

## BRAIN. Broad Research in Artificial Intelligence and Neuroscience

e-ISSN: 2067-3957 | p-ISSN: 2068-0473

Covered in: Web of Science (ESCI); EBSCO; JERIH PLUS (hkdir.no); IndexCopernicus; Google Scholar; SHERPA/RoMEO; ArticleReach Direct; WorldCat; CrossRef; Peeref; Bridge of Knowledge (mostwiedzy.pl); abcdindex.com; Editage; Ingenta Connect Publication; OALib; scite.ai; Scholar9; Scientific and Technical Information Portal; FID Move; ADVANCED SCIENCES INDEX (European Science Evaluation Centre, neredataltics.org); ivySCI; exaly.com; Journal Selector Tool (letpub.com); Citefactor.org; fatcat!; ZDB catalogue; Catalogue SUDOC (abes.fr); OpenAlex; Wikidata; The ISSN Portal; Socolar; KVK-Volltitel (kit.edu) 2026, Volume 17, Issue 3, pages: 43-57

Submitted: May 15<sup>th</sup>, 2026 | Accepted for publication: July 21<sup>st</sup>, 2026

### PSO-Based Hybrid Lexical-Semantic Feature Selection and Ensemble Learning Framework for Urdu Hate Speech Detection

#### Haseeb Ullah

Gomal Research Institute of Computing (GRIC),  
Faculty of Computing, Gomal University, Dera Ismail  
Khan, Khyber Pakhtunkhwa, Pakistan.  
h2seeb@gmail.com,  
<https://orcid.org/0009-0003-4259-7298>

#### Husnain Saleem\*

Gomal Research Institute of Computing (GRIC),  
Faculty of Computing, Gomal University, Dera Ismail  
Khan, Khyber Pakhtunkhwa, Pakistan.  
jilani.husnain@yahoo.com,  
<https://orcid.org/0009-0001-7513-1086>

#### Asia Kanwal

Department of Artificial Intelligence, University of  
Agriculture, Dera Ismail Khan, Khyber Pakhtunkhwa,  
Pakistan. aassia.baloch@gmail.com,  
<https://orcid.org/0009-0004-2548-5582>

#### Muhammad Javed

Gomal Research Institute of Computing (GRIC),  
Faculty of Computing, Gomal University, Dera Ismail  
Khan, Khyber Pakhtunkhwa, Pakistan.  
javed\_gomal@gu.edu.pk,  
<https://orcid.org/0000-0001-6884-6641>

#### Hamid Masood Khan

Gomal Research Institute of Computing (GRIC),  
Faculty of Computing, Gomal University, Dera Ismail  
Khan, Khyber Pakhtunkhwa, Pakistan.  
hamidmasoodkhan@gu.edu.pk,  
<https://orcid.org/0000-0001-9235-9886>

#### Kiran Hanif

Gomal Research Institute of Computing (GRIC), Faculty of Computing, Gomal University, Dera Ismail Khan, Khyber  
Pakhtunkhwa, Pakistan.  
safdarkiran3@gmail.com,  
<https://orcid.org/0009-0000-7573-7515>

\* Corresponding author

**Abstract:** Automated Urdu hate-speech detection remains challenging because annotated resources are limited, orthography varies, and harmful meaning depends on both lexical and contextual cues. This study evaluates a leakage-safe lexical-semantic framework combining 5,000-dimensional word/bigram TF-IDF features with 768-dimensional multilingual sentence embeddings, followed by binary particle swarm optimisation (BPSO) and classical ensemble learning. Experiments used the Urdu text and binary labels from MMHS11K: 8,800 balanced training records and an untouched balanced test set of 2,200 records. BPSO selected 2,849 of 5,768 hybrid dimensions, reducing dimensionality by 50.61%. On the official test set, Stacking with the complete hybrid representation achieved Macro-F1 = 0.8468 and ROC-AUC = 0.9265; the PSO-selected representation achieved Macro-F1 = 0.8391 and ROC-AUC = 0.9196. After mask selection, classifier fitting time decreased by 52.24%, excluding sentence-embedding extraction and BPSO search. Leakage-safe nested five-fold validation showed a similar trade-off: fitting time decreased by 52.64%, with Macro-F1 =  $0.8444 \pm 0.0066$  after PSO versus  $0.8492 \pm 0.0110$  without PSO. Soft Voting and Stacking were statistically comparable on selected features. Three-seed transformer aggregates performed better: Urdu-RoBERTa achieved the highest Macro-F1 (0.8836), and XLM-R achieved the highest ROC-AUC (0.9529). Holm-corrected exact McNemar tests confirmed that transformer aggregates significantly outperformed PSO Stacking. Overall, BPSO approximately halves representation size and downstream classical-model fitting time, with a small predictive decrease that was not statistically significant relative to complete-hybrid Stacking after correction.

**Keywords:** Urdu hate speech; low-resource NLP; binary particle swarm optimisation; TF-IDF; sentence embeddings; ensemble learning; feature selection.

**How to cite:** Ullah, A. H., Saleem, H., Kanwal, A., Javed, M., Khan, H. M., & Hanif, K. (2026). PSO-based hybrid lexical-semantic feature selection and ensemble learning framework for Urdu hate speech detection. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 17(3), 43-57. <https://doi.org/10.70594/brain/17.3/3>

## 1. Introduction

Online platforms enable rapid public communication but also facilitate abusive, discriminatory, and hateful content. Hate-speech detection is not a simple keyword-matching problem: the same token can be benign, quoted, sarcastic, reclaimed, or targeted depending on context. These ambiguities are particularly consequential in low-resource languages, where annotated corpora, normalisation resources, tokenizers, and benchmark protocols are less mature than in English (Fortuna & Nunes, 2018; Poletto et al., 2021; Schmidt & Wiegand, 2017; Vidgen & Derczynski, 2020).

Urdu is written predominantly in a Perso-Arabic script and exhibits character variants, optional diacritics, borrowed vocabulary, spelling variation, and code mixing. Research has addressed Roman Urdu hate and offensive language (Rizwan et al., 2020), Urdu tweet classification using feature-selection and sentiment-oriented pipelines (Ali et al., 2021), Urdu offensive-language identification using semantic and embedding models (Hussain et al., 2022), transfer learning for Urdu hate speech (Arshad et al., 2023), and Nastaliq Urdu hate-speech and target-community detection (Malik et al., 2024). Recent work has also introduced the multilingual, multimodal MMHS11K resource (Saddozai et al., 2025). Collectively, these studies show that Urdu hate-speech detection has been addressed through lexicons, conventional machine learning, deep learning, and transformers. The remaining gap is therefore not the absence of Urdu research, but the limited evidence on how a reproducible hybrid lexical-semantic representation behaves under controlled ablations, binary feature selection, leakage-safe cross-validation, strong transformer baselines, and paired statistical testing.

Feature selection is attractive for sparse text representations because it can reduce redundancy and model-fitting cost. However, a smaller representation should not automatically be described as more accurate or superior. The relevant question is whether dimensionality reduction preserves enough predictive information to justify the resulting performance trade-off. BPSO is suitable for this question because each particle can encode a binary feature mask, while a fitness function can jointly penalise validation error and selected-feature count (Kennedy & Eberhart, 1995, 1997).

This study addresses four research questions:

1. How do TF-IDF, multilingual sentence embeddings, their unselected hybrid, and their PSO-selected hybrid compare on an untouched official test set?
2. Does BPSO reduce representation size and model-fitting time while retaining competitive predictive performance?
3. Are differences among classical configurations statistically supported under paired tests and leakage-safe nested validation?
4. How does the best classical feature-selection pipeline compare with three-seed mBERT, XLM-R, and Urdu-RoBERTa baselines?

The contributions are: (i) a reproducible Urdu-specific preprocessing and hybrid feature pipeline; (ii) a fully specified BPSO objective and search configuration; (iii) controlled feature and classifier ablations, nested five-fold validation, bootstrap confidence intervals, and Holm-corrected McNemar tests; and (iv) a transparent finding that PSO offers a substantial reduction in dimensionality and fitting time, although complete hybrid features and transformer models achieve higher predictive scores.

## 2. Related Work

### 2.1 Hate-Speech Detection in Urdu and Related Varieties

Rizwan et al. (2020) released RUHSOLD for Roman Urdu and evaluated lexical, embedding, and neural baselines. Ali et al. (2021) studied Urdu tweets using sentiment information, stopword filtering, global feature selection, and imbalance handling. Hussain et al. (2022) investigated offensive-language identification in Urdu using semantic and embedding

representations, demonstrating that feature selection and contextual information had already been explored in closely related Urdu tasks. Arshad et al. (2023) developed a transfer-learning approach for Urdu hate-speech, while Malik et al. (2024) considered Nastaliq Urdu hate-speech together with target-community identification. Saddozai et al. (2025) introduced MMHS11K as a balanced Urdu-English multimodal dataset and evaluated text, image, and fusion models.

The studies summarised in Table 1 differ in script, data source, label taxonomy, modality, and evaluation protocol; their reported scores are therefore not directly comparable. The table is used to position datasets, methods, and study-specific results, not to assert a universal ranking.

*Table 1. Positioning relative to representative Urdu and Roman Urdu studies.*

| Study           | Dataset/task                                                                      | Method                                                                                     | Author-reported result and relevance                                                                                                                  |
|-----------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Rizwan et al.   | RUHSOLD; 10,012 Roman Urdu tweets; coarse- and fine-grained hate/offensive labels | Domain-specific embeddings, transfer learning, and CNN-gram                                | CNN-gram was reported as more robust than seven baselines; the corpus uses Roman Urdu and a different taxonomy.                                       |
| Ali et al.      | Urdu Twitter corpus; sentiment/aspect-based hate classification                   | Dynamic stopword filtering, VGFSS, SMOTE, SVM, and MNB                                     | The best reported SVM micro-F1 was 0.927 after feature selection; the protocol and label construction differ from MMHS11K.                            |
| Hussain et al.  | Urdu offensive-language identification                                            | Semantic embedding representations with feature selection                                  | Demonstrated feature-selection effects in a closely related Urdu task; this evidence requires moderation of any first-use claim.                      |
| Arshad et al.   | UHated; 7,800 annotated Urdu tweets; multi-class hate classification              | FastText and multilingual RoBERTa transfer learning                                        | RoBERTa achieved author-reported macro-F1 = 0.82; the corpus and multi-class protocol differ from the present binary split.                           |
| Malik et al.    | HSTC Facebook corpus; Nastaliq Urdu hate and target-community tasks               | Urdu-RoBERTa and Urdu-DistilBERT fine-tuning                                               | Best reported binary F1 = 86.52% and target-community F1 = 83.91%; a different corpus and multi-task setting were used.                               |
| Saddozai et al. | MMHS11K; 11,000 Urdu-English multimodal records                                   | BiLSTM text branch, EfficientNet image branch, and early fusion                            | Reported Urdu F1 = 81.2% and English F1 = 75.5%; the present study isolates only the Urdu text stream.                                                |
| Present study   | MMHS11K Urdu text; official 8,800/2,200 binary split                              | TF-IDF + sentence embeddings + BPSO; seven classical models and three transformer families | PSO Stacking official-test macro-F1 = 0.8391; Urdu-RoBERTa aggregate macro-F1 = 0.8836; includes ablations, nested validation, and paired statistics. |

## 2.2. Hybrid Representations and Feature Selection

TF-IDF remains effective for hate-speech detection because explicit insults, group references, slogans, and recurrent n-grams are often discriminative. Dense sentence embeddings can complement these sparse lexical signals by encoding contextual similarity and paraphrastic information (Reimers & Gurevych, 2019). Concatenating sparse and dense representations may improve coverage, but it also increases dimensionality and can introduce redundant or weakly informative variables. Feature selection can therefore be useful even when the predictive gain is modest.

Particle swarm optimisation was originally proposed for continuous search (Kennedy & Eberhart, 1995) and later adapted to binary spaces (Kennedy & Eberhart, 1997). In text classification, each particle can encode a subset of the vocabulary and embedding dimensions. Unlike filter methods that rank variables independently, wrapper-based PSO evaluates masks through a downstream validation classifier. This flexibility comes at a substantial search cost and creates a risk of optimistic estimates unless feature selection is confined to the training data and repeated within cross-validation folds.

### 2.3. Ensemble and Transformer Baselines

Voting combines model probabilities, while Stacking learns a meta-classifier from base-model outputs. Bagging and random forests reduce variance through resampling and tree aggregation, and gradient-boosted trees model nonlinear interactions (Chen & Guestrin, 2016; Wolpert, 1992). For multilingual text, mBERT and XLM-R provide widely used contextual baselines (Conneau et al., 2020; Devlin et al., 2019). A language-specific Urdu RoBERTa model offers a complementary comparison because its tokenizer and pretraining corpus are targeted more closely tailored to Urdu. The present experiments include both classical and transformer families so that PSO is assessed as a feature-reduction method rather than being presented as a replacement for modern contextual language models.

## 3. Materials and Methods

### 3.1. Dataset, Provenance, and Study Scope

The experiments used MMHS11K, a publicly released dataset of 11,000 Urdu tweets with English translations, image-derived information, and binary Hate/No-Hate labels (Saddozai et al., 2025). The released split contains 8,800 training records (4,400 Hate and 4,400 No-Hate) and 2,200 test records (1,100 per class). As reported by the source authors, annotation guidelines were developed iteratively with Urdu-domain experts, and inter-annotator agreement for the final corpus reached Fleiss’  $\kappa = 0.816$  (Saddozai et al., 2025).

The present study is deliberately text-only. It uses the Urdu Text field and the binary Label; images, OCR/image-text fields, and English translations are excluded. This restriction supports a controlled analysis of lexical-semantic text representations but does not reproduce the multimodal setting for which MMHS11K was originally designed.

An audit performed before modelling found no missing records after filtering, no exact within-split duplicates, and no train-test record overlap. Stable record identifiers and SHA-256 text hashes were retained for every prediction file. The official test set was never used to learn the TF-IDF vocabulary or IDF weights, scale embeddings, select features, choose models, tune thresholds, or stop training.

Table 2. Dataset composition after verification.

| Split             | Records | Hate  | No-Hate | Exact duplicates | Cross-split overlap |
|-------------------|---------|-------|---------|------------------|---------------------|
| Official training | 8,800   | 4,400 | 4,400   | 0                | 0                   |
| Official test     | 2,200   | 1,100 | 1,100   | 0                | 0                   |

Because hate labels may expose readers to abusive language, examples are paraphrased, and slurs are redacted. A Hate case can contain a direct dehumanising insult or a threat aimed at a person or group; a No-Hate case may report a civic issue, public service, or non-targeted criticism without attacking a protected or social group.

Mentions and URLs were removed before modelling. Annotation remains culturally sensitive because criticism, quoted speech, sarcasm, political rhetoric, and threats can be difficult to distinguish without broader conversational or visual context.

### 3.2. Urdu Preprocessing

Preprocessing was defined before fitting and applied identically to the training and test text. URLs and user mentions were removed. The # marker was deleted while the hashtag lexical content was retained. Arabic/Persian glyph variants were mapped to canonical Urdu code points: yeh variants U+064A, U+0649, and U+0626 to U+06CC; Arabic kaf U+0643 to keheh U+06A9; heh/teh-marbuta variants U+06C0 and U+0629 to heh goal U+06C1; and waw with hamza U+0624 to waw U+0648.

Diacritics and tatweel were removed. Punctuation was filtered for the classical pipeline, whitespace was normalised, and a fixed list of 26 frequent Urdu function words was removed. No stemming or lemmatisation was used. Classical features used whitespace-separated tokens after cleaning; transformer baselines used checkpoint-specific subword tokenizers on normalised text.

The fixed stopword inventory contained the following 26 items:

اس، ان، اور، ایک، بھی، تو، تھا، تھی، تھے، جو، سے، میں، نہیں، نے، وہ، پر، کا، کر، کو، کی، کیا، کے، ہی، ہیں، ہے،  
یہ۔

No training or test record became empty after preprocessing.

### 3.3. Lexical and Semantic Representations

#### 3.3.1. TF-IDF Features

Word unigrams and bigrams were represented using smoothed IDF, sublinear term frequency, a minimum document frequency of 2, and a maximum vocabulary of 5,000 features. For term  $t$  in document  $d$ ,

$$tfidf(t, d) = \left(1 + \log f_{t,d}\right) \left[\log\left(\frac{N+1}{df(t)+1}\right) + 1\right] \quad (1)$$

where  $f_{t,d}$  is the within-document count,  $N$  is the number of training documents, and  $df(t)$  is the document frequency. The vocabulary was learned from the training data only.

For the full-budget BPSO search, the TF-IDF vocabulary was learned only on the 80% PSO-core subset and then frozen. After mask selection, the fixed vocabulary was retained while its IDF weights were refitted on all 8,800 official training records before the official test set was transformed. No test record influenced the vocabulary or IDF estimation.

#### 3.3.2. Sentence Embeddings and Hybrid Representation

Dense semantic vectors were generated with sentence-transformers/use-cmlm-multilingual at immutable revision 736a29561b4df04fc2d03bc7f6a574e4d0081b56.

The encoder produces 768-dimensional multilingual sentence embeddings and was retained because it was the exact model specified in the original submission.

Let  $t_i \in R^{5000}$  denote the TF-IDF vector and  $e_i \in R^{768}$  denote the sentence embedding for record  $i$ . Their concatenation is given by

$$x_i = [t_i \parallel e_i] \in R^{5768} \quad (2)$$

For the BPSO search, a StandardScaler was fitted only on the PSO-core embedding vectors and applied unchanged to the held-out PSO validation vectors. After mask selection, a new scaler was fitted to all 8,800 official training embeddings and applied unchanged to the official test embeddings.

Thus, all scaling parameters were estimated solely from the training records available at each stage.

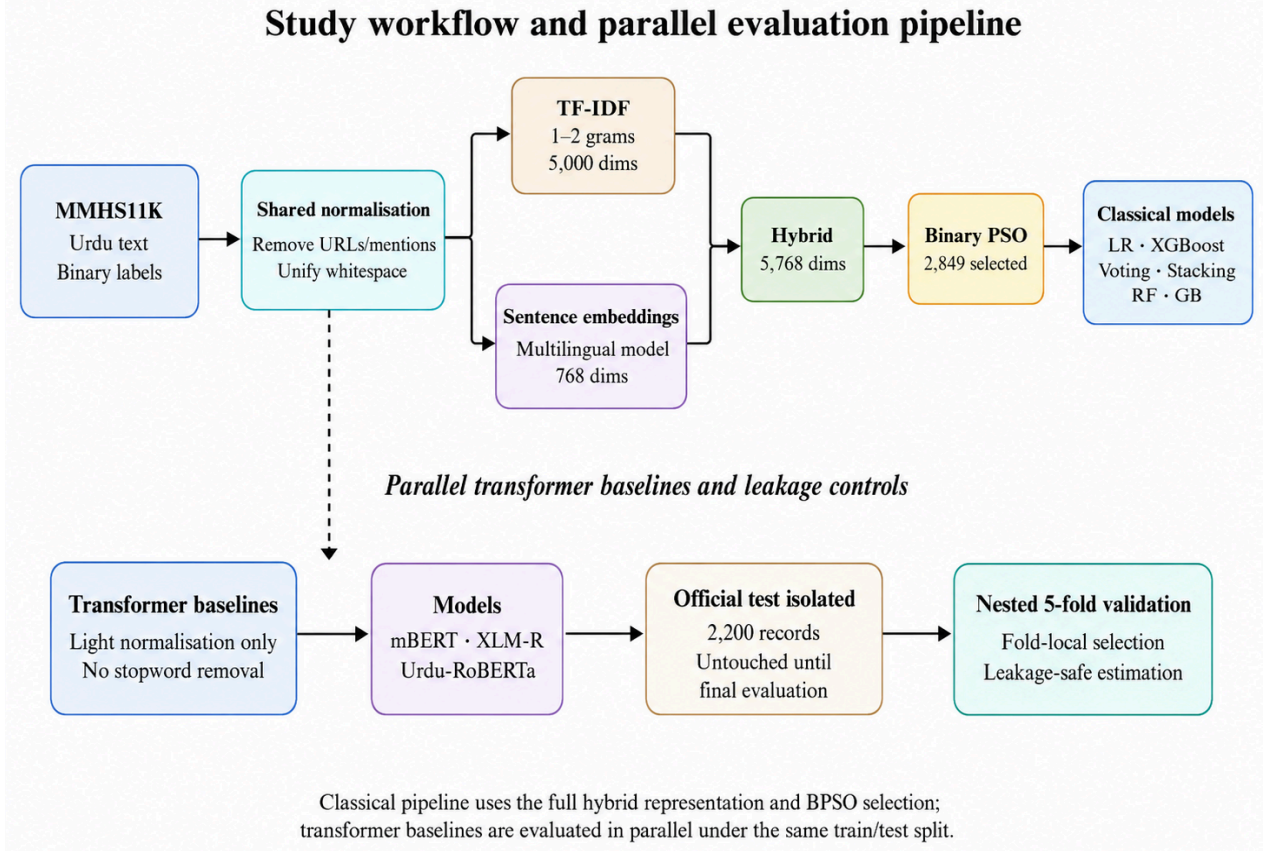


Figure 1. Study workflow. The classical branch removed punctuation and 26 stopwords, whereas transformer baselines used normalised text without stopwords removal. The official test set remained isolated from representation learning, BPSO, model selection, thresholding, and early stopping.

### 3.4. Binary Particle Swarm Optimisation

Each particle encoded a binary mask  $z_i^{(k)} \in \{0, 1\}^D$  for  $D = 5,768$  features at iteration  $k$ . Velocity was updated as

$$v_{ij}^{(k+1)} = \text{clip}\left(wv_{ij}^{(k)} + c_1 r_{1,ij} (p_{ij} - z_{ij}^{(k)}) + c_2 r_{2,ij} (g_j - z_{ij}^{(k)}), v_{\min}, v_{\max}\right) \quad (3)$$

where  $p_{ij}$  and  $g_j$  are the particle-best and global-best positions,  $r_1, r_2 \sim U(0, 1)$ ,  $w$  is inertia weight, and  $c_1, c_2$  are the cognitive and social coefficients. Continuous velocities were mapped to selection probabilities through

$$S(v) = \frac{1}{1 + e^{-v}} \quad (4)$$

and binary positions were sampled by

$$z_{ij}^{(k+1)} = I\left[u_{ij} < S\left(v_{ij}^{(k+1)}\right)\right], \quad u_{ij} \sim U(0, 1) \quad (5)$$

The wrapper fitness function minimised validation error and selected-feature proportion:

$$J(z) = \{\alpha[1 - \text{MacroF1}_{\text{val}}(z)] + (1 - \alpha)\frac{\|z\|_0}{D}, \quad \|z\|_0 \geq q_{\min}, \quad + \infty, \text{ otherwise}, \quad (6)$$

with  $\alpha = 0.99$  and  $q_{\min} = 10$ . The fitness was estimated using `SGDClassifier(loss="log_loss")` on a stratified 80:20 split drawn only from the official training set.

Table 3. Full-budget BPSO configuration.

| Parameter               | Value          | Parameter                 | Value                 |
|-------------------------|----------------|---------------------------|-----------------------|
| Particles               | 40             | Maximum iterations        | 100                   |
| Random seed             | 42             | Validation fraction       | 0.20                  |
| Inertia                 | 0.70           | Cognitive coefficient     | 1.50                  |
| Social coefficient      | 2.00           | Velocity bounds           | [-4, 4]               |
| Position initialisation | Bernoulli(0.5) | Velocity initialisation   | Uniform within bounds |
| Transfer function       | Sigmoid        | Minimum selected features | 10                    |
| Performance weight      | 0.99           | Feature penalty           | 0.01                  |
| Early-stopping patience | 20 iterations  | Fitness classifier        | Logistic-loss SGD     |

### 3.5. Classical Classifiers and Ensembles

Seven classifiers were evaluated on the fixed PSO-selected representation: Logistic Regression, XGBoost, Soft Voting, Stacking, Bagging, Random Forest, and Gradient Boosting. Soft Voting averaged posterior probabilities from its component models:

$$\hat{y}_i = I \left[ \frac{1}{M} \sum_{m=1}^M p_m(y_i = 1|x_i) \geq 0.5 \right] \quad (7)$$

Stacking used out-of-fold base-model probabilities as inputs to a logistic meta-classifier:

$$\hat{y}_i = h(p_1(1|x_i), ..., p_M(1|x_i)) \quad (8)$$

where  $h(\cdot)$  is the meta-classifier.

The same Stacking specification, including the Logistic Regression and XGBoost base learners and the Logistic Regression meta-classifier, was used unchanged across all four feature-ablation experiments so that observed differences could be attributed to the representation rather than to changes in classifier identity or configuration.

Implementation settings were fixed before the official test evaluation. Logistic Regression used  $C = 1.0$ , the liblinear solver, and 2,000 maximum iterations. XGBoost used 300 trees, maximum depth 5, learning rate 0.05, subsample = 0.80, column subsample = 0.80, minimum child weight = 1, L2 regularisation = 1.0, and the histogram tree method. Soft Voting combined Logistic Regression and XGBoost with equal weights. Stacking used the same two base learners, five-fold out-of-fold probability features, a Logistic Regression meta-classifier, and no passthrough of the original features. Bagging used 20 Logistic Regression estimators with 80% bootstrap samples. Random Forest used 500 trees and sqrt feature sampling. Gradient Boosting used 200 estimators, a learning rate of 0.05, and a tree depth of 3. Seed 42 was used for classical-model reproducibility.

### 3.6. Transformer Baselines

Three contextual baselines were fine-tuned: google-bert/bert-base-multilingual-cased (mBERT), FacebookAI/xlm-roberta-base (XLM-R), and urduhack/roberta-urdu-small (Urdu-RoBERTa). Immutable model revisions were recorded. Each model was trained for three epochs with a maximum sequence length of 128 using seeds 42, 52, and 62. An internal 7,920/880 train-validation split was generated independently for each seed from the official training data. The official test set was evaluated only after training. In addition to the mean  $\pm$  standard deviation across seeds, a family-level aggregate was obtained by averaging the three test probabilities before thresholding at 0.5.

Fine-tuning used a training batch size of 16, an evaluation batch size of 32, a learning rate of  $2 \times 10^{-5}$ , weight decay of 0.01, a warmup ratio of 0.10, dynamic padding, and mixed-precision training on a GPU. Models were evaluated once per epoch. For XLM-R and Urdu-RoBERTa, the checkpoint with the highest validation Macro-F1 was restored. To resolve the earlier mBERT checkpoint-reloading ambiguity, mBERT was rerun under a compatible Transformers environment. During this compatibility rerun, the best validation state was retained in CPU memory and restored with strict key matching, thereby avoiding checkpoint serialisation. Early-stopping patience was 2. The validation fraction was 10% of the official training split for each seed. Transformer text used checkpoint-specific subword tokenisation and the lighter normalisation stream without stopword removal, stemming, or lemmatisation.

### 3.7. Evaluation and Statistical Analysis

Accuracy, Hate-class precision, Hate-class recall, Hate-class F1, Macro-F1, Matthews correlation coefficient (MCC), and ROC-AUC were reported. For the Hate class,

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F_1 = \frac{2PR}{P+R} \quad (9)$$

Binary Macro-F1 was the unweighted mean of class-specific F1 scores.

The primary official-test analysis used fixed predictions for all 2,200 records. Nonparametric bootstrap resampling with 1,000 repetitions produced 95% confidence intervals (Efron & Tibshirani, 1993).

Pairwise differences in correctness were assessed using two-sided exact McNemar tests. Because multiple models were compared with the PSO Stacking reference, p-values were adjusted using Holm’s sequential procedure (Holm, 1979; McNemar, 1947). Statistical significance was set at adjusted  $p < 0.05$ .

Leakage-safe nested five-fold validation compared Stacking with and without PSO. TF-IDF fitting, embedding scaling, PSO selection, and model fitting were repeated within each outer fold. To control computational burden, fold-local PSO used 10 particles, at most 20 iterations, and patience 8; these settings are reported separately from the full-budget official-test experiment.

All primary experiments were executed on a Tesla T4 GPU under Python 3.12.13. Key main-workflow library versions were scikit-learn 1.6.1, XGBoost 3.3.0, PyTorch 2.11.0, Transformers 5.14.1, and sentence-transformers 5.6.0 (Pedregosa et al., 2011; Wolf et al., 2020).

The targeted mBERT compatibility rerun used the same GPU, Python and PyTorch versions, identical ordered train/test hashes and seeds, and Transformers 4.57.1. Its best validation weights were retained in CPU memory and restored with strict=True, so no checkpoint serialisation was used.

Reported fit times measure the relevant model-fitting call after the representation or feature mask was available; they exclude data loading, text preprocessing, sentence-embedding extraction, and the BPSO search. Consequently, the timing results quantify downstream fitting savings rather than end-to-end wall-clock efficiency. Because the clean mBERT rerun used a different compatible Transformers version, between-family transformer fit-time comparisons are descriptive.

## 4. Results

### 4.1. PSO Convergence and Selected Representation

The full-budget search stopped after 36 iterations because the global best did not improve for 20 consecutive iterations.

The best internal validation Macro-F1 was 0.8341, and the final fitness loss was 0.1692. PSO selected 2,849 of 5,768 dimensions: 2,469 TF-IDF features and 380 embedding dimensions.

The resulting 50.61% dimensionality reduction is shown in Figure 2.

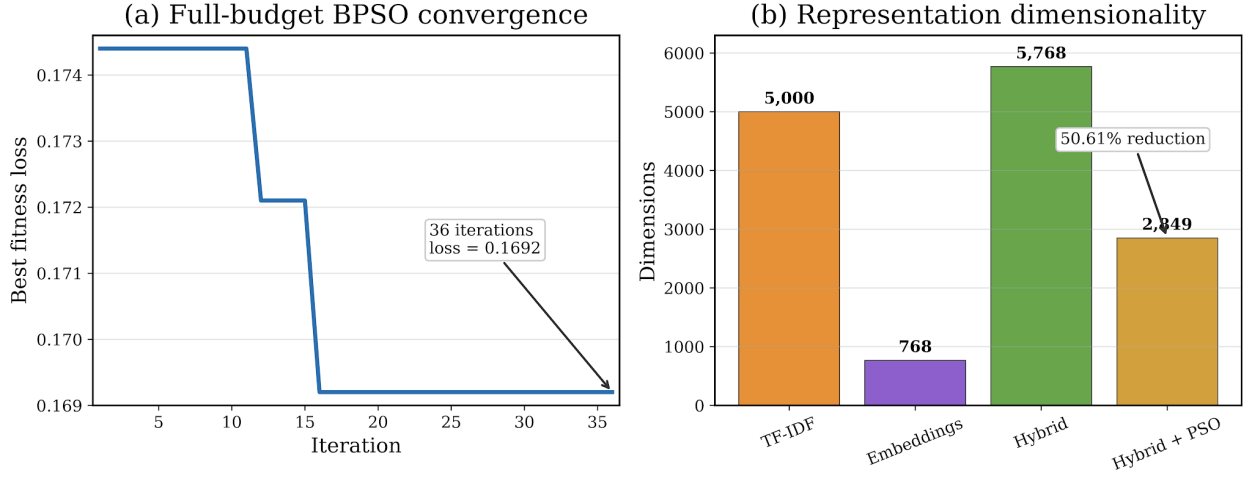


Figure 2. (a) Full-budget BPSO convergence showing the best fitness loss over iterations. (b) Representation dimensionality before and after PSO feature selection.

#### 4.2. Feature Ablation

Table 4. Official-test feature ablation using the same Stacking classifier.

| Representation           | Dimensions | Accuracy | Hate F1 | Macro-F1 | MCC    | ROC-AUC | Fit time (s) |
|--------------------------|------------|----------|---------|----------|--------|---------|--------------|
| TF-IDF only              | 5,000      | 0.8445   | 0.8440  | 0.8445   | 0.6891 | 0.9239  | 75.3         |
| Sentence embeddings only | 768        | 0.8359   | 0.8369  | 0.8359   | 0.6719 | 0.9199  | 299.6        |
| Hybrid without PSO       | 5,768      | 0.8468   | 0.8479  | 0.8468   | 0.6937 | 0.9265  | 406.4        |
| Hybrid with PSO          | 2,849      | 0.8391   | 0.8394  | 0.8391   | 0.6782 | 0.9196  | 194.1        |

Bootstrap 95% intervals for Macro-F1 were 0.8318–0.8609 for the complete hybrid and 0.8222–0.8546 for the PSO-selected hybrid. Figure 3 visualises the predictive and post-selection fitting-time trade-off; the displayed times exclude the BPSO search and representation-extraction stages.

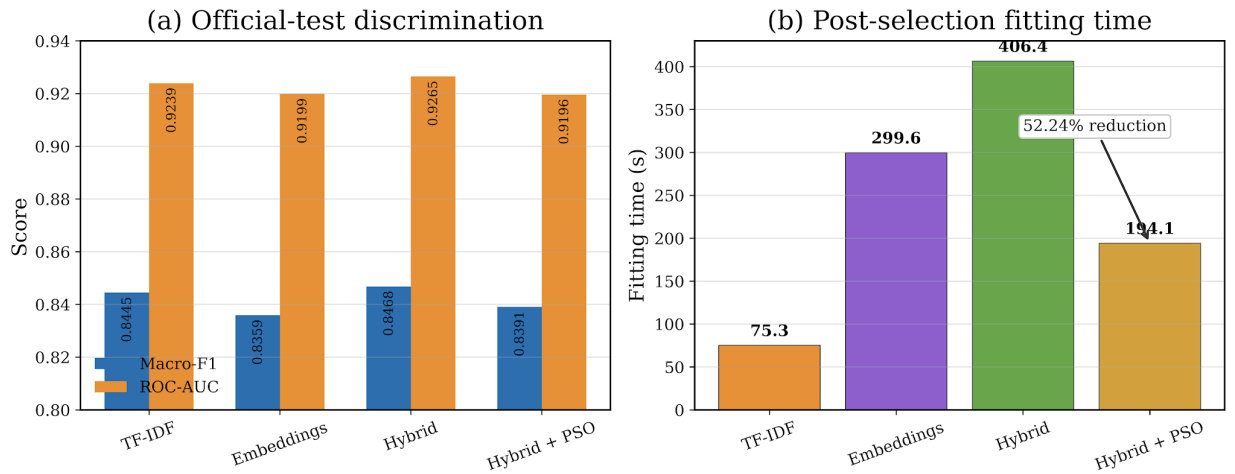


Figure 3. (a) Official-test predictive performance (Macro-F1 and ROC-AUC) across feature representations. (b) Post-selection classifier fitting times. Timing excludes sentence-embedding extraction and the BPSO search.

#### 4.3. Classical Classifier Comparison on PSO-Selected Features

Soft Voting achieved the highest Macro-F1 among models using the PSO mask (0.8400), while Stacking had a marginally higher ROC-AUC (0.9196 versus 0.9194). Their predictions differed on only four test records, and the Holm-adjusted McNemar p-value was 1.000;

consequently, the two ensembles should be interpreted as statistically comparable rather than ranked as substantively different.

*Table 5. Official-test Classifier Comparison on the 2,849-Dimensional PSO Representation*

| Model               | Accuracy | Precision | Recall | Hate F1 | Macro-F1 | ROC-AUC | Fit time (s) |
|---------------------|----------|-----------|--------|---------|----------|---------|--------------|
| Logistic Regression | 0.8359   | 0.8332    | 0.8400 | 0.8366  | 0.8359   | 0.9164  | 5.9          |
| XGBoost             | 0.8236   | 0.8173    | 0.8336 | 0.8254  | 0.8236   | 0.9054  | 33.2         |
| Soft Voting         | 0.8400   | 0.8375    | 0.8436 | 0.8406  | 0.8400   | 0.9194  | 38.5         |
| Stacking            | 0.8391   | 0.8379    | 0.8409 | 0.8394  | 0.8391   | 0.9196  | 190.8        |
| Bagging             | 0.8286   | 0.8231    | 0.8373 | 0.8301  | 0.8286   | 0.9121  | 41.9         |
| Random Forest       | 0.8245   | 0.8115    | 0.8455 | 0.8281  | 0.8245   | 0.9006  | 99.6         |
| Gradient Boosting   | 0.8027   | 0.7937    | 0.8182 | 0.8057  | 0.8027   | 0.8889  | 534.7        |

*Note.* The 194.1 s value in Table 4 and the 190.8 s value in Table 5 arose from separate timed fits of the same Stacking specification in the ablation and classifier-comparison workflows; the predictions and performance metrics were identical.

#### 4.4. Nested Five-Fold Validation

Nested validation reproduced the official test trade-off. Without PSO, the mean Macro-F1 was  $0.8492 \pm 0.0110$ . With fold-local PSO, the mean Macro-F1 was  $0.8444 \pm 0.0066$ , and the average representation contained  $2,873 \pm 35$  dimensions. Mean post-selection model-fitting time fell from 360.3 to 170.6 seconds, a 52.64% reduction; fold-local PSO search time and feature extraction were not included in this comparison.

*Table 6. Leakage-safe nested five-fold validation results*

| Configuration      | Accuracy            | Hate F1             | Macro-F1            | MCC                 | ROC-AUC             | Selected dims  | Fit time (s)    |
|--------------------|---------------------|---------------------|---------------------|---------------------|---------------------|----------------|-----------------|
| Hybrid without PSO | $0.8492 \pm 0.0110$ | $0.8497 \pm 0.0115$ | $0.8492 \pm 0.0110$ | $0.6985 \pm 0.0221$ | $0.9236 \pm 0.0057$ | 5,768          | $360.3 \pm 2.3$ |
| Hybrid with PSO    | $0.8444 \pm 0.0066$ | $0.8449 \pm 0.0069$ | $0.8444 \pm 0.0066$ | $0.6889 \pm 0.0133$ | $0.9188 \pm 0.0065$ | $2,873 \pm 35$ | $170.6 \pm 4.2$ |

#### 4.5. Transformer Baselines

Across individual seeds, Urdu-RoBERTa had the highest mean Macro-F1 ( $0.8747 \pm 0.0096$ ) and the lowest mean fitting time among the three transformer families. XLM-R had the highest mean ROC-AUC ( $0.9480 \pm 0.0019$ ) and Hate-class precision ( $0.9047 \pm 0.0053$ ), but lower recall. The clean mBERT rerun obtained a mean Macro-F1 of  $0.8640 \pm 0.0076$  and a mean ROC-AUC of  $0.9400 \pm 0.0020$ . Probability aggregation improved family-level performance, with Urdu-RoBERTa reaching Macro-F1 = 0.8836; XLM-R remained strongest by ROC-AUC (0.9529), and clean mBERT reached aggregate Macro-F1 = 0.8722.

*Table 7. Three-Seed Transformer Official Test Results*

| Model        | Mean Accuracy       | Mean Macro-F1       | Mean ROC-AUC        | Aggregate Precision | Aggregate Recall | Aggregate Macro-F1 | Aggregate ROC-AUC |
|--------------|---------------------|---------------------|---------------------|---------------------|------------------|--------------------|-------------------|
| mBERT        | $0.8641 \pm 0.0075$ | $0.8640 \pm 0.0076$ | $0.9400 \pm 0.0020$ | 0.8845              | 0.8564           | 0.8722             | 0.9447            |
| XLM-R        | $0.8745 \pm 0.0086$ | $0.8744 \pm 0.0087$ | $0.9480 \pm 0.0019$ | 0.9082              | 0.8364           | 0.8757             | 0.9529            |
| Urdu-RoBERTa | $0.8747 \pm 0.0096$ | $0.8747 \pm 0.0096$ | $0.9446 \pm 0.0047$ | 0.8809              | 0.8873           | 0.8836             | 0.9469            |

*Note.* The targeted mBERT compatibility rerun was completed under Transformers 4.57.1 using in-memory retention of the best validation state and strict=True restoration. For seeds 42, 52, and 62, the audit reported zero missing keys, zero unexpected keys, and no LayerNorm beta/gamma naming mismatch. The clean aggregate results therefore replace the earlier checkpoint-reload outputs.

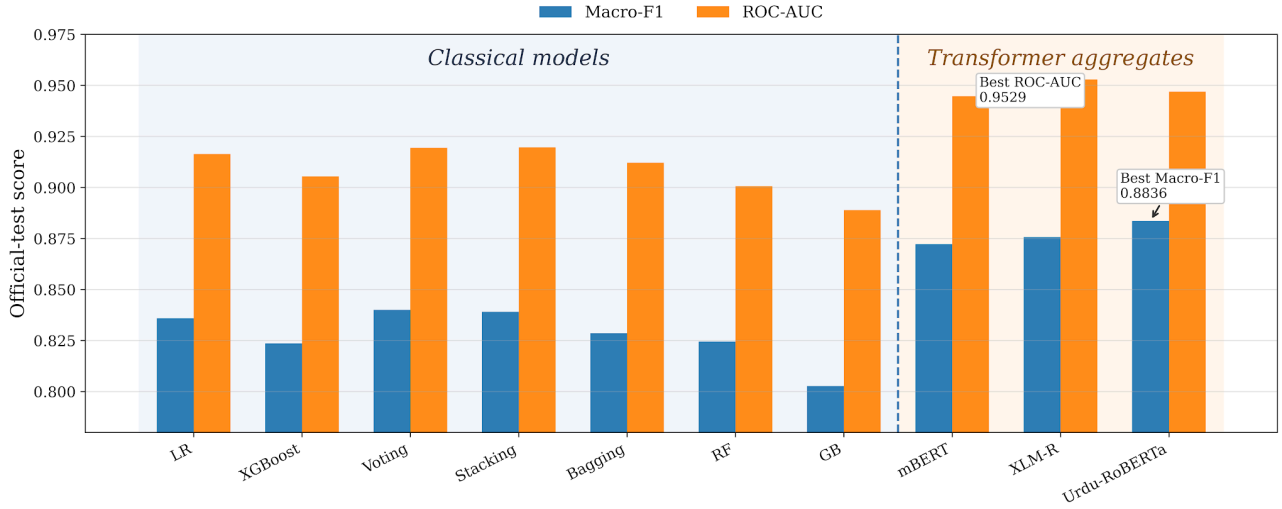


Figure 4. Official-test Macro-F1 and ROC-AUC for classical classifiers trained on the BPSO-selected representation and three-seed transformer probability aggregates

#### 4.6. Paired Statistical Tests and Error Analysis

The exact McNemar comparison between PSO Stacking and complete-hybrid Stacking was not significant after Holm correction (adjusted  $p=0.7274$ ), despite the higher official-test Macro-F1 achieved by complete-hybrid Stacking (0.8468 versus 0.8391).

PSO Stacking was also not significantly different from TF-IDF-only Stacking, embedding-only Stacking, Logistic Regression, Soft Voting, Bagging, or Random Forest after adjustment. It significantly outperformed XGBoost (adjusted  $p = 0.0359$ ) and Gradient Boosting (adjusted  $p = 4.43 \times 10^{-7}$ ). All three transformer aggregates significantly outperformed PSO Stacking: clean mBERT (adjusted  $p = 2.59 \times 10^{-4}$ ), XLM-R (adjusted  $p = 3.26 \times 10^{-5}$ ), and Urdu-RoBERTa (adjusted  $p = 1.07 \times 10^{-7}$ ).

Table 8. Selected exact McNemar comparisons with PSO Stacking as reference

| Comparator             | Discordant pairs (reference better / comparator better) | Holm-adjusted $p$     | Interpretation                       |
|------------------------|---------------------------------------------------------|-----------------------|--------------------------------------|
| Soft Voting            | 1 / 3                                                   | 1.000                 | No significant difference            |
| Hybrid without PSO     | 52 / 69                                                 | 0.727                 | No significant difference            |
| TF-IDF-only Stacking   | 140 / 152                                               | 1.000                 | No significant difference            |
| XGBoost                | 85 / 51                                                 | 0.0359                | PSO Stacking is significantly better |
| Gradient Boosting      | 146 / 66                                                | $4.43 \times 10^{-7}$ | PSO Stacking is significantly better |
| mBERT aggregate        | 113 / 186                                               | $2.59 \times 10^{-4}$ | Comparator significantly better      |
| XLM-R aggregate        | 109 / 190                                               | $3.26 \times 10^{-5}$ | Comparator significantly better      |
| Urdu-RoBERTa aggregate | 96 / 194                                                | $1.07 \times 10^{-7}$ | Comparator significantly better      |

PSO Stacking produced 175 false negatives and 179 false positives. A non-systematic qualitative inspection of the high-confidence error candidates suggested several recurring difficulties:

- (i) hostile political or institutional criticism that is not always explicitly group-directed;
- (ii) threats or violent imagery expressed without a conventional slur;
- (iii) reporting or quotation of harmful content; (iv) sarcasm and indirect insinuation;
- and (v) cases whose original multimodal label may depend partly on an excluded image or OCR stream.

These observations are illustrative rather than a formally annotated error taxonomy.

## 5. Discussion

The controlled ablation changes the interpretation of the original framework. The complete hybrid representation was the strongest classical configuration, but its gain over TF-IDF alone was small. This suggests that sparse lexical information remains highly informative for the present Urdu dataset, while sentence embeddings add a complementary but not dominant signal. Embeddings alone were weaker than TF-IDF and relatively expensive to fit within the Stacking pipeline.

PSO should be interpreted primarily as a representation-reduction mechanism. It selected approximately half of the hybrid dimensions and approximately halved the downstream Stacking fitting time in both the official-test experiment and nested validation. These timing comparisons begin after the representation or fold-local mask is available and therefore exclude embedding extraction and the PSO search itself. The reduced representation did not improve point-estimate performance: Macro-F1 decreased by 0.77 percentage points on the official test and 0.48 percentage points in nested validation. The paired difference from complete-hybrid Stacking was not statistically significant after correction, but this absence of significance should not be interpreted as proof of equivalence. The defensible result is a substantial reduction in dimensionality with a small predictive cost and a lower post-selection fitting time.

Soft Voting and Stacking were practically indistinguishable on the PSO mask. Soft Voting was much faster and obtained a marginally higher Macro-F1, whereas Stacking obtained a marginally higher ROC-AUC. The four discordant predictions and adjusted p-value of 1.000 provide no evidence that either ensemble is generally superior on this test set. Model choice between them should therefore depend on operational constraints rather than a trivial score difference.

Transformer baselines achieved substantially stronger scores, especially after probability aggregation. Urdu-RoBERTa's higher Macro-F1 and balanced precision-recall profile indicate the value of Urdu-focused tokenization and pretraining. XLM-R's high precision and ROC-AUC suggest strong ranking ability but a more conservative Hate-decision profile. These results provide a stronger reference benchmark for the classical framework. The classical PSO pipeline remains relevant when fixed representations, reduced dimensionality, or conventional model deployment is desired, but it should not be presented as state-of-the-art in predictive accuracy.

## 6. Limitations

First, the study uses one balanced dataset and one fixed official split; generalisation to other platforms, time periods, dialects, and naturally imbalanced prevalence is unknown. Second, MMHS11K was designed as a multimodal dataset, whereas this study excludes images and OCR. Some text-only errors may consequently reflect missing visual context or a mismatch between multimodal annotation and unimodal input. Third, binary Hate/No-Hate labels do not identify target groups, severity, incitement, quoted speech, or offensive-but-non-hateful content. Fourth, source annotations remain culturally and contextually subjective even though the original study reported substantial expert agreement. Fifth, BPSO is stochastic and computationally expensive; a single full-budget mask cannot characterise every possible search trajectory, although fold-local PSO and fixed seeds reduce this concern. Sixth, the experiments do not evaluate Roman Urdu, Urdu-English code mixing, adversarial spelling, temporal drift, interpretability, or out-of-domain robustness. Finally, transformer findings remain specific to the evaluated checkpoints, seeds, and software environments; the clean mBERT compatibility audit resolved the earlier checkpoint-reload ambiguity but does not replace external-dataset replication.

## 7. Conclusion and Future Work

This study presents a reproducible evaluation of BPSO-selected hybrid lexical-semantic features for Urdu hate-speech detection. The complete TF-IDF-plus-embedding representation achieved the strongest classical Stacking result, whereas BPSO reduced 5,768 dimensions to 2,849 and approximately halved post-selection fitting time with a small decrease in Macro-F1. The timing

comparison excludes embedding extraction and the BPSO search. Nested validation and Holm-corrected McNemar tests show that the paired difference from the complete classical hybrid was not statistically significant, but they do not establish equivalence and do not support claims that PSO improves accuracy or outperforms transformers. Urdu-RoBERTa achieved the highest aggregate Macro-F1, and XLM-R achieved the highest aggregate ROC-AUC.

Future work should test the feature-selection trade-off on independent, temporally separated, and naturally imbalanced Urdu datasets; restore image and OCR streams for multimodal modelling; distinguish target categories and quoted or counter-speech content; extend evaluation to Roman Urdu and code-mixed text; add error-aware calibration and interpretable evidence extraction; and replicate the transformer comparisons across additional Urdu-specific checkpoints and software environments.

## **Declarations**

### **Author contributions**

Conceptualisation, methodology, data curation, software, validation, formal analysis, investigation, visualisation, and Writing - original draft, review and editing: Haseeb Ullah, Muhammad Javed, Asia Kanwal, Hamid Masood Khan, Kiran Hanif, and Husnain Saleem. All authors contributed equally to these activities, read the manuscript, and approved the submitted version.

### **Data availability**

The MMHS11K dataset used in this study is publicly available through Figshare at DOI: <https://doi.org/10.6084/m9.figshare.27310764>. The processed data, prediction-level outputs, statistical results, and supporting materials required to verify the reported findings are supplied with this revision as a controlled data-and-code repository for editorial and reviewer verification. Public redistribution remains subject to the source dataset licence, applicable platform terms, and the repository's controlled-sharing notice.

### **Code availability**

The source code, preprocessing scripts, feature-extraction procedures, BPSO implementation, model-training workflows, evaluation scripts, and statistical-analysis code are supplied with this revision in the controlled data-and-code repository. The repository includes verification scripts, environment records, hashes, and audit manifests.

### **Ethics statement**

This study conducted secondary computational analysis of a publicly released dataset. No new participants were recruited, and no new human annotation was performed. The source publication reports the original data collection, consent, and annotation procedures.

### **Funding**

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### **Competing interests**

The authors declare that they have no competing interests.

## References

- Ali, M. Z., Ehsan-Ul-Haq, Rauf, S., Javed, K., & Hussain, S. (2021). Improving hate speech detection of Urdu tweets using sentiment analysis. *IEEE Access*, 9, 84296-84305. <https://doi.org/10.1109/ACCESS.2021.3087827>
- Arshad, M. U., Ali, R., Beg, M. O., & Shahzad, W. (2023). UHated: Hate speech detection in Urdu language using transfer learning. *Language Resources and Evaluation*, 57(2), 713-732. <https://doi.org/10.1007/s10579-023-09642-7>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM. <https://doi.org/10.1145/2939672.2939785>
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440-8451). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.747>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171-4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>
- Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall/CRC. <https://doi.org/10.1201/9780429246593>
- Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys*, 51(4), Article 85, 1-30. <https://doi.org/10.1145/3232676>
- Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65-70.
- Hussain, S., Malik, M. S. I., & Masood, N. (2022). Identification of offensive language in Urdu using semantic and embedding models. *PeerJ Computer Science*, 8, e1169. <https://doi.org/10.7717/peerj-cs.1169>
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942-1948). IEEE. <https://doi.org/10.1109/ICNN.1995.488968>
- Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics* (Vol. 5, pp. 4104-4108). IEEE. <https://doi.org/10.1109/ICSMC.1997.637339>
- Malik, M. S. I., Nawaz, A., & Jamjoom, M. M. (2024). Hate speech and target community detection in Nastaliq Urdu using transfer learning techniques. *IEEE Access*, 12, 116875-116890. <https://doi.org/10.1109/ACCESS.2024.3444188>
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153-157. <https://doi.org/10.1007/BF02295996>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85), 2825-2830.
- Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., & Patti, V. (2021). Resources and benchmark corpora for hate speech detection: A systematic review. *Language Resources and Evaluation*, 55, 477-523. <https://doi.org/10.1007/s10579-020-09502-8>
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language*

- Processing* (pp. 3982-3992). Association for Computational Linguistics. <https://doi.org/10.18653/v1/D19-1410>
- Rizwan, H., Shakeel, M. H., & Karim, A. (2020). Hate-speech and offensive language detection in Roman Urdu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (pp. 2512-2522). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.emnlp-main.197>
- Saddozai, F. K., Badri, S. K., Alghazzawi, D., Khattak, A., & Asghar, M. Z. (2025). Multimodal hate speech detection: A novel deep learning framework for multilingual text and images. *PeerJ Computer Science*, 11, e2801. <https://doi.org/10.7717/peerj-cs.2801>
- Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media* (pp. 1-10). Association for Computational Linguistics. <https://doi.org/10.18653/v1/W17-1101>
- Vidgen, B., & Derczynski, L. (2020). Directions in abusive language training data: A systematic review - Garbage in, garbage out. *PLOS ONE*, 15(12), e0243300. <https://doi.org/10.1371/journal.pone.0243300>
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., ... Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38-45). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
- Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2), 241-259. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)