERTa	**79.16**	78.63	78.90
urdu-bert-ner	76.84	77.73	**77.94**
roberta-urdu-small	77.63	77.96	**78.46**
mmBERT-base	78.38	**78.57**	78.26

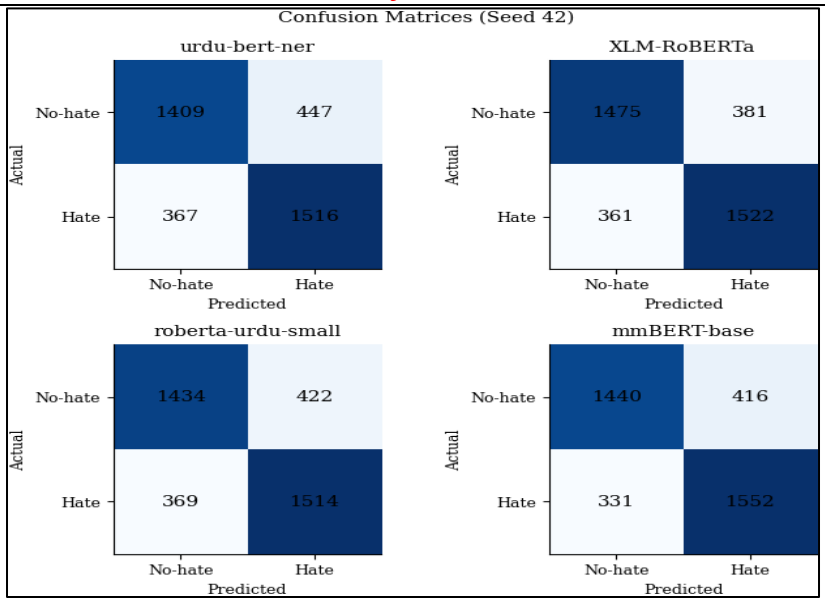

Figure 5. Confusion matrices of the seed-42 run of each model on the test set

Statistical Comparison:

Table 8 gives the paired bootstrap comparison of macro-F1. XLM-RoBERTa and mmBERT-base were significantly better than urdu-bert-ner, by about 1.4 points (Holm-adjusted $p = 0.005$ and $p < 0.001$). mmBERT-base was also significantly better than roberta-urdu-small, by 1.1 points ($p = 0.028$). The difference between XLM-RoBERTa and roberta-urdu-small (1.1 points) was borderline and not significant after correction ($p = 0.063$). The two Urdu-specific models were not distinguishable from each other, and neither were XLM-RoBERTa and mmBERT-base. These intervals describe variation due to the test sample; the differences are small in practical terms.

Table 8. Paired group-bootstrap comparison of macro-F1 (percentage points; 2,000 resamples; seeds averaged; Holm correction over six pairs).

Comparison (A − B)	Difference	95% CI	Holm p	Significant
XLM-RoBERTa − urdu-bert-ner	+1.39	0.53 to 2.27	0.005	Yes
mmBERT-base − urdu-bert-ner	+1.41	0.57 to 2.18	< 0.001	Yes
mmBERT-base − roberta-urdu-small	+1.14	0.27 to 1.99	0.028	Yes
XLM-RoBERTa − roberta-urdu-small	+1.12	0.15 to 2.13	0.063	No
roberta-urdu-small − urdu-bert-ner	+0.27	−0.63 to 1.20	1.000	No
XLM-RoBERTa − mmBERT-base	−0.02	−0.81 to 0.79	1.000	No

Calibration and Evaluation Loss:

XLM-RoBERTa had the lowest mean log loss and the smallest variation in log loss (0.460 ± 0.015), as shown in Figure 6. mmBERT-base reached the same accuracy but had a higher and much more variable loss (0.536 ± 0.209). Its validation loss also varied strongly with the learning rate (0.43 at 2e-5 versus 0.72 to 0.75 at the other two values). This pattern is consistent with overconfident errors in some runs, although we did not test the cause. In practice, mmBERT-base probabilities should be calibrated before a decision threshold is applied to them. The loss of roberta-urdu-small (0.511 ± 0.085) and urdu-bert-ner (0.515 ± 0.056) lay between the two.

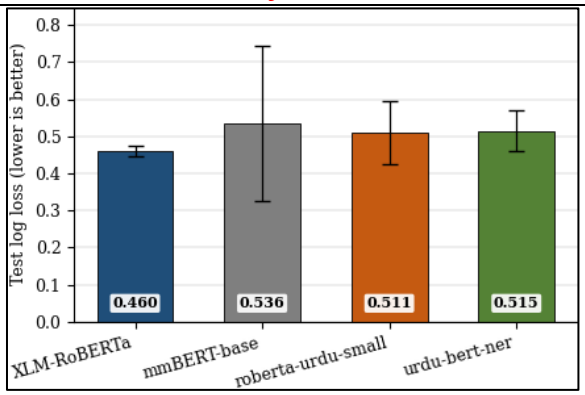

Figure 6. Test log loss of the four models.

Effect of Contradictory Labels:

The corpus contains texts that appear with both labels, and 404 of the 3,739 test records (10.8%) belong to such texts. Table 9 repeats the evaluation on the 3,335 test records with a single label. All models improve by about 3.4 to 3.6 points, and the ordering stays the same, with mmBERT-base and XLM-RoBERTa at 83.4%, roberta-urdu-small at 82.2%, and urdu-bert-ner at 81.9%. Contradictory labels therefore reduce measured performance for every model without changing the comparison.

Table 9. Conflict-free test subset (n = 3,335 records whose text has a single label).

Model	Accuracy (%)	Macro-F1 (%)	SD of macro-F1
mmBERT-base	83.45	83.44	0.93
XLM-RoBERTa	83.44	83.43	0.33
roberta-urdu-small	82.21	82.18	0.31
urdu-bert-ner	81.87	81.86	0.62

Computational Cost:

Table 10 and Figure 7 report inference cost on a Tesla T4. XLM-RoBERTa had the lowest median latency (12.4 ms for one text) and the highest throughput (943 texts/s). roberta-urdu-small and urdu-bert-ner followed closely at batch size 1 (14.3 and 13.5 ms), and roberta-urdu-small matched XLM-RoBERTa's throughput (922 texts/s), whereas urdu-bert-ner reached 686 texts/s. mmBERT-base was the slowest, with a median of 29.3 ms and 385 texts/s, about 2.4 times slower per request than XLM-RoBERTa and with less than half its throughput, for no gain in accuracy. Parameter count did not predict speed: roberta-urdu-small has less than half the parameters of XLM-RoBERTa but was not faster, plausibly because much of XLM-RoBERTa's size lies in its embedding table, which adds little inference cost (not tested here). The 95th-percentile latencies of the two Urdu-specific models (53 to 61 ms) are several times their medians and may reflect noise on a shared GPU, so tail latency should be interpreted with caution. During training, the selected configuration of roberta-urdu-small took about 560 s for four epochs, compared with 1,250 s for five epochs of XLM-RoBERTa and 1,500 s for four epochs of mmBERT-base (seed 42, including validation passes, on the same GPU), so the smallest model was the cheapest to train.

Table 10. Inference cost on a Tesla T4 (fp16; 200 single-text requests after 20 warm-up requests; throughput at batch size 32).

Model	Parameters (M)	Median latency (ms)	95th percentile (ms)	Throughput (texts/s)
XLM-RoBERTa	278.0	12.4	15.2	943
urdu-bert-ner	177.9	13.5	60.7	686
roberta-urdu-small	126.0	14.3	53.1	922
mmBERT-base	307.5	29.3	35.9	385

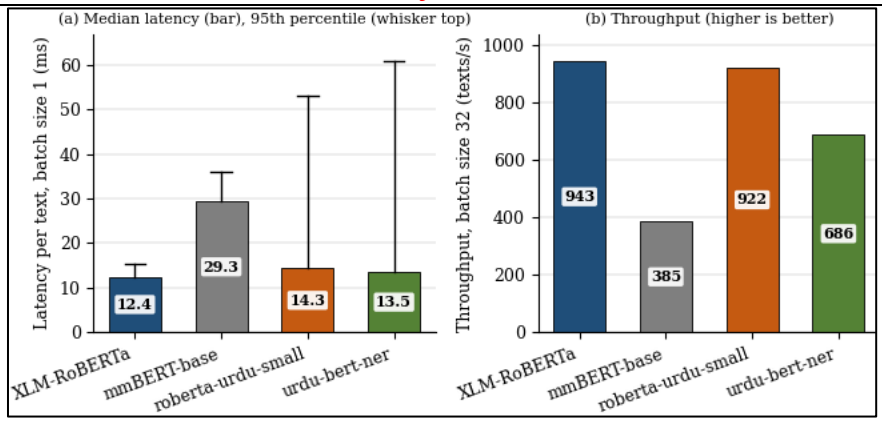

Figure 7. Median inference latency and throughput of the four models on a Tesla T4.

The results show no advantage for Urdu-specific pretraining in this setting. The multilingual models were at least as accurate as the Urdu-specific ones, and significantly more accurate in three of the four cross-group comparisons. urdu-bert-ner, whose pretraining objective was named-entity recognition, scored lowest, which is consistent with, but does not prove, a mismatch between that objective and hate speech classification. The recent multilingual encoder mmBERT-base matched XLM-RoBERTa in accuracy but was slower and less well calibrated, so its more modern pretraining recipe did not translate into an advantage on this corpus. Among the models tested, XLM-RoBERTa gave the best combination of accuracy, calibration, and speed.

Accuracy is lower than values often reported for random splits of corpora with repeated texts. Duplicates shared between training and test data are known to inflate scores [26][27] and the split used here prevents this. Because the earlier random split was not re-run, we cannot quantify the inflation for this corpus. Results are also consistent with the label noise reported for hate speech data [28]. On texts with a single label, accuracy rose to 82% to 83%.

Practical Implications and Limitations:

For moderation support, the differences among the four models are small enough that cost, calibration, and engineering fit can matter more than the last point of accuracy. On the tested GPU, XLM-RoBERTa and the smaller Urdu models are fast enough for near-real-time screening of short texts, whereas mmBERT-base is more expensive for no accuracy gain. Predictions at this accuracy level should support human reviewers rather than replace them, since roughly one in five texts is misclassified.

Conclusion:

This study compared four transformer models for binary Urdu hate speech detection under one controlled, duplicate-aware protocol. Mean test accuracy and macro-F1 lay between 78.4% and 79.8%, with XLM-RoBERTa and mmBERT-base highest at 79.8%. Both were significantly ahead of urdu-bert-ner by about 1.4 points, and mmBERT-base was also ahead of roberta-urdu-small by 1.1 points, whereas XLM-RoBERTa and mmBERT-base did not differ from each other. Seed-to-seed variation was small for XLM-RoBERTa and roberta-urdu-small (about 0.3 points) and larger for mmBERT-base (0.8 points). An audit showed that 1,419 texts carry both labels, which depresses measured performance by roughly 3.4 to 3.6 points and limits attainable accuracy to about 94.6%. On a Tesla T4, XLM-RoBERTa had the lowest median latency (12.4 ms) and the highest throughput (943 texts/s), the lowest log loss (0.460), and top-tier accuracy, whereas mmBERT-base was about 2.4 times slower with no accuracy gain. The evidence therefore does not show that Urdu-specific pretraining helps in this setting, and XLM-RoBERTa gave the best overall balance among the models tested. These conclusions are limited to this corpus and hardware and should not be read as general superiority claims.

Future Work:

Future work should include external validation on independent Urdu corpora; re-annotation or removal of contradictory labels with reported inter-annotator agreement; a wider learning-rate grid and more seeds; CPU and quantised-model latency tests; calibration methods for mmBERT-base; extension to Roman Urdu, code-mixed text, and multi-class categories; and detailed error analysis of false positives and false negatives.

Funding:

The authors received no funding for this work.

Conflict of Interest statement:

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability:

The dataset used in this study is publicly available through Hugging Face. It can be accessed at: https://huggingface.co/datasets/Adnan855570/Urdu_Hate_Speech

Author Contributions:

Maimoona Khilji: Led the research design, experiments, analysis, and methodology; produced key visualizations; authored core manuscript sections; interpreted results; and approved the final draft.

Haris Iqbal: Supported experiments, data preparation, methodology refinement, and manuscript drafting; prepared visuals; and approved the final draft.

Afsheen Khalid: Performed data analysis, provided domain insights, refined methodology and discussion, validated results, prepared visuals, and approved the final draft.

Dilawar Khan: Performed data analysis, provided domain insights, refined methodology and discussion, validated results, prepared visuals, and approved the final draft.

Muhammad Shahan Ibad: Performed data analysis, provided domain insights, refined methodology and discussion, validated results, prepared visuals, and approved the final draft.

References:

[1] M. S. I. Malik, A. Nawaz, and M. M. Jamjoom, "Hate Speech and Target Community Detection in Nastaliq Urdu Using Transfer Learning Techniques," *IEEE Access*, vol. 12, pp. 116875–116890, 2024, doi: 10.1109/ACCESS.2024.3444188.

[2] I. Hussain, M. M. A. Arshad, A. N. Cheema, and I. M. Almanjahie, "Enhancing Urdu hate speech detection through differential transfer learning and adaptive loss functions," *Sci. Reports 2025 151*, vol. 15, no. 1, pp. 37407-, Oct. 2025, doi: 10.1038/s41598-025-21306-w.

[3] M. S. Khan, M. S. I. Malik, and A. Nadeem, "Detection of violence incitation expressions in Urdu tweets using convolutional neural network," *Expert Syst. Appl.*, vol. 245, p. 123174, Jul. 2024, doi: 10.1016/J.ESWA.2024.123174.

[4] G. Musa Raza, A. Naseem, S. Iqbal, T. Masood, B. S. Kim, and T. N. Nguyen, "Intelligent model for target identification from Urdu threatening tweets in low-resource language using deep learning and fine-tuning of language models," *J. Inf. Telecommun.*, vol. 10, no. 3, pp. 383–401, Jul. 2026, doi: 10.1080/24751839.2025.2576951.

[5] J. A. Siddiqui, S. S. Yuhaniz, G. M. Shaikh, S. A. Soomro and Z. A. Mahar, "Fine-Grained Multilingual Hate Speech Detection Using Explainable AI and Transformers," *IEEE Access*, vol. 12, pp. 143177–143192, 2024, doi: 10.1109/ACCESS.2024.3470901.

[6] M. Ahmad, M. Waqas, A. Hamza, S. Usman, I. Batyrshin, and G. Sidorov, "UA-HSD-2025: Multi-Lingual Hate Speech Detection from Tweets Using Pre-Trained Transformers," *Comput. 2025, Vol. 14, Page 239*, vol. 14, no. 6, p. 239, Jun. 2025, doi: 10.3390/COMPUTERS14060239.

[7] T. Srikissoon and V. Marivate, "Combating Hate: How Multilingual Transformers Can Help Detect Topical Hate Speech," *Epic Ser. Comput.*, vol. 93, pp. 203–215, 2023, doi: 10.29007/1CM6.

[8] Muhammad Usman, Muhammad Ahmad, Moein Shahiki Tash, Irina Gelbukh, Rolando Quintero Tellez, Grigori Sidorov, "Multilingual Hate Speech Detection in Social Media Using Translation-Based Approaches with Large Language Models," *Prepr. Submitt. to Results Phys.*, 2025, [Online]. Available: https://arxiv.org/pdf/2506.08147

[9] A. Ullah *et al.*, "Threatening language detection from Urdu data with deep sequential model," *PLoS One*, vol. 19, no. 6, Jun. 2024, doi: 10.1371/JOURNAL.PONE.0290915.

[10] S. Ud Din, S. Khusro, F. Ali Khan, M. Ahmad, O. Ali, and T. M. Ghazal, "An Automatic Approach for the Identification of Offensive Language in Perso-Arabic Urdu Language: Dataset Creation and Evaluation," *IEEE Access*, vol. 13, pp. 19755–19769, 2025, doi: 10.1109/ACCESS.2025.3534662.

[11] A. Rashid, S. Mahmood, U. Inayat, and M. F. Zia, "Urdu Toxicity Detection: A Multi-Stage and Multi-Label Classification Approach," *AI 2025, Vol. 6, Page 194*, vol. 6, no. 8, p. 194, Aug. 2025, doi: 10.3390/AI6080194.

[12] E. Shoukat, R. Irfan, I. Basharat, M. A. Tahir, and S. Shaukat, "Attention based Bidirectional GRU hybrid model for inappropriate content detection in Urdu language," Jan. 2025, Accessed: Oct. 25, 2026. [Online]. Available: https://arxiv.org/pdf/2501.09722

[13] M. Ali, F. Sarwar, and A. Ashraf, "Hate Speech Detection Using Transformer-Based BERT model," *Gd. Asian J. Comput. Emerg. Technol.* , vol. 1, no. 1, pp. 12–19, Aug. 2025, Accessed: Oct. 25, 2026. [Online]. Available: https://gajcet.com/article/view/14

[14] Muhammad Bilal, Atif Khan, "Roman Urdu Hate Speech Detection Using Transformer-Based Model for Cyber Security Applications," *Sensors*, vol. 23, no. 8, p. 3909, 2023, doi: https://doi.org/10.3390/s23083909.

[15] F. Maqbool, B. Spahiu, and A. Maurino, "Impact of Data Augmentation on Hate Speech Detection in Roman Urdu," *CEUR Workshop Proc.*, vol. 3741, pp. 321–330, 2024, Accessed: Oct. 25, 2026. [Online]. Available: https://boa.unimib.it/handle/10281/490399

[16] M. Zain *et al.*, "RU-OLD: A Comprehensive Analysis of Offensive Language Detection in Roman Urdu Using Hybrid Machine Learning, Deep Learning, and Transformer Models," *Algorithms*, vol. 18, no. 7, Jul. 2025, doi: 10.3390/A18070396.

[17] "(PDF) Fine-Tuning Large Language Models with QLoRA for Offensive Language Detection in Roman Urdu-English Code-Mixed Text." Accessed: Oct. 25, 2026. [Online]. Available: https://www.researchgate.net/publication/396249813_Fine-Tuning_Large_Language_Models_with_QLoRA_for_Offensive_Language_Detection_in_Roman_Urdu-English_Code-Mixed_Text

[18] N. Hussain, A. Qasim, G. Mehak, O. Kolesnikova, A. Gelbukh, and G. Sidorov, "Hybrid Machine Learning and Deep Learning Approaches for Insult Detection in Roman Urdu Text," *AI 2025, Vol. 6, Page 33*, vol. 6, no. 2, p. 33, Feb. 2025, doi: 10.3390/AI6020033.

[19] M. O. Raza, A. Umar, and M. Awan, "Slur and Emoji Aware Models for Hate and Sentiment Detection in Roman Urdu Transgender Discourse," 2025. doi: 10.26615/978-954-452-100-4-014.

[20] T. Zubair, M. J. Asif, F. Kamiran, H. H. Saeed, and R. F. Ahmad, "Efficient Adaptation of LLMs for Hate Speech Detection in Low-Resource Languages: A

Comparative Study on Roman Urdu," Aug. 2026, Accessed: Oct. 25, 2026. [Online]. Available: https://arxiv.org/pdf/2608.18142

[21] K. Ullah, M. Aslam, M. U. G. Khan, F. S. Alamri, and A. R. Khan, "UEF-HOCUrdu: Unified Embeddings Ensemble Framework for Hate and Offensive Text Classification in Urdu," *IEEE Access*, vol. 13, pp. 21853–21869, 2025, doi: 10.1109/ACCESS.2025.3532611.

[22] S. Harris, H. J. Hadi, N. Ahmad, and M. A. Alshara, "Multi-domain Urdu fake news detection using pre-trained ensemble model," *Sci. Reports 2025 151*, vol. 15, no. 1, pp. 8705-, Mar. 2025, doi: 10.1038/s41598-025-91054-4.

[23] "Sentiment Analysis of Urdu Text using Hybrid Deep Learning Techniques | Journal of Computing & Biomedical Informatics." Accessed: Oct. 25, 2026. [Online]. Available: https://www.jcbi.org/index.php/Main/article/view/962

[24] M. S. I. Malik, A. Nazarova, and M. M. Jamjoom, "Explainable threatening tweets identification using abstract meaning representation graph and pattern structure approaches," *Intell. Data Anal.*, vol. 29, no. 6, pp. 1501–1519, Nov. 2025, doi: 10.1177/1088467X251330977.

[25] A. Mehmood, M. T. Hassan, and Q. Baig, "Analyzing and Visualizing Urdu News Articles Using NLP and Concept Graphs," *2025 6th Int. Conf. Adv. Comput. Sci. ICACS 2025*, 2025, doi: 10.1109/ICACS64902.2025.10937859.

[26] "Memorization vs. Generalization : Quantifying Data Leakage in NLP Performance Evaluation | Request PDF." Accessed: Jun. 20, 2026. [Online]. Available: https://www.researchgate.net/publication/355429440_Memorization_vs_Generalization_Quantifying_Data_Leakage_in_NLP_Performance_Evaluation

[27] K. Lee *et al.*, "Deduplicating Training Data Makes Language Models Better," *Proc. Annu. Meet. Assoc. Comput. Linguist.*, vol. 1, pp. 8424–8445, Jul. 2021, doi: 10.18653/v1/2022.acl-long.577.

[28] E. Leonardelli, S. Menini, A. P. Aprosio, M. Guerini, and S. Tonelli, "Agreeing to Disagree: Annotating Offensive Language Datasets with Annotators' Disagreement," *EMNLP 2021 - 2021 Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 10528–10539, Sep. 2021, doi: 10.18653/v1/2021.emnlp-main.822.

[29] M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, and B. Van Durme, "mmBERT: A Modern Multilingual Encoder with Annealed Language Learning," Sep. 2025, Accessed: Oct. 25, 2026. [Online]. Available: https://arxiv.org/pdf/2509.06888

[30] "Adnan855570/Urdu_Hate_Speech · Datasets at Hugging Face." Accessed: Oct. 25, 2026. [Online]. Available: https://huggingface.co/datasets/Adnan855570/Urdu_Hate_Speech

[31] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," *Proc. Annu. Meet. Assoc. Comput. Linguist.*, pp. 8440–8451, 2020, doi: 10.18653/V1/2020.ACL-MAIN.747.

[32] "mirfan899/urdu-bert-ner · Hugging Face." Accessed: Oct. 25, 2026. [Online]. Available: https://huggingface.co/mirfan899/urdu-bert-ner

[33] "urduhack/roberta-urdu-small · Hugging Face." Accessed: Oct. 25, 2026. [Online]. Available: https://huggingface.co/urduhack/roberta-urdu-small

Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.