Code_Gen at BLP-2025 Task 1: Enhancing Bangla Hate Speech Detection with Transformers through Token-Aware Adversarial Contrastive Training and Layer-wise Learning Rate Decay

Shifat Islam[1]* **Abhishek Agarwala**[1]* **Emon Ghosh**[2]*
Department of Computer Science
[1]Bangladesh University of Engineering and Technology
[2]Ahsanullah University of Science and Technology
{shifat.islam.buet, abhishek.agarwal0395, emonghosh005}@gmail.com

Abstract

Bangla social media contains several types of hate speech and slurs, but automatic detection is tough due to linguistic complexity, data imbalance and limited resources. We address this challenge in the BLP-2025 shared task by combining Token-Aware Adversarial Contrastive Training (TACT) with Layer-wise Learning Rate Decay (LLRD) to fine-tune transformer models like BanglaBERT, MuRIL, mE5-base and Twitter XLM-R. To capture the complementary strengths of each model, we aggregate the model outputs through logits ensembling and get a robust system for multiclass classification. On the official test set, our model achieved F1 scores of 0.7362 for hate type, 0.7335 for severity, and 0.7361 for target ranking, placing it **1st, 2nd, and 3rd,** respectively. The findings indicate that adversarial fine-tuning with logits ensemble learning is a robust way to detect hate speech in resource-limited languages and provides valuable insights for multilingual and low-resource NLP research.

1 Introduction

The rapid growth of social content has resulted in a large amount of user-posted data – particularly comments, which express public opinion. Because these platforms are anonymous and have a huge audience, bad information like hate speech and slurs has spread quickly. The problem is crucial for Bengali-language social media since Bangla is being used largely in both Bangladesh and the Indian subcontinent, and still, there aren't any good methods for finding hate speech (Raihan et al., 2023; Zampieri et al., 2023). The particular linguistic characteristics and socio-cultural nuances of Bangla make hate speech analysis challenging in the context of informal and dynamic social media language, which they adopt (Saha et al., 2023).

The detection of Bengali hate speech has become a critical need in the current situation. While existing models for detecting hate speech are effective in languages like English, they fail to generalize well to Bangla due to the distinct linguistic structures and lack of vocabulary data and advanced techniques for domain-specific models. As a result, there is a pressing demand for different tailored approaches to the detection of hate speech in the Bengali language properly, particularly considering its linguistic peculiarities and cultural context (Raihan et al., 2023; Saha et al., 2023).

To address this gap, we propose a novel approach for Bengali hate speech detection that combines advanced fine-tuning techniques with state-of-the-art transformer models and adversarial contrastive training to develop a robust system.

Our contributions are given below:

- Fine-tuned BanglaBERT, mE5-base, MuRIL, and Twitter XLM-R using Layer-wise Learning Rate Decay (LLRD) for efficient fine-tuning across model layers to perform better in specific tasks.
- Proposed a novel approach that combines fine-tuned Transformer models with LLRD and Token-level Adversarial Contrastive Learning (TACT) using the Fast Gradient Method (FGM).
- Applied logits ensembling for multiclass classification, combining model outputs' logits to improve accuracy in detecting diverse hate speech categories.
- Benchmarked our approach on development and test datasets, demonstrating superior performance with various model combinations and evaluation metrics.

Codes are available in the GitHub repository [1].

*Equal contribution.

[1]https://github.com/ShifatIslam/BLP25-Task-1

Proceedings of the Second Workshop on Bangla Language Processing (BLP-2025), pages 513–522

2 Related Work

Hate speech in Bangla has been studied using transformer-based approaches, deep learning (DL), and machine learning (ML). Despite using ML techniques like SVM, Naïve Bayes, Random Forests with TF–IDF and n-grams along with lexicon-based approaches, earlier research had difficulty with contextual richness (Alkomah and Ma, 2022; Al Maruf et al., 2024). With explainable systems like DeepHateExplainer, which integrates BanglaBERT, mBERT, and Twitter XLM-R for better interpretability, DL models like CNNs, LSTMs, and hybrid Conv-LSTMs improved sequential modeling and reduced feature engineering (Karim et al., 2021). The introduction of transformers led to significant advancements: BanglaBERT achieved state-of-the-art performance on multiclass detection of political, religious, gender, and personal hate (Islam et al., 2025), and misogyny-focused detection showed the effectiveness of BanglaBERT, mBERT, XLM-R, Electra, and DistilBERT (Mondal et al., 2025). BanTH created the first multi-label dataset for transliterated Bangla that included encoder baselines and LLM prompts (Haider et al., 2024). A recent study has emphasized adversarial and label-aware contrastive training for Bengali multiclass classification (Swarnali et al., 2024), token-aware contrastive learning (Su et al., 2021), and multimodal transformer frameworks that combine BERT with CLIP and UNITER for meme hate speech detection (Kapil and Ekbal, 2025). However, surveys always show that issues with transliteration, generalization, and dataset scarcity remain (Alkomah and Ma, 2022; Al Maruf et al., 2024).

3 System Description

3.1 Task Description

The objective of the Bangla Multi-task Hate Speech Identification shared task (Hasan et al., 2025b) is to improve hate speech detection in Bengali through three subtasks. This task uses a multi-task learning framework to train models to classify hate speech into type, severity and target group instead of a single task.

3.2 Dataset Description

The dataset utilized in this research was derived from the BLP Workshop Task (Hasan et al., 2025a), focused on Bangla Multi-task Hatespeech Identification in Bengali. Each dataset was divided into three parts: the train set, the dev set, and the test set, and each set contains 35522, 2512, and 10200 samples, respectively.

Each sample in the dataset has 3 fields: **id, text,** and **label** with the test set excluding the label column shown in Tables 3, 4, and 5. The id column was a unique identifier for each sample. The text column had the Bengali text, which was an example of hate speech meant to be classified. The label column contained the class names representing a category in the hate speech. The task of the Bangla Multi-task Hate Speech Identification was divided into 3 subtasks, and all the tasks had a common literature, except the label column, which differed from task to task. The data distribution is given in Appendix A Figure 3.

4 Method Description

4.1 Token-Aware Adversarial Contrastive Training (TACT)

At the token-embedding level, TACT (Huang et al., 2021) is implemented as adversarial training using a single-step FGM inside a custom TACT Trainer. It enhances model robustness by introducing adversarial perturbations to input embeddings.

The process begins by calculating the **clean loss** L_{clean}, which is the negative log-likelihood of the true label y given the predicted probability distribution $p_\theta(y|x)$ for the input x:

$$L_{\text{clean}} = -\log p_\theta(y|x) \tag{1}$$

Next, the **gradient** G of the clean loss with respect to the input embeddings $E(x)$ is computed:

$$G = \nabla_{E(x)} L_{\text{clean}} \tag{2}$$

A **perturbation** R is then calculated by scaling the gradient, ensuring it is norm-bounded:

$$R = \epsilon \frac{G}{\|G\|_F} \tag{3}$$

where ϵ controls the magnitude of the perturbation and $\|G\|_F$ is the Frobenius norm of the gradient.

The **adversarial embeddings** $E_{\text{adv}}(x)$ are generated by adding the perturbation to the original embeddings:

$$E_{\text{adv}}(x) = E(x) + R \tag{4}$$

The **adversarial loss** L_{adv} is then computed using the adversarial embeddings:

$$L_{\text{adv}} = -\log p_\theta(y|x; E_{\text{adv}}(x)) \tag{5}$$

Finally, the total objective function L is a weighted sum of the clean loss and the adversarial loss, with λ controlling the balance between the two:

$$\boxed{\mathcal{L} = \mathcal{L}_{\text{clean}} + \lambda \mathcal{L}_{\text{adv}}} \tag{6}$$

where λ is a hyperparameter that determines the importance of adversarial training.

Equations 1--6 describe the objective employed in TACT using FGM. The clean loss L_{clean} is computed from the standard cross-entropy on clean data. The gradient of the clean loss is used to generate adversarial examples, and the adversarial loss L_{adv} is computed using the perturbed embeddings. The total loss L is a weighted sum of the clean and adversarial losses, with λ controlling the trade-off between them.

4.2 LLRD

LLRD (Ishii and Sato, 2017) makes transformer fine-tuning more stable by giving lower layers smaller learning rates and upper layers larger rates. Parameters are organized by depth, such as embeddings, encoder layers, and the classifier head. Each group is then optimized with its own learning-rate "bucket."

For an encoder with L layers indexed from bottom to top by $l \in \{0, \ldots, L-1\}$, the learning rate η_l for each layer is given by:

$$\eta_l = \eta_0 \alpha^{L-1-l}, \quad l = 0, \ldots, L-1 \tag{7}$$

where η_0 is the base learning rate applied to the top layer, $\alpha \in (0, 1)$ is the decay factor, and L is the total number of layers. Equation 7 ensures that the learning rate decreases progressively from the top layers to the bottom layers, with the decay factor α controlling the rate at which this reduction occurs.

For the embedding block, the learning rate is calculated separately:

$$\eta_{\text{emb}} = \eta_0 \alpha^L \tag{8}$$

This learning rate η_{emb} applies to the embedding layer, which is smaller than the learning rates of the upper layers, as it follows the same decay pattern.

4.2.1 AdamW with group-wise decay and LLRD

Let $\{\mathcal{G}_l\}$ be parameter groups aligned with depth l (plus an embedding group), and let $\mathbf{1}_{\text{decay}}(w) \in \{0, 1\}$ mask weight decay (e.g., $\mathbf{1}_{\text{decay}}(w)$=0 for biases and LayerNorm scales). The optimization objective is

$$\min_\theta \; E\big[\mathcal{L}(\theta)\big] + \sum_l \lambda_l \sum_{w \in \mathcal{G}_l} \mathbf{1}_{\text{decay}}(w) \, \|w\|_2^2 \tag{9}$$

with per-group step sizes set by the LLRD schedule:

$$\eta(\mathcal{G}_l) = \eta_l, \qquad \eta(\text{emb}) = \eta_{\text{emb}} \tag{10}$$

The Equation 7,8 explain LLRD, where the learning rate for each encoder layer (l) goes down based on a decay factor (α) and the total number of layers (L). The embedding block's learning rate is set to (η_{emb}) and is also decreased based on (α^L). The AdamW optimizer only applies weight decay to some parameters, and the total optimization objective is the loss function ($\mathcal{L}(\theta)$) plus the weight decay regularization in Equation 9, and 10. The LLRD schedule sets the step sizes for each group of parameters.

4.3 Logits Generation and Ensemble Technique

Logits generation involves getting raw output scores from each model for each sample in the test or validation set. For each input x_i, each model m produces logits $z_m(x_i)$, which are used to make the final predictions shown in Equation 11.

$$z_m(x_i) = f_m(B_i), \quad \forall x_i \in D_{\text{test}} \tag{11}$$

where f_m represents the function (or model) m applied to the input batch B_i. The logits from the models are then aggregated using a weighted sum to form the ensemble logits:

$$z_{\text{ens}}(x_i) = \sum_{m \in M'} w_m z_m(x_i) \tag{12}$$

where M' denotes the subset of models used in the ensemble, and w_m is the learned weight for each model. The final prediction for each subtask is obtained by applying the **argmax** function to the ensembled logits:

$$\hat{y}_i = \text{argmax}(z_{\text{ens}}(x_i)) \tag{13}$$

Model	Hate-type without TACT+LLRD	Hate-type TACT+LLRD	To-whom without TACT+LLRD	To-whom TACT+LLRD	Hate-severity without TACT+LLRD	Hate-severity TACT+LLRD
BanglaBERT	0.7013	0.7265	0.6914	0.7277	0.7270	0.7431
MuRIL	0.6992	0.7179	0.7087	0.7143	0.7022	0.7305
mE5-base	0.7122	0.7220	0.6961	0.7105	0.7228	0.7303
Twitter XLM-R	0.6986	0.7100	0.7089	0.6917	0.7024	0.7232

Table 1: Performance comparison of our Models on F1 score

Ensembling $z_{\text{ens}}(x_i)$combines the logits from different model combinations. To get the final prediction for each subtask, use$\arg\max$ of the ensembled logits shows in Equation 12, 13.

4.4 Our Approach

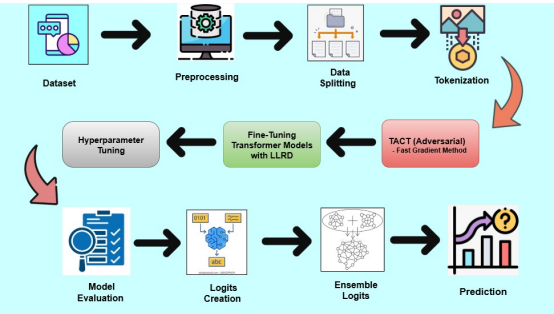


Figure 1: Workflow Diagram of our proposed methodology

Figure 1 illustrates our framework for multi-class Bengali hate speech detection. The first step is preprocessing (cleaning, normalizing, and splitting the dataset into groups), and then tokenization to ensure uniform input representation. We add TACT with FGM during training to improve accuracy and get better results. Furthermore, we use AdamW with LLRD to fine-tune BanglaBERT and other Transformer models (Multilingual-E5, MuRIL, Twitter XLM-R). In the end, we perform hyperparameter tuning, and the logits from each model are standardized and combined through an ensemble just by adding the logits, which makes the predictions more accurate and reliable. The algorithm of our whole process is shown in Appendix C.

5 Result Analysis

As defined in our methodology 4.4, **TACT with LLRD** has performed significantly well, outperforming benchmark results of separate models, as shown in Table 1. Ensembling different models' logits through aggregation, which were standardised, further improved the performance. This ensemble technique is showed in equation 12. Using fixed, uniform ensemble weights (i.e., $w_m = 1$ for all models) further improved our performance across **Subtasks 1A, 1B, and 1C**. The results are presented in Tables 9, 10, and 11, which summarize all ensemble combinations evaluated in our experiments. The combinations for which we got the peak scores in each task are shown in Table 2. However, we also performed other methods.
Instead of aggregating the results, we tried to use a neural network to learn the weights w_m. However, it could not achieve the peak accuracy as shown in Table 13. To address the dataset imbalance, we also experimented with a two-step classification approach. First, we trained a binary classifier to distinguish between None and Not None, achieving an accuracy of 0.7718 for this binary task. In the second step, only the instances classified as Not None were further assigned to the remaining classes. However, this pipeline resulted in an overall accuracy of only 0.7127.
Despite these approaches, we also tried to mitigate the imbalance of the dataset with other methods, which are shown in Table 12 with accuracy. However, none of these models could beat the superior result of our novel approach.

Sub task	Class	Ensemble Model	F1-score
1A	Hate-type	BanglaBERT MuRIL mE5-base	0.7362
1B	To-whom	BanglaBERT MuRIL mE5-base	0.7335
1C	Hate-type To-whom Hate-severity	BanglaBERT + MuRIL + mE5-base BanglaBERT + mE5-base BanglaBERT + MuRIL + mE5-base	0.7361

Table 2: F1-scores of the best-performing ensemble combinations for all subtasks.

Our initial pool included seven pretrained encoder models chosen for their ability to capture the nuances and intricacies of Bengali. We found 4 models which were consistent among the tasks shown in Table 8. Although Twitter XLM-R had a lower accuracy than some of the models, as shown in the table, it was chosen because of its superior performance in other tasks.

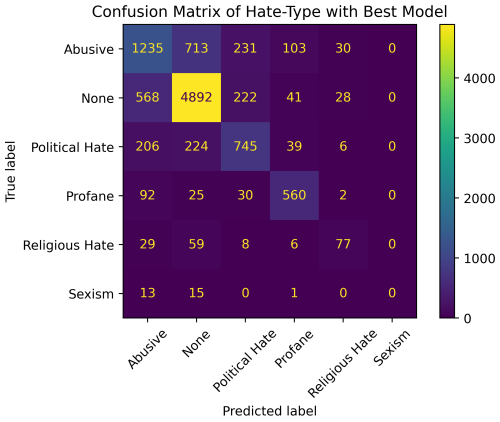


Figure 2: Confusion Matrix for Subtask 1A (Hate Type) using the Best-Performing Ensemble Model

5.1 Error Analysis

In Figure 2, we can see the classification report of the ensemble model (BanglaBERT + MuRIL + mE5-base), which achieved the top performance score in the leaderboard for the hate-type class. The ensemble models performed well for the None (85.06%) and Profane (78.98%) classes, but not very well for the Political Hate (61.06%) and Abusive (58.69%) classes. Also, the results for Religious Hate and Sexism were not consistent; for example, there were no correct predictions for Sexism. This difference is mostly because the datasets were not balanced, since these classes had a lot fewer samples. In fact, a direct comparison between class frequency and per-class F1-score shows that classes with more samples achieve higher performance, while underrepresented classes—such as Religious Hate, and Sexism—consistently perform worse. Thus, data imbalance can largely be attributed to this underperformance. Similar reasoning can be made for the other classes, and this imbalance of the dataset will be the reason for the inferior performance for the other classes.

However, data imbalance is not the sole reason for poor performance; the dataset contains samples that may fit multiple classes due to intertwined semantics. Certain Bengali sentences have overlapping meanings, causing ambiguity that remains unresolved even through human evaluation, as shown in Table 14.

6 Conclusion

This research developed a comprehensive framework for detecting Bangla hate speech by employing TACT with LLRD on transformer models, which was augmented by logits ensembling. The approach achieved the highest score, ranking first in subtask A, second in subtask B, and third in subtask C. Despite existing challenges such as imbalanced datasets and linguistic disparities, the suggested technique represents a commendable initial step towards enhancing hate speech identification in resource-scarce languages. It also provides valuable insights for other multilingual initiatives, paving the way for future works, making languages accessible and communication easier.

Limitations

There were several limitations in our work. The dataset, which was provided, had a small size and was highly imbalanced, as shown in the Figure 3. This imbalance had a lasting effect on our results, and despite trying a lot of approaches, the imbalance was noteworthy. Secondly, we chose 7 initial models based on their applicability in Bengali. These models were carefully curated. However, there may be further models that can be explored. Thirdly, the dataset had some mislabeled data, as shown in the error analysis, which had a detrimental effect on the accuracy.

References

Abdullah Al Maruf, Ahmad Jainul Abidin, Md Mahmudul Haque, Zakaria Masud Jiyad, Aditi Golder, Raaid Alubady, and Zeyar Aung. 2024. Hate speech detection in the bengali language: a comprehensive survey. *Journal of Big Data*, 11(1):97.

Fatimah Alkomah and Xiaogang Ma. 2022. A literature review of textual hate speech detection methods and datasets. *Information*, 13(6):273.

Fabiha Haider, Fariha Tanjim Shifat, Md Farhan Ishmam, Deeparghya Dutta Barua, Md Sakib Ul Rahman Sourove, Md Fahim, and Md Farhad Alam. 2024. Banth: A multi-label hate speech detection dataset for transliterated bangla. *arXiv preprint arXiv:2410.13281*.

Md Arid Hasan, Firoj Alam, Md Fahad Hossain, Usman Naseem, and Syed Ishtiaque Ahmed. 2025a. Llm-based multi-task bangla hate speech detection: Type, severity, and target. *arXiv preprint arXiv:2510.01995*.

Md Arid Hasan, Firoj Alam, Md Fahad Hossain, Usman Naseem, and Syed Ishtiaque Ahmed. 2025b. Overview of blp 2025 task 1: Bangla hate speech identification. In *Proceedings of the Second International Workshop on Bangla Language Processing*

(BLP-2025), India. Association for Computational Linguistics.

Qiushi Huang, Tom Ko, H Lilian Tang, Xubo Liu, and Bo Wu. 2021. Token-level supervised contrastive learning for punctuation restoration. *arXiv preprint arXiv:2107.09099*.

Masato Ishii and Atsushi Sato. 2017. Layer-wise weight decay for deep neural networks. In *Pacific-Rim Symposium on Image and Video Technology*, pages 276--289. Springer.

Md Samiul Islam, Rifat Nawaz, Jannatul Ferdous Salma, Md Kamrul Islam, Mahabubur Rahman, Sapayev Valisher Odilbek, and Muhabbat Jumaniyozova. 2025. Leveraging banglabert: A transformer-based approach for multiclass hate speech detection in bangla social media. In *2025 4th International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, pages 487--492. IEEE.

Prashant Kapil and Asif Ekbal. 2025. A transformer based multi task learning approach to multimodal hate speech detection. *Natural Language Processing Journal*, 11:100133.

Md Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Md Azam Hossain, and Stefan Decker. 2021. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *2021 IEEE 8th international conference on data science and advanced analytics (DSAA)*, pages 1--10. IEEE.

Snaholata Mondal, Md Samiul Alom, Md Mahbub Alum, and Kazi Lamia Sinja Sunjida. 2025. Multiclass detection of misogynistic bangla text from youtube using transformer-based models. In *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1--6. IEEE.

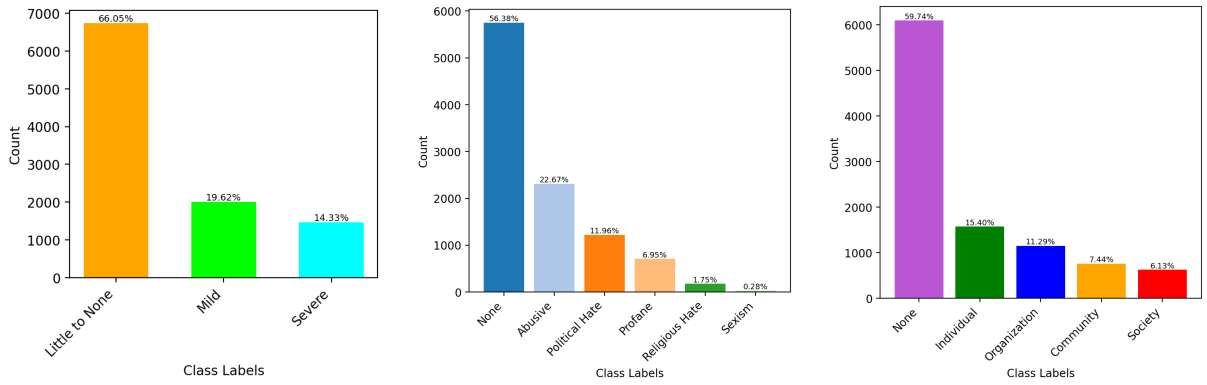
Md Nishat Raihan, Dhiman Goswami, Sadiya Sayara Chowdhury Puspo, and Marcos Zampieri. 2023. nlpbdpatriots at blp-2023 task 1: A two-step classification for violence inciting text detection in bangla. In *The First Workshop on Bangla Language Processing (BLP-2023)*, page 179.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahouti, Syed Ishtiaque Ahmed, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the First Workshop on Bangla Language Processing (BLP-2023)*, pages 72--84.

Yixuan Su, Fangyu Liu, Zaiqiao Meng, Tian Lan, Lei Shu, Ehsan Shareghi, and Nigel Collier. 2021. Tacl: Improving bert pre-training with token-aware contrastive learning. *arXiv preprint arXiv:2111.04198*.

Farhana Hossain Swarnali, Jannatim MaishaL, Muhammad Azmain Mahtab, M Saymon Islam Iftikar, and Faisal Muhammad Shah. 2024. Bengali multi-class text classification via enhanced contrastive learning techniques. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 1576--1581. IEEE.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Cagrı Cöltekin. 2023. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020).



(a) Hate Severity Distribution (b) Hate Type Distribution (c) Target Distribution

Figure 3: Dataset class distributions across severity, type, and target dimensions.

A Dataset Samples

The given dataset is for hate speech in Bangla across severity, type and target. As shown in Figure 3, severity is biased towards less harmful speech, and type is towards abuse and religious hate. The target is towards individuals more than organizations or communities.

id	text	label
432313	আসলে মহান নেতা এটাই তার পরিচয়	None
359516	ইরান ধ্বংশ হউক	Abusive
578332	আলহামদুলিল্লাহ দেশ এগিয়ে যাচ্ছে বিএনপি জামাতীদের জুতা পেটা করতে হবে	Political Hate
404893	সময় টিভি একটা জাউড়া মিডিয়া মিথ্যা তথ্য প্রচার করে বেড়ায়	Profane
764029	ইহুদি নাসাদের শিক্ষা মুসলমানদের জন্য হারাম	Religious Hate
639002	স্লোগানে আমি প্রথমে যৌন নেত্রীর আগমন শুনছি	Sexism

Table 3: Examples of hate type dataset samples.

id	text	label
165894	হেন কাপ পুলিশের মারে অন্যরা তাহলে পলিশের কি হবে বিচার হবে কি	None
587800	ওনি আমার বালের ওলি বালের ভান্ডারী কুত্তার বাচ্চারা সব ভন্ড	Individual
241030	ভারতীয় দালাল সময় টিভিকে বয়কট করুন	Organization
124999	আল্লাহ্ এসব জানোয়ারদের শেষ করে দাও	Community
12764	ইজরায়েলের বিচার হওয়া উচিৎ	Society

Table 4: Examples of to whom dataset samples.

id	text	label
165894	হেন কাপ পুলিশের মাঝে অন্যরা থাকলে পুলিশের কি হবে বিচার হবে কি	Little to None
814896	হালার এই দেশে বড় আইনের ফুজিটিভদের বাসা বাড়িতে দিয়ে দেয় খালাসন মাজার জন্য	Mild
124999	আল্লাহ এইসব জানোয়ারদের শেষ করে দাও	Severe

Table 5: Examples of hate severity dataset samples.

B Baseline & Observations

There were several noteworthy observations. We found that during validation, the model chosen with the best validation accuracy resulted in a better overall model. In the ensemble method, we found that logits, when standardized, had a better impact on the score. Also, all the approaches that we tried for a single model had a poorer result than Banglabert, which was consistently the best model throughout the tasks.

The organisers have also provided baseline results for this task on both the Dev-Test and Test Datasets. Three different models were used: the Random Baseline, Majority Baseline, and the n-gram Baseline. As shown in the Table 7, our model did much better than all of the baselines on both the Test and Dev sets. On the Test set, it got micro-F1 scores of 0.7362, 0.7335, and 0.7361 across Subtasks 1A, 1B, and 1C. On the Dev set, it got even better scores of 0.7579, 0.7531, and 0.7558.

C Algorithm

Algorithm 1: Multitask Bangla Hate Speech Detection with FGM (TACT) and LLRD

Input: $\mathcal{D} = \{(x_i, y_i^{(A)}, y_i^{(B)}, y_i^{(C)})\}_{i=1}^{N}$; model set $\mathcal{M}$; FGM radius ϵ; mix weight λ_{adv}; LLRD decay $\alpha \in (0, 1)$; train ratio r=0.7

Output: Prediction labels for 1A (type), 1B (severity), 1C (target)

Preprocess & split: Clean/normalize texts; Split $\mathcal{D} \to \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}$ with $|\mathcal{D}_{\text{train}}|/N \approx r$.;

foreach *model* $m \in \mathcal{M}$ **do**

  Tokenize with m's tokenizer ;

  Build AdamW with **LLRD** parameter groups: for encoder layer l=0 . . . L−1, set $\eta_l \leftarrow \eta_0\, \alpha^{L-1-l}$.;;

  embeddings $\eta_{\text{emb}} \leftarrow \eta_0\, \alpha^{L}$; (no-decay for biases/LN).;

  for *epoch* $= 1 \ldots E$ **do**

    foreach *mini-batch* $\mathcal{B} \subset \mathcal{D}_{train}$ **do**

      // Clean forward & loss

      Get logits $z = f_\theta(\mathcal{B})$; $\mathcal{L}_{\text{clean}} = \text{CE}(\text{softmax}(z), y)$.;

      // FGM perturbation (TACT)

      $G = \nabla_E \mathcal{L}_{\text{clean}}$;;

      $R = \epsilon\, G/\|G\|_F$;;

      set $E_{\text{adv}} = E + R$.;

      Get $z_{\text{adv}} = f_\theta(\mathcal{B}; E_{\text{adv}})$; $\mathcal{L}_{\text{adv}} = \text{CE}(\text{softmax}(z_{\text{adv}}), y)$.;

      // Total loss & update

      $\mathcal{L} = \mathcal{L}_{\text{clean}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}$; update θ with LLRD rates.;

    Validate on $\mathcal{D}_{\text{val}}$; keep best checkpoint.;

  Generate/store per-subtask logits on dev/test.;

Ensemble: For each sample, combine logits across chosen $\mathcal{M}' \subseteq \mathcal{M}$.

This algorithm depicts the whole approach where we fine-tune transformer models for Bangla hate speech detection using TACT and LLRD. In each epoch tokenized text is perturbed and passed through the model to compute the clean and adversarial loss, and then the parameters are updated. Finally logits are generated for each subtask (hate type, severity, target), and an ensemble is applied for prediction.

D Experimental Setup and Hyperparameter

We fine-tuned transformer-based encoders like BanglaBERT, MuRIL, multilingual E5-base, and Twitter XLM-R on the Bangla hate speech dataset for three different tasks. We tokenized the preprocessed text and put it into models that had been trained with AdamW and Layer-wise Learning Rate Decay (LLRD). Token-Aware Adversarial Contrastive Training (TACT) was employed with minor adjustments at the embedding level to enhance the system's robustness.

Hyperparameter	Value
Number of epochs	5
Train batch size (per device)	16
Eval batch size (per device)	16
Learning rate	2e-5
Weight decay	0.01
Warmup ratio	0.1

Table 6: Selected hyperparameters used for model training.

We trained for five epochs with a batch size of 16, a learning rate of 2e-5, a weight decay of 0.01, and a warm-up ratio of 0.1 as shown in Table 6. To make sure that models were stable, hyperparameters were tuned within small ranges. Final predictions were made by combining the logits of different models based on how well they did on the development set.

Subtask 1A Model	micro-F1	Subtask 1B Model	micro-F1	Subtask 1C Model	weighted micro-F1
Random Baseline	0.1638	Random Baseline	0.2043	Random Baseline	0.2304
Majority Baseline	0.5638	Majority Baseline	0.5974	Majority Baseline	0.6072
n-gram Baseline	0.6020	n-gram Baseline	0.6209	n-gram Baseline	0.6305
Our Model (TestSet)	**0.7362**	Our Model (TestSet)	**0.7335**	Our Model (TestSet)	**0.7361**
Our Model (DevSet)	**0..7579**	Our Model (DevSet)	**0.7531**	Our Model (DevSet)	**0.7558**

Table 7: Comparison with the baseline

Model Name	Model name (Short)	F1-score (Dev Test)
csebuetnlp/banglabert	BanglaBERT	**0.7389**
google/muril-base-cased	MuRIL	0.7281
intfloat/multilingual-e5-base	mE5-base	0.7253
cardiffnlp/twitter-XLM-R-base-sentiment	Twitter XLM-R	0.7134
sagorsarker/bangla-bert-base	sagorsarker_bert	0.7054
distilbert-base-multilingual-cased	distilbert	0.7166
FacebookAI/roberta-base	xlm-roberta	0.7209

Table 8: Performance of different pretrained models on the development test set.

Task Name	Ensemble Model Combination	F1-Score
Subtask 1A (Hate Type)	BanglaBERT	0.7265
	MuRIL	0.7179
	mE5-base	0.7220
	Twitter XLM-R	0.7100
	BanglaBERT + MuRIL	0.7336
	BanglaBERT + mE5-base	0.7358
	BanglaBERT + Twitter XLM-R	0.7299
	MuRIL + mE5-base	0.7277
	MuRIL + Twitter XLM-R	0.7246
	mE5-base + Twitter XLM-R	0.7209
	BanglaBERT + MuRIL + mE5-base	**0.7361**
	BanglaBERT + MuRIL + Twitter XLM-R	0.7323
	BanglaBERT + mE5-base + Twitter XLM-R	0.7331
	MuRIL + mE5-base + Twitter XLM-R	0.7292
	BanglaBERT + MuRIL + mE5-base + Twitter XLM-R	0.7327

Table 9: F1-scores of different ensemble model combinations for Subtask 1A (Hate Type). The best-performing score is highlighted in bold.

Task Name	Ensemble Model Combination	F1-Score
Subtask 1B (To-whom)	BanglaBERT	0.7277
	MuRIL	0.7143
	mE5-base	0.7105
	Twitter XLM-R	0.6917
	BanglaBERT + MuRIL	0.7314
	BanglaBERT + mE5-base	0.7319
	BanglaBERT + Twitter XLM-R	0.7312
	MuRIL + mE5-base	0.7312
	MuRIL + Twitter XLM-R	0.7255
	mE5-base + Twitter XLM-R	0.7193
	BanglaBERT + MuRIL + mE5-base	**0.7335**
	BanglaBERT + MuRIL + Twitter XLM-R	0.7334
	BanglaBERT + mE5-base + Twitter XLM-R	0.7299
	MuRIL + mE5-base + Twitter XLM-R	0.7268
	BanglaBERT + MuRIL + mE5-base + Twitter XLM-R	0.7330

Table 10: F1-scores of different ensemble model combinations for Subtask 1B (To-whom). The best-performing score is highlighted in bold.

Task Name	Ensemble Model Combination (Hate Type)	Ensemble Model Combination (To-Whom)	Ensemble Model Combination (Hate Severity)	F1-Score
Subtask 1C (Hate-Type, To-Whom, Hate-Severity)	BanglaBERT + Multilingual	BanglaBERT + MuRIL	BanglaBERT + Multilingual	0.7345
	BanglaBERT + MuRIL + Multilingual	BanglaBERT + Multilingual	BanglaBERT + MuRIL + Multilingual	0.7342
	BanglaBERT + MuRIL	BanglaBERT + MuRIL + Multilingual	BanglaBERT + Multilingual	0.7356
	BanglaBERT + Multilingual	BanglaBERT + Multilingual	BanglaBERT + Multilingual	0.7353
	BanglaBERT + MuRIL + Multilingual	BanglaBERT + Multilingual	BanglaBERT + MuRIL + Multilingual	**0.7361**
	BanglaBERT + MuRIL	BanglaBERT + MuRIL + Multilingual	BanglaBERT + MuRIL + Multilingual	0.7358
	BanglaBERT + Multilingual	BanglaBERT + Multilingual	BanglaBERT + Multilingual	0.7341
	BanglaBERT + MuRIL + Multilingual	BanglaBERT + MuRIL	BanglaBERT + MuRIL + Multilingual	0.7328
	BanglaBERT + MuRIL	BanglaBERT + Multilingual	BanglaBERT + Multilingual	0.7352
	BanglaBERT + Multilingual	BanglaBERT + MuRIL + Multilingual	BanglaBERT + Multilingual	0.7349
	BanglaBERT + MuRIL + Multilingual	BanglaBERT + Multilingual	BanglaBERT + MuRIL + Multilingual	0.7357
	BanglaBERT + Multilingual	BanglaBERT + MuRIL	BanglaBERT + Multilingual	0.7354

Table 11: F1-scores of different ensemble model combinations for Subtask 1C (Hate-Type, To-Whom, Hate-Severity). The best-performing score is highlighted in bold.

Serial No	Model	Validation Accuracy
1	Binary classification	0.7127
2	LSTM with Banglabert logits	0.5820
3	Class-Weighted Loss	0.6644
4	Class-Weighted Loss with TACT	0.6835
5	Random Over-Sampling	0.7026
6	Focal Loss	0.6696
7	Banglabert	0.7229

Table 12: Validation accuracy of other approaches.

Serial No	Model	F1-score (Validation Test)
1	Linear Neural Network with 3 logits	0.7245
2	Multilayer perceptron with 3 logits	0.7258
3	Linear Neural Network with 2 logits	0.7365
4	Multilayer perceptron with 2 logits	0.7372
5	Our best ensemble model	**0.7580**

Table 13: F1-scores of different ensemble approaches on validation test.

ID	Text	Label	Correct Label
891820	আওয়ামী লীগের বাকশালী রাজনীতি ঢুকে গেছে ক্রিকেট টিমে	Abusive	Political Hate
272543	চাচি তুমি ভোট চোর	Political Hate	Abusive
269938	কি উন্নয়ন আলু ৭০ পেয়াজ ১৪০ চাল ৭০ হেইয়ো উন্নয়ন	Political Hate	None
428911	এই সব বিক্ষোভ আন্দোলন ইসরাইল ভয় পায় না	Religious Hate	Political Hate
813236	তীব্র গরম জাহান্নামের নিঃশ্বাস	Religious Hate	None
165501	মেয়েদের কোথাও নিরাপদ নেই	Sexism	None
514297	এ শলায় মেয়ে লোকের দালাল শাকিব খান	None	Abusive

Table 14: Original and proposed classes for possible misclassified samples in Subtask A (Hate-type).