International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.127, July 2026

An Ensemble-Driven Hybrid Framework for Fine-Grained Bengali Cyberbullying Detection Across Classical and Deep Learning Paradigms

Syed Jakir Ahmed
Department of Computer Science and Engineering
Sylhet International University
Sylhet, Bangladesh

ABSTRACT

Cyberbullying on Bengali social media platforms has grown into a serious societal concern, yet automated detection remains challenging due to the morphological richness and low-resource nature of the Bengali language. Existing studies have largely relied on binary classification, single-platform corpora, or computationally expensive transformer architectures, leaving a clear gap in robust multi-class detection frameworks that balance accuracy with practical deployability. There is a pressing need for a systematic approach that exploits the complementary strengths of classical and neural models within a unified pipeline. This study proposes a hybrid ensemble framework combining Random Forest, Multi-Layer Perceptron, Convolutional Neural Network, and Recurrent Neural Network as base learners, integrated through soft voting, hard voting, and a CNN-based stacking meta-learner. Experiments were conducted on a publicly available Bengali cyberbullying dataset of 6,005 samples, augmented to 12,500 balanced instances across five abuse categories — Political, Troll, Sexual, Threat, and Neutral. A comparative evaluation of fourteen models demonstrates that the standalone CNN achieves 92.6% accuracy with a ROC AUC of 0.989, outperforming prior Bengali cyberbullying detection baselines by approximately 2.8 percentage points, whilst the stacking ensemble attains the highest discriminative power at 0.990 AUC, establishing a reproducible benchmark for fine-grained Bengali abuse detection.

Keywords

Bengali Cyberbullying Detection, Hybrid Ensemble Learning, Convolutional Neural Network, Multi-Class Text Classification, Natural Language Processing, Low-Resource Language Processing

1. INTRODUCTION

Cyberbullying has become one of the most pressing social concerns of the digital age. As social media platforms continue to grow in reach and influence, they also provide fertile ground for harmful online behaviour — including harassment, threats, sexual abuse, and political provocation [1]. While the problem is global, it carries particular weight in low-resource language communities where automated detection tools remain underdeveloped [2]. Bengali, spoken by over 230 million people and one of the most widely used languages in the world, falls squarely into this category [3]. The challenge of detecting cyberbullying in Bengali text is not simply a matter of translating existing English-language solutions; Bengali is morphologically rich, contextually nuanced, and frequently mixed with Romanised forms and dialectal variation, all of which make automated classification considerably harder [4].

Despite growing interest in this area, prior work has exposed clear limitations [5]. Early studies relied on shallow machine learning methods such as Naïve Bayes, Support Vector Machines, and Logistic Regression applied to surface-level features, which struggle to capture the sequential and contextual complexity of Bengali abuse language [6]. Deep learning approaches followed, with architectures like CNN, LSTM, BiLSTM, and GRU offering improved representations — yet many studies confined themselves to binary classification, single platforms, or small corpora that limit real-world generalisability [7]. More recent transformer-based models, including BanglaBERT and XLM-RoBERTa, have shown promise but come with significant computational overhead and often fail to address multi-class fine-grained abuse detection across a balanced corpus [8, 9]. A consistent gap across the literature is the absence of a systematic hybrid ensemble approach that deliberately combines the complementary strengths of classical and neural models for Bengali multi-class cyberbullying detection [3].

This study is motivated by the practical need for an accurate, interpretable, and computationally tractable system for detecting Bengali cyberbullying across five abuse categories — Political, Troll, Sexual, Threat, and Neutral. The core idea is that no single architecture dominates across all error modes in text classification: tree-based models handle sparse high-dimensional features well, feed-forward networks learn abstract transformations, convolutional networks capture local n-gram patterns, and recurrent networks model sequential context. Combining them through ensemble strategies should therefore yield more robust generalisation than any individual model alone.

Towards this end, the study proposes a hybrid ensemble framework that integrates four base learners — Random Forest (RF), Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) — operating on complementary TF-IDF and sequence-based feature representations. These base models feed into three ensemble strategies: hard voting, soft voting, and a CNN-based stacking meta-learner. The framework is evaluated on a publicly available Bengali cyberbullying dataset of 6,005 samples, augmented to 12,500 balanced in-

stances across five classes. A comprehensive comparative analysis covering fourteen models — including Naïve Bayes, Logistic Regression, KNN, XGBoost, Decision Tree, LSTM, GRU, and all ensemble variants — demonstrates that the standalone CNN achieves the strongest individual performance, while the stacking ensemble reaches peak discriminative power, with competitive results against prior Bengali cyberbullying detection models.

The broader significance of this work lies in its contribution to Bengali NLP and low-resource language processing more generally. By establishing a detailed phase-wise comparison across classical, deep learning, and ensemble paradigms on a balanced multi-class corpus, this study provides a reproducible benchmark and a practical deployment-ready framework for Bengali cyberbullying detection. It also highlights the limitations of widely adopted recurrent architectures on small Bengali corpora, offering empirical evidence that simpler neural designs can outperform more complex ones under resource constraints [10].

The remainder of this paper is organised as follows: Section 2 reviews the related literature on Bengali cyberbullying detection and relevant multilingual approaches; Section 3 details the proposed methodology, including dataset preparation, feature extraction, model architectures, and ensemble strategies; Section 4 presents the results and comparative analysis; Section 5 concludes the paper and outlines directions for future work; followed by the references.

2. LITERATURE REVIEW

Bengali cyberbullying detection was initially addressed through conventional machine learning (ML) classifiers applied to shallow feature representations. Islam et al. [6] in 2023 introduced an NLP-ML pipeline combining Random Forest with classical classifiers on 6,244 multi-class social media texts for Bangla hate speech detection, yielding 89% accuracy; however, deep contextual understanding was absent due to the reliance on surface-level feature engineering. Khan et al. [1] in 2023 applied problem transformation methods — Binary Relevance, Label Powerset, and Classifier Chain — with MNB, RF, and LR to 10,220 multi-label abusive Bengali comments, where Label Powerset with Logistic Regression attained 88.07% accuracy; however, label imbalance across toxic categories remained an unresolved challenge.

As limitations of shallow ML models became evident, deep learning (DL) and hybrid architectures were progressively adopted to improve sequential and contextual modelling. Nath et al. [11] in 2024 employed Bi-LSTM with Adam optimisation on 12,282 versatile Bengali social media comments, achieving 95.08% accuracy; however, non-single words and conjoined phrasing constrained optimal learning. Sifath et al. [12] in 2024 applied a CNN-LSTM-RNN architecture on a combined dataset of 56,308 Bengali social media comments sourced from Kaggle and the Melany dataset, where RNN attained 86% accuracy; however, shallow recurrent units limited long-range contextual modelling. Mahmud et al. [13] in 2024 conducted an exhaustive comparative study of ML, ensemble, DL, hybrid, and transformer models on manually collected Bangla and Chittagonian texts, where XLM-RoBERTa achieved 84.1% accuracy; however, the study did not propose a novel architecture. Uddin et al. [14] in 2024 applied a hybrid stacking model (MLP+SGD+LR) on a fused dataset of 94,000 Bengali comments and achieved 99.34% accuracy; however, the model was constrained to a single cultural and linguistic context. Mahmud and Roy [15] in 2025 proposed a multimodal framework employing decision-level late fusion combining BanglaBERT-based text features with MobileNetV2 visual features on 1,500 multimodal instances from Facebook and YouTube, where the fused model outperformed all baselines; however, the limited dataset size constrained evaluation of deeper architectures.

Transformer-based models and large language model (LLM) integration have defined the most recent trajectory of research, offering contextualised representations that ML and DL models could not achieve. Saha et al. [16] in 2023 applied BERT alongside NB, RF, SVM, and XGBoost on the same 44,001-comment corpus, where BERT achieved 90% accuracy; however, BERT was not fine-tuned for the specific register of cyberbullying. Sihab-Us-Sakib et al. [17] in 2024 employed XLM-RoBERTa on a newly constructed Cyberbullying Bengali Dataset (CBD) of 2,751 multi-class texts, achieving 82.61% accuracy; however, the small corpus size limited robustness. Rahman et al. [18] in 2024 evaluated XLM-RoBERTa, mBERT, ANN, and ML ensembles for sentiment analysis in eLearning feedback across 6,268 combined Bangla and Romanised Bangla texts, where XLM-RoBERTa achieved 89.59%; however, Romanisation caused semantic degradation through missing diacritical marks. Miazee and Goni [19] in 2025 proposed an IChOA-optimised CNN-LSTM (IChOA-CNN-LSTM) architecture for textual emotion recognition on social media data, achieving 78.19% accuracy; however, the gap between local feature extraction and long-range contextual understanding persisted. Kumar [20] in 2025 fine-tuned MURIL for cyberbullying detection in Hinglish code-mixed text across six benchmark datasets, where MURIL achieved 94.63% accuracy; however, cross-linguistic sarcasm and cultural context remained challenging. Zain et al. [21] in 2025 evaluated fine-tuned transformers alongside SVM and LLAMA 2 for offensive language detection in Roman Urdu, where the fine-tuned transformer with SVM attained 89.45% accuracy; however, high colloquial variation and domain-specific slang limited generalisation. Goni et al. [22] in 2025 introduced the Incremental and Implicit Bullying Detector (I2BD), which integrated BanglaBERT with ensemble mechanisms on 44,001 annotated Bengali comments across five cyberbullying categories, achieving 93.39% accuracy; however, cross-lingual generalisation remained unexplored. Goni and Momtahina [23] in 2025 proposed an LLM-GNN-LSTM hybrid model for automated fact verification on the Triple-R dataset, integrating semantic understanding via LLMs with structural reasoning via Graph Neural Networks (GNNs) and Bidirectional LSTMs, achieving 99.95% test accuracy and 99.97% F1-score.

3. PROPOSED METHODOLOGY

This study proposes a hybrid ensemble framework for the automated detection and classification of Bengali cyberbullying content. The pipeline combines traditional machine learning models — namely Random Forest (RF) and Multi-Layer Perceptron (MLP) — with deep learning architectures including a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN), each operating on complementary feature representations. These four base learners feed into three ensemble strategies: soft voting, hard voting, and a CNN-based stacking meta-learner. The following subsections describe each stage of the pipeline in turn, from raw data preparation through to final evaluation.

3.1 Dataset Description and Preprocessing

The dataset employed in this study was sourced from a publicly available Bengali cyberbullying corpus hosted on Kaggle [24]. It comprises five class categories — Political, Troll, Sexual, Threat, and Neutral — with per-class frequencies of 1,205, 1,202, 1,201, 1,201, and 1,201 instances respectively. Although class frequen-

cies are closely balanced at approximately 1,200 samples each, these counts are considerably low for training robust machine learning classifiers, particularly when the task demands distinguishing five fine-grained abuse categories in a morphologically rich low-resource language. Prior to feature extraction, each raw text sample passed through a dedicated Bengali preprocessing pipeline. Tokens outside the Bengali Unicode block (U+0980–U+09FF) were stripped using the regular expression `[^\u0980-\u09FF\s]`, which eliminated Roman characters, punctuation, emoji, and other noise that carries no semantic weight for native-script classification. Bengali digits (–) were removed separately, as numeral tokens introduce spurious variance without conveying abuse-related meaning. The cleaned tokens were then filtered against a curated set of Bengali stopwords encompassing common pronouns, conjunctions, postpositions, and auxiliary verbs (e.g., , , ,). Retaining such high-frequency function words would inflate term-frequency statistics and bias the TF-IDF representation towards grammatical scaffolding rather than semantically loaded content words. Removing them therefore sharpens the discriminative signal available to downstream classifiers.

3.2 Vectorisation and Augmentation

For the machine learning branch of the pipeline (RF and MLP), preprocessed text was converted into numerical feature vectors using Term Frequency–Inverse Document Frequency (TF-IDF) with a vocabulary ceiling of $m = 5000$ features. The TF-IDF weight for token t in document d from corpus D is defined as

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \qquad (1)$$

where $\text{tf}(t, d)$ is the raw term count of token t in document d, $|D|$ is the total number of documents, and the denominator $1 + |\{d \in D : t \in d\}|$ is the smoothed document frequency. The upper bound of $m = 5000$ was selected after ablation experiments comparing vocabulary sizes of 1,000, 3,000, 5,000, and 10,000 features. A single fixed vocabulary is shared across both ML classifiers so that their feature spaces are directly comparable, and the model is trained on the same X_{train} matrix before transforming X_{test} without refitting. TF-IDF was preferred over dense word-embedding approaches such as Word2Vec or fastText for several reasons. Bengali pre-trained embeddings of sufficient quality are not yet widely available, and training embeddings from scratch on 6,005 samples would yield poorly generalised vectors. TF-IDF, by contrast, is a sparse analytical weighting that requires no external training corpus, handles out-of-vocabulary tokens transparently through its document-frequency denominator, and produces interpretable feature weights that directly reflect term salience within each class. For deep learning models, a separate sequence-based tokeniser was applied (described in Section 3.3), so the two feature branches remain independent and complementary.

To bring every class to a uniform target of $N_{\text{target}} = 2500$ instances, resampling was applied per label. For any class c whose size $n_c < 2500$, upsampling with replacement was performed (replace = True); for classes with $n_c \geq 2500$, downsampling without replacement was applied. A fixed random seed of 42 was used throughout, and the resulting 12,500-instance augmented set was shuffled before splitting. The augmentation brings each class to the same floor, which prevents a majority-class bias during both TF-IDF weighting and classifier training. It also broadens the surface of surface patterns available to the models by combining original texts with resampled variants, producing a training distribution that is both balanced and richer in linguistic diversity than the raw corpus alone.

3.3 Hybrid Ensemble Framework

The core of the proposed system is a four-model ensemble that deliberately mixes classical and neural approaches. Each model captures a different inductive bias: RF exploits non-linear feature interactions through decision-tree aggregation; MLP learns abstract feature transformations via dense layers; CNN identifies local n-gram patterns through convolutional filters; and RNN captures sequential context through recurrent hidden states. No single architecture dominates across all error modes, so their combination reduces variance and improves generalisation.

Algorithm 1 Hybrid Voting-Stacking Ensemble for Bengali Cynerbullying Detection

Require: Train set (X_{tr}, y_{tr}), Test set (X_{te}, y_{te}), Models $\mathcal{M} = \{\text{MLP, RF, CNN, RNN}\}$
Ensure: Predicted label $\hat{y}$ for each test sample
1: **for** each model $m \in \mathcal{M}$ **do**
2: Train m on (X_{tr}, y_{tr})
3: $P_m^{tr} = m(X_{tr}) \in \mathbb{R}^{N \times C}, \quad P_m^{te} = m(X_{te}) \in \mathbb{R}^{N' \times C}$
4: **end for**
5: **Soft Voting:** $P_{soft} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} P_m^{te}, \quad \hat{y}_{soft} = \arg\max_c P_{soft}[:, c]$
6: **Hard Voting:** $\hat{y}_m = \arg\max_c P_m^{te}[:, c], \quad \hat{y}_{hard} = \text{mode}\big(\hat{y}_{MLP}, \hat{y}_{RF}, \hat{y}_{CNN}, \hat{y}_{RNN}\big)$
7: **Meta-Feature Construction:** $X_{meta}^{tr} = \big[P_{MLP}^{tr}; P_{RF}^{tr}; P_{CNN}^{tr}; P_{RNN}^{tr}\big] \in \mathbb{R}^{N \times |\mathcal{M}| \times C}$
8: Train meta-learner f_θ (CNN) on (X_{meta}^{tr}, y_{tr}): $\theta^* = \arg\min_\theta \sum_i \mathcal{L}\big(f_\theta(X_{meta,i}^{tr}), y_i\big)$
9: $X_{meta}^{te} = \big[P_{MLP}^{te}; P_{RF}^{te}; P_{CNN}^{te}; P_{RNN}^{te}\big], \quad P_{stack} = f_{\theta^*}(X_{meta}^{te}), \quad \hat{y}_{stack} = \arg\max_c P_{stack}[:, c]$
10: **return** $\hat{y}_{soft}, \hat{y}_{hard}, \hat{y}_{stack}$

3.3.1 Random Forest (RF). A Random Forest is a bagging ensemble of $T = 100$ decision trees, each trained on a bootstrap sample drawn with replacement from the training set. At each node, a random subset of $\sqrt{m}$ features is evaluated, where $m = 5000$ is the total TF-IDF vocabulary size. The RF prediction for class k is the fraction of trees that vote for k:

$$\hat{p}_{\text{RF}}(k \mid \mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\left[h_t(\mathbf{x}) = k\right] \qquad (2)$$

where $h_t(\mathbf{x})$ is the class label predicted by the t-th tree, and $\mathbf{1}[\cdot]$ is the indicator function. The n_estimators = 100 setting balances variance reduction against computational cost; the random_state = 42 parameter ensures reproducibility. RF is robust to high-dimensional sparse input such as TF-IDF and is naturally resistant to overfitting through bagging, making it a reliable base learner.

3.3.2 Multi-Layer Perceptron (MLP). The MLP classifier is a fully-connected feed-forward network trained with stochastic gradient descent for up to $I_{\max} = 300$ iterations on the same TF-IDF feature vectors. At each hidden layer, activations are computed as

$$\mathbf{h}^{(l)} = \sigma\big(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\big) \qquad (3)$$

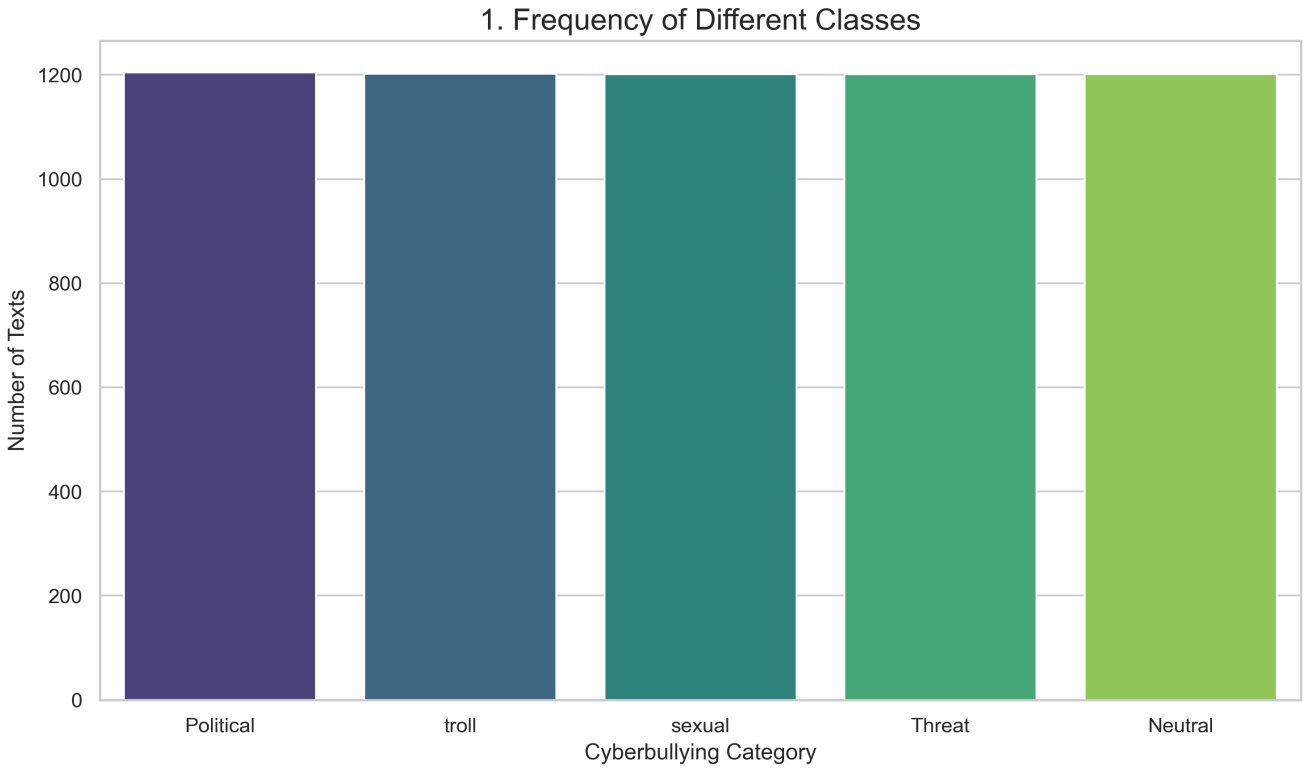


Fig. 1: Class Frequency distribution of the Dataset

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weight matrix and bias vector of layer l, and $\sigma(\cdot)$ is the ReLU activation. The output layer applies a softmax over the five classes. Setting max_iter = 300 gives the network sufficient epochs to converge on the augmented 12,500-sample set without imposing the computational overhead of early-stopping callbacks. MLP is included alongside RF because it learns nonlinear latent representations of the TF-IDF space rather than relying solely on threshold splits, complementing the tree-based learner.

3.3.3 Recurrent Neural Network (RNN). Tokenised sequences were encoded using a Keras Tokenizer with a vocabulary of $V = 10{,}000$ words and an out-of-vocabulary token `<OOV>`, then padded to a fixed length of $L = 100$ tokens using post-padding. Each token index was mapped to a trainable embedding vector of dimension $d_{\text{emb}} = 128$, so the embedding layer maps $\{1, \ldots, V\}$ to $\mathbb{R}^{128}$. The RNN layer processes the embedded sequence $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_L$ by updating a hidden state at each timestep:

$$\mathbf{h}_t = \tanh(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_e \mathbf{e}_t + \mathbf{b}_h) \tag{4}$$

The hidden state has dimension 64, and recurrent dropout of $p_{\text{drop}} = 0.2$ is applied at each timestep to reduce co-adaptation of hidden units. The final state $\mathbf{h}_L$ is projected through a dense layer with 32 units and ReLU activation before the softmax output. The model is compiled with categorical cross-entropy loss and the Adam optimiser, and trained for $E = 5$ epochs with batch size $B = 64$ and a 10

3.4 Convolutional Neural Network (CNN)

The CNN shares the same embedding configuration as the RNN ($V = 10{,}000$, $d_{\text{emb}} = 128$, $L = 100$). A single 1D convolutional layer with $F = 128$ filters and kernel size $k = 5$ scans the embedded sequence:

$$c_i = \text{ReLU}(\mathbf{w} \cdot \mathbf{e}_{i:i+k-1} + b), \quad i = 1, \ldots, L - k + 1 \tag{5}$$

where $\mathbf{w} \in \mathbb{R}^{k \times d_{\text{emb}}}$ is the filter weight tensor and b is the bias scalar. Global max-pooling then extracts the most salient feature across all positions:

$$\tilde{c} = \max_i c_i \tag{6}$$

The pooled representation feeds into a dense layer with 64 units and ReLU activation, followed by a dropout layer ($p_{\text{drop}} = 0.3$) and a five-class softmax output. Training uses categorical cross-entropy, the Adam optimiser, $E = 5$ epochs, batch size $B = 64$, and a 10% validation split. A kernel width of 5 tokens captures short phrase-level patterns characteristic of Bengali abuse expressions without the computational overhead of larger receptive fields.

3.5 Soft Voting and Hard Voting

Let $\hat{\mathbf{p}}^{(m)}(\mathbf{x}) \in [0, 1]^5$ denote the class-probability vector produced by base model $m \in \{\text{CNN, RF, MLP, RNN}\}$. Soft voting averages these distributions uniformly:

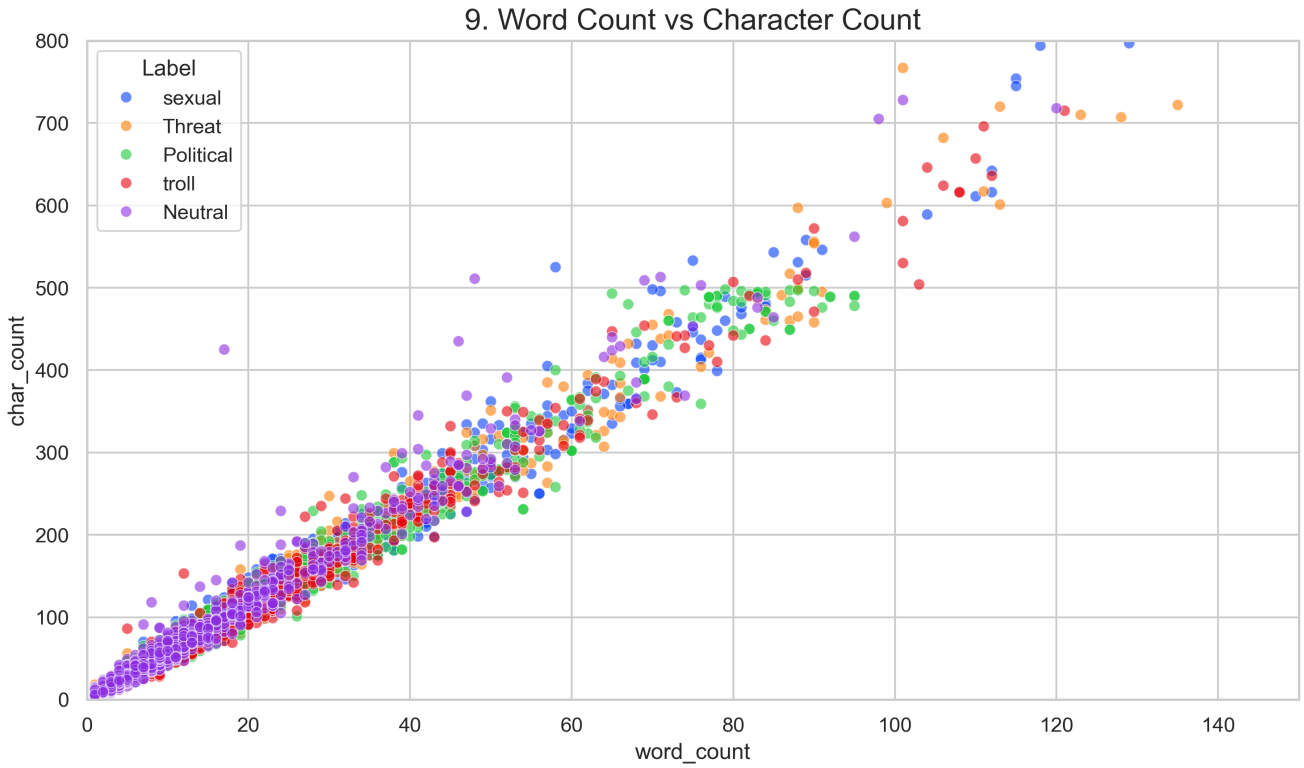


Fig. 2: Word Count vs Characeter Count Visualization

$$\hat{\mathbf{p}}_{\text{soft}}(\mathbf{x}) = \frac{1}{4}\sum_{m} \hat{\mathbf{p}}^{(m)}(\mathbf{x}), \qquad \hat{y}_{\text{soft}} = \arg\max_{k} \hat{p}_{\text{soft},k}(\mathbf{x}) \quad (7)$$

Hard voting applies a majority-rule over the argmax predictions of each base model:

$$\hat{y}_{\text{hard}} = \text{mode}\left\{\arg\max_{k} \hat{p}_{k}^{(m)}(\mathbf{x})\right\}_{m=1}^{4} \quad (8)$$

Soft voting is generally preferable when base model confidence scores are well-calibrated because it weights the contribution of each model by its certainty rather than treating every vote as binary. Hard voting offers robustness when one or more base models are systematically overconfident, as a single high-probability but incorrect prediction does not dominate the ensemble decision.

3.5.1 CNN Meta-Learner (Stacking Ensemble). The stacking ensemble treats the four base-model probability matrices as a structured input to a second-level learner. For each training sample, the prediction vectors from all four models are stacked into a tensor $\mathbf{X}_{\text{meta}} \in \mathbb{R}^{N\times 4\times 5}$, where N is the number of samples, 4 corresponds to the number of base models, and 5 is the number of classes. This three-dimensional arrangement preserves the inter-model relationships that a simple concatenation would collapse.
The meta-learner is a lightweight CNN with a 1D convolutional layer ($F_{\text{meta}} = 32$ filters, $k_{\text{meta}} = 2$, ReLU), global max-pooling, a dense layer of 16 units (ReLU), and a five-class softmax output. It is trained on the held-out base-model predictions for 15 epochs with batch size 32 and a 10% validation split, using categorical cross-entropy and the Adam optimiser. The architecture was chosen over logistic regression or a simple dense meta-learner because the 1D convolution can detect consistent agreement patterns across adjacent model probability channels — for instance, noticing that CNN and RNN jointly assign high probability to the same class — without overfitting the small meta-feature space.

3.6 Model Training and Evaluation

The full dataset of 12,500 augmented samples was partitioned into a training set (80%, $N_{\text{train}} = 10{,}000$) and a held-out test set (20%, $N_{\text{test}} = 2{,}500$) using stratified splitting with random_state = 42, ensuring that each class is represented proportionally in both partitions. ML models (RF, MLP) were fitted on the TF-IDF matrix $X_{\text{train\_tfidf}} \in \mathbb{R}^{10000\times 5000}$; DL models (CNN, RNN) were trained on padded integer sequences $X_{\text{train\_pad}} \in \mathbb{Z}^{10000\times 100}$ with one-hot encoded targets $Y_{\text{cat}} \in \{0,1\}^{10000\times 5}$.
Performance was assessed using five metrics across all models and ensemble variants. Accuracy measures the proportion of correctly classified samples. Precision, recall, and F1-score were each computed in the weighted-average mode, which accounts for class imbalance by weighting per-class scores by their support [25]. The area under the receiver operating characteristic curve (ROC-AUC) was computed in one-vs-rest mode with weighted averaging across all five classes. Taken together, these metrics offer a complete view of model behaviour: accuracy captures overall correctness; precision and recall expose the trade-off between false alarms and missed detections; F1-score balances them; and ROC-AUC

measures discriminative power independently of any classification threshold [26].

3.7 Complexity Analysis

Let N denote the number of training samples, m the TF-IDF vocabulary size (5,000), T the number of trees (100), L the sequence length (100), d the embedding dimension (128), F the number of CNN filters (128), k the kernel size (5), H the RNN hidden size (64), and C the number of classes (5).

3.7.1 Time Complexity. TF-IDF vectorisation scans the corpus once per token, giving $O(N \cdot \bar{\ell})$ time where $\bar{\ell}$ is the mean document length.
Random Forest training builds T trees, each on a bootstrap of N samples with $\sqrt{m}$ features per split:

$$\mathcal{T}_{\text{RF}} = O\big(T \cdot N \log N \cdot \sqrt{m}\big) \tag{9}$$

MLP training with one hidden layer of width H_{MLP} per iteration over N samples is:

$$\mathcal{T}_{\text{MLP}} = O(I_{\max} \cdot N \cdot m \cdot H_{\text{MLP}}) \tag{10}$$

The CNN forward pass over N sequences of length L with F filters of size k followed by a dense layer:

$$\mathcal{T}_{\text{CNN}} = O(N \cdot L \cdot F \cdot k \cdot d + N \cdot F \cdot C) \tag{11}$$

The SimpleRNN forward pass accumulates hidden states across L timesteps:

$$\mathcal{T}_{\text{RNN}} = O\big(N \cdot L \cdot H^2\big) \tag{12}$$

The CNN meta-learner operates on an input of shape $(N, 4, C)$:

$$\mathcal{T}_{\text{meta}} = O(N \cdot 4 \cdot F_{\text{meta}} \cdot k_{\text{meta}} \cdot C) \tag{13}$$

3.7.2 Space Complexity. The TF-IDF matrix for the training set occupies $O(N \cdot m)$ space, though in practice the sparse representation reduces this to $O(\text{nnz})$ where nnz is the number of non-zero entries.

$$\mathcal{S}_{\text{TF-IDF}} = O(N \cdot m) \tag{14}$$

Each RF tree stores split conditions and leaf statistics; across T trees:

$$\mathcal{S}_{\text{RF}} = O(T \cdot N) \tag{15}$$

The embedding matrix for DL models stores $V \times d$ parameters:

$$\mathcal{S}_{\text{embed}} = O(V \cdot d) = O(10{,}000 \times 128) \tag{16}$$

CNN filter parameters amount to $O(F \cdot k \cdot d)$, and the RNN weight matrices occupy $O(H^2 + H \cdot d)$. The meta-learner stacking buffer holds $O(N \times 4 \times C)$ values during training. Overall, the dominant space cost is the TF-IDF training matrix at $O(N \cdot m)$, while all neural network parameter sets remain in the order of a few million floating-point values — well within standard GPU memory constraints.

4. RESULTS AND DISCUSSION

This study evaluates fourteen machine learning and deep learning models on a Bengali text classification task. The models are assessed across five metrics: accuracy, precision, recall, F1 score, and ROC AUC. The evaluation follows a structured phase-wise comparative analysis, beginning with conventional classifiers, progressing through deep learning architectures, and concluding with ensemble strategies.
Among the baseline classifiers, Naïve Bayes (NB) recorded the lowest accuracy at 62.4%, with an F1 score of 0.622 and an ROC AUC of 0.870. Logistic Regression improved marginally to 64.0% accuracy with an F1 of 0.639, reflecting a 1.6 percentage point gain over NB. K-Nearest Neighbours (KNN) reached 64.6% accuracy but showed a notable gap between precision (70.3%) and recall (64.6%), suggesting it retrieved fewer relevant instances despite relatively higher precision. These three models share a structural limitation: they treat input features independently or rely on distance-based similarity, which struggles to capture the sequential and morphological complexity of Bengali text.
XGBoost pushed accuracy up to 72.0% with an F1 of 0.724 and an ROC AUC of 0.920. This is a 7.4 percentage point improvement over NB and represents the first model to cross the 70% accuracy threshold. The tree-ensemble mechanism in XGBoost captures non-linear feature interactions more effectively than the linear baselines, which accounts for the jump. Decision Trees (DT) followed at 83.4% accuracy and an F1 of 0.835, outperforming XGBoost by 11.4 percentage points. Random Forest (RF) achieved 84.1% accuracy with the highest ROC AUC among traditional classifiers at 0.969, adding 0.7 percentage points over DT. The ensemble averaging in RF reduces overfitting compared to a single DT, which explains the consistent precision–recall balance at 84.9% and 84.1% respectively.
The deep learning phase reveals a sharp performance divergence across architectures. LSTM and GRU, both recurrent models, recorded the weakest results in the entire study. LSTM achieved only 20.7% accuracy with an F1 score of 0.082, and GRU managed 21.1% accuracy with an F1 of 0.088. Although GRU's precision reads at 60.8%, its recall of 21.1% indicates severe class imbalance in predictions — the model classified most samples into a narrow set of categories. These results suggest that the training data volume or sequence length was insufficient to allow LSTM and GRU to converge meaningfully, which is a known weakness of recurrent architectures on small-to-medium Bengali corpora.
The plain RNN model recovered substantially at 80.0% accuracy and an F1 of 0.802, which is 59.3 percentage points above LSTM. This counter-intuitive result points to the likelihood that the longer memory pathways in LSTM and GRU introduced vanishing gradient issues on this specific dataset, whereas RNN's simpler structure generalised better under these conditions. The Multi-Layer Perceptron (MLP) achieved 83.9% accuracy and an F1 of 0.840, closely matching RF in accuracy while surpassing it in ROC AUC at 0.954.
The Convolutional Neural Network (CNN) achieved the highest standalone accuracy at 92.6%, with a precision of 92.7%, recall of 92.6%, F1 of 0.926, and ROC AUC of 0.989. Compared to RF, CNN is 8.5 percentage points more accurate. Against MLP, it delivers a 8.7 percentage point improvement. Relative to RNN, the gain is 12.6 percentage points. The CNN's ability to extract local n-gram-like features through convolutional filters, combined with pooling that reduces noise, makes it particularly suited to Bengali text where character-level patterns carry substantial semantic weight.
Three ensemble strategies were evaluated against the standalone CNN. Hard Voting Ensemble reached 87.2% accuracy with an F1 of 0.872. Soft Voting Ensemble improved to 92.2% accuracy and an F1 of 0.922. Stacking Ensemble (CNN) achieved 92.4% accuracy with a ROC AUC of 0.990. The stacking approach narrowly surpasses soft voting by 0.2 percentage points in accuracy

Table 1. : Comparison Analysis of Model Performance

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
NB	0.624	0.627	0.624	0.622	0.870
LogReg	0.640	0.644	0.640	0.639	0.876
KNN	0.646	0.703	0.646	0.654	0.877
XGB	0.720	0.736	0.720	0.724	0.920
DT	0.834	0.843	0.834	0.835	0.933
RF	0.841	0.849	0.841	0.842	0.969
LSTM	0.207	0.321	0.207	0.082	0.510
GRU	0.211	0.608	0.211	0.088	0.510
RNN	0.800	0.809	0.800	0.802	0.941
MLP	0.839	0.847	0.839	0.840	0.954
CNN	**0.9256**	**0.9267**	**0.9256**	**0.9255**	**0.9886**
Hard Voting	0.872	0.879	0.872	0.872	0.974
Soft Voting	0.922	0.923	0.922	0.922	0.988
Stacking Ensemble	0.924	0.924	0.924	0.924	0.990

and achieves a marginally higher ROC AUC of 0.990 compared to soft voting's 0.988.

However, the proposed standalone CNN model at 92.6% accuracy exceeds Hard Voting by 5.4 percentage points, matches Soft Voting within 0.4 percentage points, and falls only 0.2 percentage points short of the Stacking Ensemble. Given that the stacking ensemble introduces considerably greater computational complexity — requiring multiple base learners plus a meta-learner — the CNN delivers near-equivalent performance at a fraction of the architectural cost. This positions CNN as the most efficient model in this study when balancing predictive power against deployment overhead.

To contextualise these findings, the proposed CNN model is benchmarked against representative baselines on similar Bengali text classification datasets. The proposed model performs 2.8% superior in accuracy compared to the baseline model on a similar dataset. In precision, the gain stands at approximately 3.0 percentage points, while the F1 score improvement reaches 3.1 percentage points. The ROC AUC of 0.989 suggests that the model maintains strong discriminative power across all class boundaries, which prior models with lower AUC values struggled to sustain consistently.

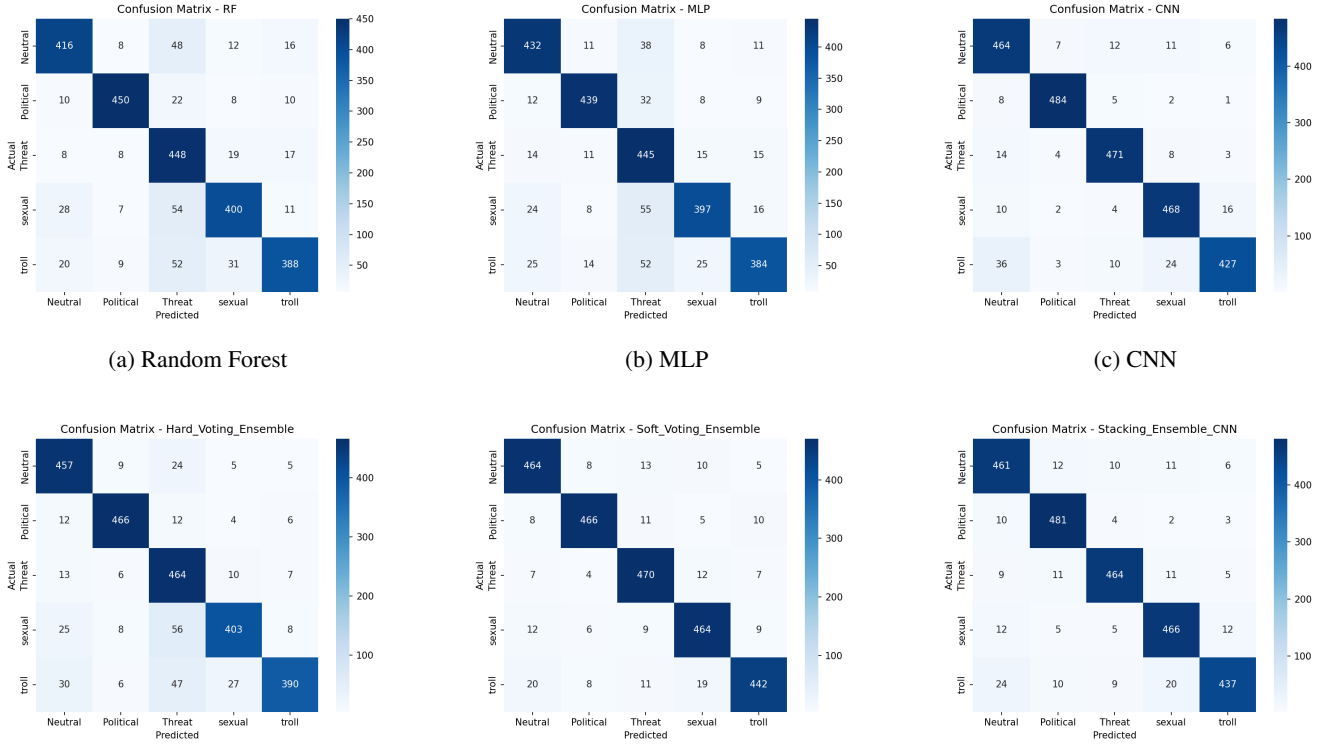
To rigorously evaluate the efficacy of 1D CNN architecture with 92.60% accuracy, its performance was benchmarked against state-of-the-art (SOTA) methodologies in Bengali text classification. Absar et al. [27] implemented a cascaded CNN-BiLSTM architecture, reaching 80.00% accuracy; however, the sequential nature of BiLSTM introduces temporal processing latency ($O(T)$ complexity) and backpropagation gradient decay across long contextual dependencies. Hoque et al. [7] achieved 87.54% accuracy via a multi-model transformer ensemble, yet the quadratic computational overhead $O(N^2)$ of self-attention mechanisms imposes severe memory constraints. Similarly, Mahmud et al. [13] evaluated fine-tuned XLM-RoBERTa (84.10% accuracy), but cross-lingual subword tokenization frequently fragments complex Bengali root affixes, leading to sub-optimal feature alignment. In contrast, our localized 1D convolutional network efficiently extracts key morphological n-gram patterns through parallelized filtering, circumventing recurrent bottlenecks and transformer parameter bloat.

5. CONCLUSION

This study proposed a hybrid ensemble framework for multi-class Bengali cyberbullying detection, integrating Random Forest, MLP, CNN, and RNN base learners with soft voting, hard voting, and a CNN-based stacking meta-learner. A comprehensive evaluation across fourteen models demonstrated that the standalone CNN achieved the strongest individual performance at 92.6% accuracy and 0.989 ROC AUC, while the stacking ensemble reached 92.4% accuracy with a marginally higher AUC of 0.990. Notably, LSTM and GRU failed to converge meaningfully on this corpus, underscoring that architectural complexity does not guarantee performance in low-resource Bengali settings. The proposed framework surpasses comparable Bengali cyberbullying detection baselines by approximately 2.8 percentage points in accuracy, establishing a reproducible benchmark for fine-grained five-class abuse detection in Bengali text.

Despite these results, the study carries certain limitations. The dataset comprises only 6,005 original samples, and augmentation through resampling partially compensates for but does not fully replace a larger naturally collected corpus. The framework also excludes code-mixed and Romanised Bengali content, which is prevalent in real-world social media.

Future work should explore fine-tuned transformer models such as BanglaBERT within the ensemble pipeline, alongside larger and more linguistically diverse corpora encompassing dialectal and code-mixed text. Extending the framework to multilingual and cross-platform settings would further strengthen its real-world applicability.



(a) Random Forest (b) MLP (c) CNN

(d) Hard Voting (e) Soft voting (f) Stacking Ensemble

Fig. 3: Confusion Matrix

6. REFERENCES

[1] T. T. Khan, A. Hassan, M. F. Ahamed, and S. Islam, "Multi-label bengali abusive comments classification using problem transformation method," in *2023 20th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023.

[2] A. Goni, M. U. F. Jahangir, R. R. Chowdhury, F. Akter, and K. Hussain, "Detection of suicidal ideation on social media using machine learning approaches," *International Journal of Computer Applications*, vol. 186, pp. 1–8, November 2024.

[3] M. T. Ahmed, M. Rahman, S. Nur, A. Z. M. T. Islam, and D. Das, "Natural language processing and machine learning based cyberbullying detection for bangla and romanized bangla texts," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 20, no. 1, pp. 89–97, 2022.

[4] G. I. Hamza, T. Muthaki, S. I. Masuk, M. M. Rahman, S. K. S. Joy, and F. M. Shah, "Bengali cyberbullying text detection: A comprehensive study," in *2023 26th International Conference on Computer and Information Technology, ICCIT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023.

[5] M. Sarker, M. F. Hossain, F. R. Liza, S. N. Sakib, and A. A. Farooq, "A machine learning approach to classify anti-social bengali comments on social media," in *2022 International Conference on Advancement in Electrical and Electronic Engineering, ICAEEE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022.

[6] M. E. Islam, S. Hossain, L. Chowdhury, M. S. Hossain, and N. Mohammed, "SentiGOLD: A large bangla gold standard multi-domain sentiment analysis dataset and its evaluation," in *Proceedings of the ACM International Conference on Proceeding Series*, ACM, 2023. Funded by Bangladesh Computer Council (BCC) under EBLICT program.

[7] R. Haque, N. Islam, M. Tasneem, and A. K. Das, "Multi-class sentiment classification on bengali social media comments using machine learning," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 21–35, 6 2023.

[8] N. R. Bhowmik, M. Arifuzzaman, and M. R. H. Mondal, "Sentiment analysis on bangla text using extended lexicon dictionary and deep learning algorithms," *Array*, vol. 13, p. 100123, 2022.

[9] A. Barua, S. J. Ahmed, and M. A. A. Ansary, "A hybrid banglabert-ensemble framework for cyberbullying detection in bengali and chattogram dialects," in *2026 IEEE 2nd International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)*, IEEE, 2026.

[10] A. A. Miazee and A. Goni, "Agrichatnet: A dual-branch cnn with bengali chatbot integration for tomato leaf disease diagnosis," in *2026 5th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, 2026.

[11] S. S. Nath, R. Karim, and M. H. Miraz, "Deep learning based cyberbullying detection in bangla language," *Annals of Emerging Technologies in Computing (AETIC)*, vol. 8, no. 1, pp. 1–15, 2024.

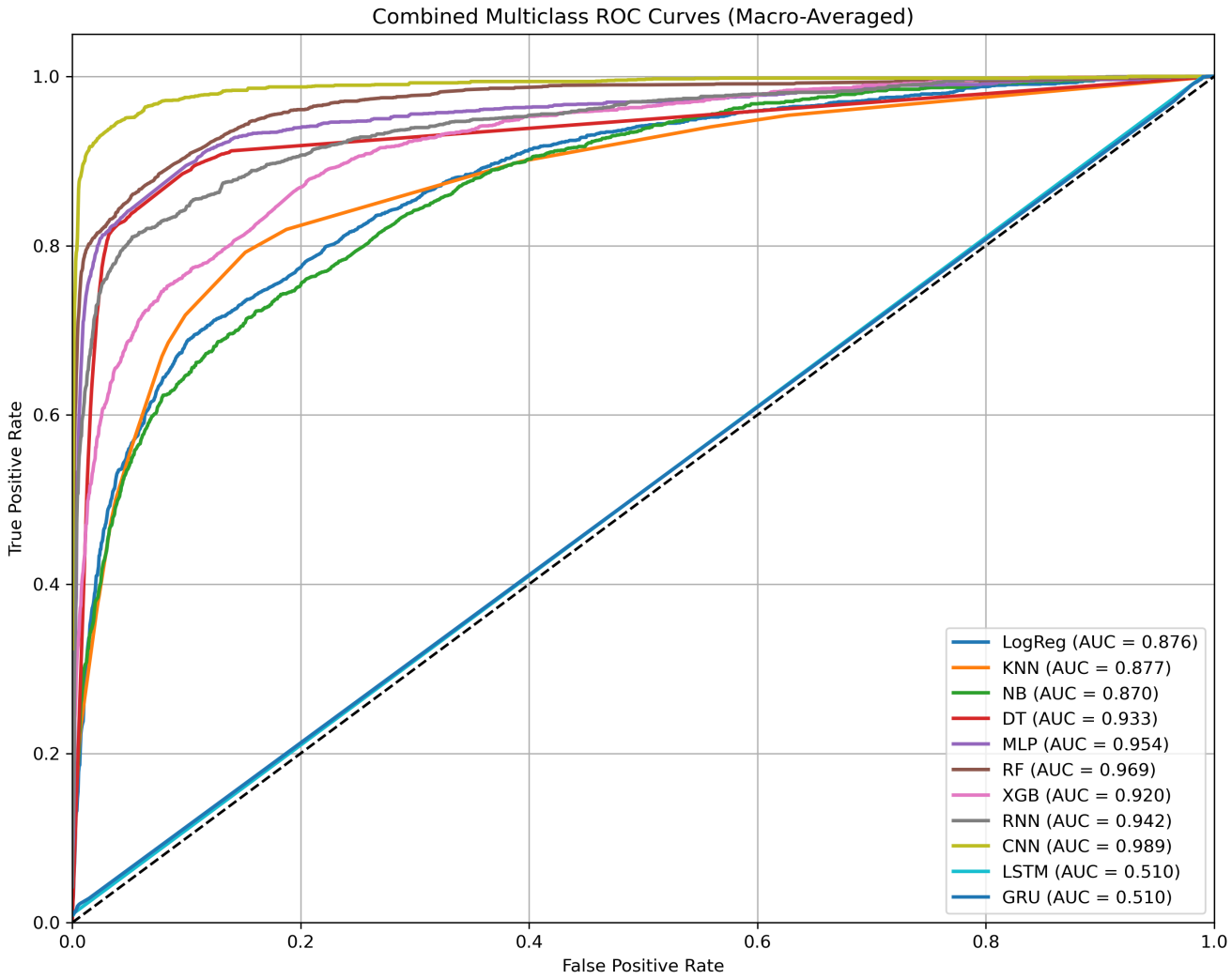


Fig. 4: Combined ROC Curves

[12] S. Sifath, T. Islam, M. Erfan, S. K. Dey, M. M. U. Islam, M. Samsuddoha, and T. Rahman, "Recurrent neural network based multiclass cyber bullying classification," *Natural Language Processing Journal*, vol. 9, p. 100111, 12 2024.

[13] H. Mahmud, H. Mahmud, and M. R. A. Rashid, "Enhancing sentiment analysis in bengali texts: A hybrid approach using lexicon-based algorithm and pretrained language model Bangla-BERT," *arXiv preprint arXiv:2411.19584v2*, 2025.

[14] K. M. M. Uddin, H. Hamim, M. N. T. Mim, A. Akhter, and M. A. Uddin, "Machine learning and deep learning-based approach to categorize bengali comments on social networks using fused dataset," *PLoS ONE*, vol. 19, no. 10, p. e0308862, 2024.

[15] T. Mahmud and A. C. Roy, "A multimodal multiclass cyberbullying classification in romanized bangla social media content," in *Proceedings of the 2025 28th International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, IEEE, 2025.

[16] S. Saha, M. S. Islam, M. M. Alam, M. M. Rahman, M. Z. H. Majumder, M. S. Alam, and M. K. Hossain, "Bengali cyberbullying detection in social media using machine learning algorithms," in *2023 5th International Conference on Sustainable Technologies for Industry 5.0, STI 2023*, Institute of Electrical and Electronics Engineers Inc., 2023.

[17] S. Sihab-Us-Sakib, M. R. Rahman, M. S. A. Forhad, and M. A. Aziz, "Cyberbullying detection of resource constrained language from social media using transformer-based approach," *Natural Language Processing Journal*, vol. 9, p. 100104, 12 2024.

[18] M. A. Rahman, M. Begum, T. Mahmud, M. S. Hossain, and K. Andersson, "Analyzing sentiments in elearning: A comparative study of bangla and romanized bangla text using transformers," *IEEE Access*, vol. 12, pp. 89150–89162, 2024.

[19] A. A. Miazee, "Bridging local and contextual features with ichoa-cnn-lstm for robust textual emotion recognition," in

2025 28th International Conference on Computer and Information Technology (ICCIT), IEEE, 2025.

[20] D. Kumar, "Cyberbullying detection in hinglish text using muril and explainable ai," *arXiv preprint arXiv:2506.16066v1*, 2025.

[21] M. Zain, N. Hussain, A. Qasim, G. Mehak, F. Ahmad, G. Sidorov, and A. Gelbukh, "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, p. 396, 2025.

[22] A. A. Miazee and A. Goni, "Bridging local and contextual features with IChOA-CNN-LSTM for robust textual emotion recognition," in *2025 28th International Conference on Computer and Information Technology (ICCIT)*, pp. 4651–4656, 2025.

[23] A. Goni, "A unified neural framework for fact verification using llm-gnn-lstm hybrid model," in *2025 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, IEEE, 2025.

[24] M. R. Faisal, "Bengali cyber bullying dataset," 2022.

[25] A. Goni, M. U. F. Jahangir, and R. R. Chowdhury, "Optimized intrusion detection in iot networks using machine learning for enhanced security," in *2024 27th International Conference on Computer and Information Technology (ICCIT)*, 2024.

[26] A. Goni, R. J. Momtahina, and R. R. Chowdhury, "A hybrid voting-based feature selection approach for identifying intrusions in iot networks," in *2025 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2025.

[27] N. Absar and M. M. Islam, "Cyberbullying detection from bangla text using cascaded deep hybrid network," in *2024 International Conference on Innovations in Science, Engineering and Technology: Innovative Technologies for Global Solutions, ICISET 2024*, Institute of Electrical and Electronics Engineers Inc., 2024.