Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.058329

ARTICLE

Optimizing BERT for Bengali Emotion Classification: Evaluating Knowledge Distillation, Pruning, and Quantization

Md Hasibur Rahman, Mohammed Arif Uddin, Zinnat Fowzia Ria and Rashedur M. Rahman*

Department of Electrical and Computer Engineering, North South University, Dhaka, 1229, Bangladesh

*Corresponding Author: Rashedur M. Rahman. Email: rashedur.rahman@northsouth.edu

Received: 10 September 2024 Accepted: 22 November 2024 Published: 27 January 2025

ABSTRACT

The rapid growth of digital data necessitates advanced natural language processing (NLP) models like BERT (Bidirectional Encoder Representations from Transformers), known for its superior performance in text classification. However, BERT's size and computational demands limit its practicality, especially in resource-constrained settings. This research compresses the BERT base model for Bengali emotion classification through knowledge distillation (KD), pruning, and quantization techniques. Despite Bengali being the sixth most spoken language globally, NLP research in this area is limited. Our approach addresses this gap by creating an efficient BERT-based model for Bengali text. We have explored 20 combinations for KD, quantization, and pruning, resulting in improved speedup, fewer parameters, and reduced memory size. Our best results demonstrate significant improvements in both speed and efficiency. For instance, in the case of mBERT, we achieved a 3.87× speedup and 4× compression ratio with a combination of Distil + Prune + Quant that reduced parameters from 178 to 46 M, while the memory size decreased from 711 to 178 MB. These results offer scalable solutions for NLP tasks in various languages and advance the field of model compression, making these models suitable for real-world applications in resource-limited environments.

KEYWORDS

Bengali NLP; black-box distillation; emotion classification; model compression; post-training quantization; unstructured pruning

1 Introduction

The exponential growth of digital data has driven the need for advanced natural language processing (NLP) models to understand human language accurately. Bidirectional Encoder Representations from Transformers (BERT) stands out for its performance across various NLP tasks, including text classification and question answering [1]. Its bidirectional context modeling enables a deep understanding of language. Still, the model's immense size—110 million parameters in its base version—creates challenges for deployment in resource-constrained environments like mobile devices or real-time applications, where memory and processing power are limited.

These challenges are even more pronounced for Bengali, the sixth most spoken language globally, with over 230 million native speakers. Despite its rich literary heritage, Bengali-specific NLP models lag behind those for more widely spoken languages, partly due to the limited availability of large-scale text corpora and pre-trained models. Thus, there is a need for efficient, high-performing models tailored for Bengali that can be deployed in resource-constrained environments.

This research addresses these challenges by compressing the BERT base model through knowledge distillation (KD). Quantization, and pruning [2–4]. KD transfers knowledge from a large model (teacher) to a smaller model (student), preserving performance while reducing size. Pruning eliminates redundant parameters, enhancing efficiency. Quantization lowers the precision of model parameters, cutting memory and computational demands.

The goal is to develop a compressed, efficient BERT-based model for Bengali text classification that balances size reduction with performance. The study also explores the combined effects of these compression techniques, contributing to model compression research and advancing NLP for underrepresented languages like Bengali.

Unique Contributions:

1. **Language-Specific Compression:** This research addresses the challenges and requirements of compressing BERT models for Bengali, a language relatively underrepresented in NLP research compared to English and other widely spoken languages.
2. **Comprehensive Evaluation:** By systematically applying and evaluating three compression techniques in 20 combinations, the research provides a comprehensive understanding of how these methods interact and their effectiveness for compression in Bengali text classification.
3. **Resource Efficiency:** The resulting compressed models will enable practical applications of BERT for Bengali text classification in resource-constrained environments, thereby broadening the accessibility and usability of advanced NLP models.

The remainder of the paper is structured as follows: Section 2 highlights the key characteristics of relevant studies on BERT models and knowledge compression techniques, additionally providing a concise summary table. Subsequently, Section 3 outlines our proposed methodology. Section 4 presents the experimental setup and analysis of results, and Section 5 provides the discussion. Sections 6 and 7 highlight limitations, future work, and conclusions.

2 Related Work

NLP tasks have revolutionized recently because of transformer-based LLMs (Large Language Models) [5]. The discipline of NLP was significantly advanced by the ground-breaking model BERT, which established a bidirectional method for pre-training. BERT's success has led to several derivative models, each with unique enhancements and improvements, such as Roberta, ALBERT, Longformer, BigBird, and more [6,7]. However, BERT and its larger variants, like BERT-large, come with many parameters (BERT-base has 110 M parameters, while BERT-large has 340 M). These large models often require extensive computational resources, making them challenging to deploy on resource-constrained devices such as mobile phones or edge devices. In recent years, various model compression techniques have been developed to enhance the efficiency of LLMs, particularly for deployment on resource-constrained devices. TinyBERT and DistilBERT are two notable examples of compressed versions of BERT, which reduce the model size while maintaining competitive performance. These approaches make it feasible to use transformer models in applications with limited computational power and memory.

TinyBERT is a compact and fast language model derived from distilling BERT-Base for natural language understanding [8]. It is 9.4 times quicker and 7.5 times smaller than BERT-Base in inference and performs competitively on several NLP tasks. TinyBERT is pre-trained and fine-tuned using a novel transformer distillation method, effectively transferring knowledge from BERT-Base to TinyBERT. This makes it suitable for tasks requiring efficient and accurate text understanding. DistilBERT [9] compresses the original BERT model while keeping 97% of its language comprehension capacity using a KD approach. DistilBERT is especially helpful for deployment in resource-constrained contexts without a noticeable decrease in performance because of its smaller size and faster speed. A breakthrough in transformer models designed for particular languages is BanglaBERT [10]. Because it is a BERT-based model that has been pre-trained on a sizable corpus of Bengali literature, it performs very well on NLP tasks that include the language. With BanglaBERT, we can get better performance and accuracy in tasks like named entity recognition, text classification, and sentiment analysis by taking advantage of Bengali's distinct linguistic features. Its development underscores the importance of creating language-specific models to enhance NLP capabilities across diverse linguistic contexts.

The swift progress in LLMs has resulted in extensive implementation across numerous fields, such as image classification, natural language understanding, and voice recognition. Although these models exhibit exceptional performance, implementing them on edge devices with constrained computational capabilities remains a significant hurdle. To mitigate this challenge, scholars have devised various model compression strategies, particularly KD, pruning, and quantization, to diminish LLMs' size and computational demands while maintaining their accuracy. KD is an effective method for model compression and transfer learning. It entails training a diminutive student model to emulate the functionalities of a larger, pre-trained teacher model. This approach exploits the soft output probabilities generated by the teacher model to inform the student model, thereby enhancing its performance relative to training it independently. KD has been further adapted to include intermediate representations and applied across diverse architectures, yielding notable model efficacy and precision enhancements. Pruning strategies focus on removing superfluous weights or filters from a trained model, culminating in a more compact and efficient network. Quantization methods aim to diminish the precision of weights and activations, transitioning them from high-precision (32-bit floating-point) to low-precision (8-bit integer) representations. This reduction in precision markedly lowers the model size and computational demands, facilitating more efficient deployment on edge devices.

In recent times, Knowledge distillation has emerged as a formidable technique for transferring knowledge from a large, pre-trained teacher model to a compact student model. Hinton [11] introduced this technique, wherein the student model is conditioned to replicate the teacher model's soft output (softmax). This methodology has enhanced the student model's performance compared to training it from an initial state. Romero et al. [12] expanded upon this framework by employing intermediate representations of the teacher model as informative clues, thus further augmenting the performance of the student model. Li et al. [13] present an innovative approach for knowledge distillation utilizing a minimal number of label-free samples to enhance data and training efficiency They introduce the paradigm of a "teacher-net" and a "student-net" to facilitate network compression and aim to eliminate superfluous weights or filters from an established model. Li et al. explore Online Knowledge Distillation (OKD) for effective pose estimation, transferring pose knowledge from a robust teacher model to a streamlined student model [14]. He et al. propose Few-Shot Slimming for efficient network compression, accentuating compression using limited unlabeled data [15]. Approaches such as low-rank decomposition for fully connected layers and various forms of weight decomposition for convolutional layers have been investigated. Yim et al. applied KD to the ResNet architecture by minimizing the L2 loss of Gramian feature matrices between the teacher and student networks [16]. Recent investigations in the domain of few-sample learning, encompassing one-shot and few-shot learning, have delved into generative models and transfer learning methodologies. Meta-learning strategies, which address problems in a learning-to-learn paradigm, have gained traction for adaptability. Even so, studies have specifically tackled the issue of KD with limited samples.

Pruning methodologies are designed to eliminate redundant neurons or connections within a network, thereby lessening its footprint and computational needs. Han et al. [17] introduced an iterative pruning strategy that discards connections with nearly negligible weights, followed by a fine-tuning process to restore the network's accuracy. Their magnitude-based pruning techniques have garnered acclaim for their straightforwardness and effectiveness. The authors in [18] proposed a neuron pruning method predicated on activation analysis, which removes neurons that do not significantly contribute to the network's outputs. However, pruning strategies are frequently confined to less complex network structures such as Visual Geometry Group (VGG) and AlexNet, and their application to more intricate architectures like ResNet may result in untrainable models due to inherent structural dependencies. The manuscript "Model Compression via Pruning, Quantization, and Knowledge Distillation," authored by Kim et al. [4], presents the Pruning Quantization KD (PQK) approach, which amalgamates pruning, quantization, and KD for streamlined model compression [19]. A pivotal innovation of PQK is using insignificant weights pruned during the initial phase to construct a teacher network, eliminating the need for a pre-trained teacher model. This methodology substantially enhances model performance for keyword detection and image classification tasks. This method entails the pruning of weights based on their magnitudes and has been employed in extensive networks to achieve notable reductions in model size with negligible accuracy degradation. Structured Pruning strategies, such as those delineated by Anwar et al., focus on eliminating entire structures within the network, such as neurons or channels, instead of individual weights [20]. This form of pruning proves particularly advantageous for optimizing hardware computations. The work "To Prune or Not to Prune" by Zhu et al. [21] provides a comprehensive comparison between pruned (sparse) and dense models across a range of domains, ultimately concluding that pruned models frequently surpass their dense equivalents in performance.

Quantization seeks to diminish the precision of weights and activations within neural networks, thereby reducing the model's size and improving computational efficiency. Initial quantization methods concentrated on uniform quantization, where all weights are consolidated to a lower bit-width representation. Jacob et al. proposed quantization-aware training, which incorporates quantization into the training regimen, alleviating the accuracy degradation typically linked with post-training quantization [22]. The emergence of data-free quantization has become a pivotal area of inquiry, given its applicability *in situ*ations where training data is not accessible. Nagel et al. introduced weight equalization and bias correction methodologies to enable effective quantization without original training data [23]. This strategy confronts the obstacles of quantizing pre-trained models while preserving their performance. Quantization techniques diminish the precision of weights within a network, transitioning them from high-precision formats (e.g., 32-bit floating-point) to low-precision forms (e.g., 8-bit integer). Leng et al. demonstrated that weight quantization can substantially compress CNNs without compromising accuracy [24]. Furthermore, quantization-aware training enhances the performance of quantized models by accounting for quantization errors during the training phase. Zhou et al. introduced Incremental Network Quantization (INQ). This technique incrementally quantized the weights of a neural network throughout training, enabling the network to acclimatize to the quantization process while maintaining high accuracy [25]. This methodology has shown potential in reducing the bit-width of weights while ensuring the preservation of model performance. Table 1 summarizes the findings and limitations of some relevant works.

Table 1: A summary table of related works highlighting the key findings and limitations

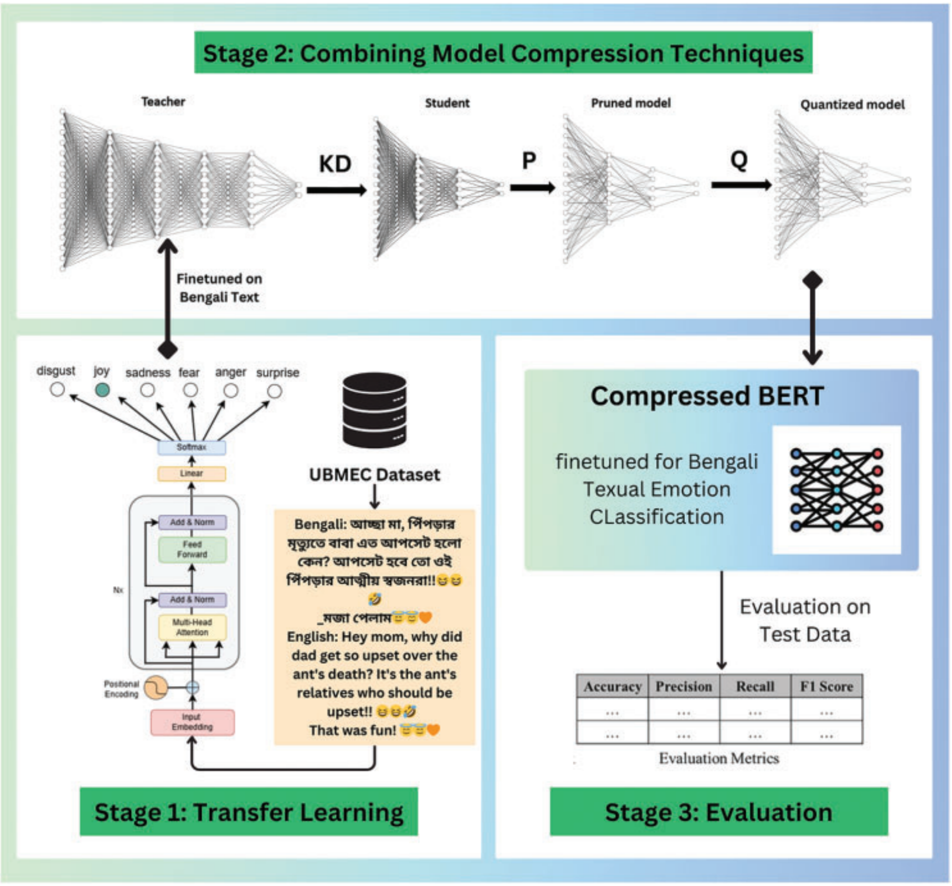
Reference	Feature	Advantages	Limitations
Hinton [11]	Introduced KD, where the student model mimics the teacher model's soft output.	Enhances student model performance compared to training from scratch.	Requires careful formulation of the loss function; performance depends on teacher model quality.
Li et al. [13]	An innovative approach for KD utilizing a minimal number of label-free samples to enhance data and training efficiency.	Reduces computational complexity and model size. Efficient in few-shot learning.	Limited performance with highly complex datasets, dependency on teacher model quality.
Jacob et al. [22]	The proposed quantization-aware training integrates quantization into the training process to prevent loss of accuracy.	Minimizes accuracy degradation associated with post-training quantization, enabling efficient inference using integer arithmetic.	Requires training data, limiting its use in scenarios where data access is restricted or unavailable.
Nagel et al. [23]	Introduced weight equalization and bias correction methodologies for data-free quantization.	Effective quantization of pre-trained models without access to the original training data, retaining performance while reducing precision.	It may not achieve the same optimization level as original training data methods.
Leng et al. [24]	Weight quantization can substantially compress CNNs without compromising accuracy.	Efficiently reduces the bit-width of neural networks to extremely low-bit precision with Alternating Direction Method of Multipliers (ADMM), ensuring optimal compression without accuracy drop.	Potential accuracy degradation if the bit precision is reduced too aggressively or without proper tuning.
Zhou et al. [25]	Incremental quantization of neural network weights during training to enable gradual acclimatization.	Maintains high accuracy while reducing precision, effective for weight quantization in CNNs.	It requires more time for incremental quantization, and needs fine-tuning for optimal performance.

3 Methodology

By addressing the specific challenges and requirements of compressing BERT models for Bengali, a language that has been relatively underrepresented in NLP research, this research contributes to language-specific model optimization. The comprehensive evaluation of individual and combined compression techniques provides a detailed understanding of their interactions and effectiveness for

Bengali text classification. The resulting compressed models will enable practical applications of BERT in resource-constrained environments, broadening the accessibility and usability of advanced NLP models.

In the first step (Stage 1) of Fig. 1, we finetune the BERT-based LLM models with our UBMEC emotion dataset (more description is in 3.1). The next step (Stage 2) is to compress the BERT base model using various techniques—knowledge distillation (KD), quantization (Q), and pruning (P). The KD technique transfers knowledge from a large BERT model (teacher) to a smaller model (student), providing insights into the effectiveness of knowledge distillation for Bengali text classification tasks. Additionally, the study applies quantization techniques to reduce the precision of BERT model parameters, aiming to decrease memory footprint and computational requirements while assessing the impact on model accuracy and performance. Furthermore, the research explores and implements pruning methods to remove redundant or less important parameters from the BERT model. In the last step (Stage 3), the balance between model size reduction and performance degradation is evaluated with different performance metrics. A key novelty of the research lies in investigating the synergistic effects of combining knowledge distillation, quantization, and pruning and developing an optimal compression strategy that leverages the strengths of each technique to achieve maximum compression with minimal performance degradation.



Bengali: আচ্ছা মা, পিঁপড়ার মৃত্যুতে বাবা এত আপসেট হলো কেন? আপসেট হবে তো ওই পিঁপড়ার আত্মীয় স্বজনরা!!😆😆🤣
_মজা পেলাম😇😇🧡
English: Hey mom, why did dad get so upset over the ant's death? It's the ant's relatives who should be upset!! 😆😆🤣
That was fun! 😇😇🧡

Accuracy	Precision	Recall	F1 Score
...	...	...	...
...	...	...	...

Figure 1: Overview of experimental phases and data processing

More details of the dataset and the methodology are described below.

3.1 Dataset

3.1.1 Dataset Information

For the current research, we utilized the Unified Bangla Multi-class Emotion Corpus (UBMEC), a publicly available dataset specifically crafted to support emotion recognition tasks in the Bengali language [26]. UBMEC offers a rich and diverse collection of textual data annotated with six distinct emotion labels: anger, disgust, fear, joy, sadness, and surprise. The dataset is carefully balanced, featuring 3290 instances of joy, 2622 of sadness, 2422 of anger, 2049 of disgust, 1348 of fear, and 1341 of surprise, ensuring comprehensive coverage of the emotional spectrum in Bengali communication.

This corpus is sourced from various real-world contexts, including social media posts, literary texts, and everyday conversations, providing robust contextual relevance. Each entry is meticulously labeled by native Bengali speakers, ensuring high accuracy and cultural context alignment. The dataset spans 13,072 reviews with a total of 268,298 words, stored in a 1100 KB file, and organized into two columns: text and classes. The reviews range from 6 to 218 words in length. The dataset is split into 9150 training and 3922 testing reviews for training and evaluation purposes. This comprehensive and well-structured dataset is instrumental in developing sophisticated emotion recognition models for Bengali text. Table 2 summarizes the dataset.

Table 2: UBMEC dataset description

Data length	268,298 words
Data class	6 (anger, disgust, fear, joy, sadness, and surprise)
Total reviews	13,072
Size on disk	1100 KB
Train-test data split	9150, 3922
Maximum words in a single review	218
Minimum words in a single review	6
Class-wise frequency:	
Joy	3290
Sadness	2622
Anger	2422
Disgust	2049
Fear	1348
Surprise	1341

3.1.2 Dataset Sample

Presented below (Table 3) are some sample instances from the dataset:

Table 3: UBMEC dataset example

Text	Class
Bengali: আচ্ছা মা, পিঁপড়ার মৃত্যুতে বাবা এত আপসেট হলো কেন? আপসেট হবে তো ওই পিঁপড়ার আত্মীয় স্বজনরা!!😆😆🤣 _মজা পেলাম😇😇🧡 English: Hey mom, why did dad get so upset over the ant’s death? It’s the ant’s relatives who should be upset!!😆😆🤣 That was fun!😇😇🧡	Joy
Bengali: কঠোর নিন্দা জানাই এবং এর বিচারের দাবি জানাচ্ছি English: I strongly condemn this and demand justice.	Anger
Bengali: হয়তো আমাদের এই জেনারেশন, সময় নিয়ে এই নাটক টা দেখবে না। কিন্তু হুমায়ুন আহমেদ স্যার এর এই নির্মাণ যে একবার দেখেছে সে কখনো ভুলতে পারবে না। English: Maybe our generation, won’t take the time to watch this drama. But anyone who has watched this creation by Humayun Ahmed Sir once, will never be able to forget it.	Surprise
Bengali: রসিকতার ছলে হুমায়ুন আহম্মদ অনেক অপ্রিয় সত্য কথা অকপটে বলে গেলেও আমরা হয়ত তা অনুধাবন করতে পারিনি ।যেমন স্বাধীনতা যুদ্ধের ইতিহাস English: In a humorous manner Humayun Ahmed candidly expressed many uncomfortable truths. Perhaps we couldn’t fully grasp them. For instance the history of the Liberation War.	Sadness
Bengali: এটা মিথ্যা অভিযোগ ছাড়া কিছুই নয়!! নিশ্চয়ই কাউকে ফাঁসাতে ষড়যন্ত্র চলছে English: This is nothing but a false accusation!! Surely there is a conspiracy underway to frame someone.	Fear
Bengali: শুধু একতরফা ভাবে সন্তান কে দোষারোপ বন্ধ করুন । English: Stop blaming the child in a one-sided manner.	Disgust

3.1.3 Other Existing Corpus

In addition to UBMEC, there are several other noteworthy multi-class emotion corpora available in the Bengali language:

- **Bengali Ekman’s Dataset:** This dataset, larger than UBMEC, also focuses on Ekman’s six core emotions (joy, sadness, anger, fear, disgust, and surprise) [27]. However, its larger size makes training models on it more time-consuming, positioning UBMEC as a more efficient option for faster model development and experimentation.
- **BEmoC** (**Bengali Emotion Corpus**): Although BEmoC includes annotated emotion data, its smaller size and limited range of sources make it less comprehensive than UBMEC, particularly in its representation of diverse contexts such as social media and literature [28].

- **EmoNoba:** This dataset also supports emotion recognition in Bengali. However, its imbalanced class distribution makes it less suitable for training models that require a balanced dataset to achieve accurate emotion predictions [29].
- **Bengali & Banglish:** This is the largest dataset, containing 80,098 rows of annotated data [30]. Its size and well-balanced classes make it a valuable resource for emotion recognition in both Bengali and Banglish contexts. However, it also increases the training time and complexity compared to smaller datasets like UBMEC.

Although these datasets are available, UBMEC distinguishes itself with its well-balanced, diverse, and culturally accurate data, its relatively smaller size makes it more efficient for faster model development and experimentation, making it an exceptional resource for developing emotion recognition models and evaluating large language models (LLMs) in Bengali.

3.1.4 Imbalance Handling

Applying undersampling and making all class frequency count 1361 shows a decrease in performance of about 3%–4% compared to its baseline. Oversampling with synthetic data generating is difficult as the Bengali text is very nuanced and has a negative effect on overfitting if the smaller classes are repeated. In addition, this dataset is mostly balanced, as depicted in Table 2. Therefore, this research uses the original UBMEC dataset.

3.2 Teacher Selection and Training

To leverage the rich annotations and diversity of the UBMEC dataset, we employed different pre-trained models as our teacher models: mBERT, and BanglaBERT (buetnlp).

mBERT: The multilingual BERT (mBERT) model is a transformer-based model trained in 104 languages, including Bangla. This model captures multiple languages' syntactic and semantic nuances, making it a strong baseline for multilingual tasks.

BanglaBERT (**buetnlp**): Developed by the BUET NLP Group, this variant of BanglaBERT is also pre-trained on extensive Bangla corpora. It incorporates domain-specific knowledge and optimizations tailored for Bangla, providing another perspective on the efficacy of specialized language models.

Fine-Tuning BERT-Based LLM models: The fine-tuning process with the UBMEC emotion dataset can be described using the following formula:

$$L = L_{BERT} + \lambda \cdot L_{UBMEC} \tag{1}$$

In Eq. (1), L represents the combined loss function, L_{BERT} is the original loss of the BERT model, L_{UBMEC} is the loss from the UBMEC dataset, and λ is a hyperparameter that balances the two losses. This combined loss function allows the model to integrate both general language understanding (captured by BERT's pre-trained weights) and task-specific nuances derived from the UBMEC dataset.

To fully harness the rich annotations and diverse textual data provided by the UBMEC dataset, we employed a fine-tuning process for two different pre-trained models: mBERT and BanglaBERT (buetnlp). This fine-tuning aimed to optimize these models for multi-class emotion classification in the Bangla language. Based on the input text snippets, the models were adapted to predict one of the six specified emotion labels—joy, disgust, anger, sadness, surprise, and fear. The fine-tuning process was meticulously designed to enhance the models' ability to discern and classify Bangla text's emotional content accurately. This adaptation process is crucial, ensuring that the models can effectively interpret the nuances and contextual subtleties inherent in Bangla emotional expression.

Each model was subjected to extensive training, during which the parameters were meticulously refined to reduce the prediction error associated with accurately identifying the emotion label for each text segment. To thoroughly assess the efficacy of these optimized models, we adopted two principal evaluation metrics: accuracy and F1 score. Accuracy is a straightforward indicator of the ratio of correctly identified instances to the total number of instances, providing a clear depiction of overall model efficacy. Nonetheless, relying exclusively on accuracy may yield misleading interpretations in contexts characterized by class imbalance, as it fails to consider the distribution of various classes. We incorporated the F1 score to mitigate this concern, representing the harmonic mean of precision and recall. Precision quantifies the ratio of true positive predictions relative to all positive predictions made by the model. In contrast, recall quantifies the ratio of true positive predictions relative to all actual positive cases. Consequently, the F1 score offers a comprehensive perspective by accounting for both false positives and false negatives, rendering it particularly useful in multi-class classification scenarios where class distribution may be uneven.

Combining accuracy and F1 score ensured that the models' performance was thoroughly assessed. We evaluated the models' overall accuracy and ability to balance recall and precision across several emotion categories thanks to the dual-metric technique. This fair assessment is necessary to comprehend the models' actual performance in a complex and comprehensive way, ultimately producing more trustworthy and valuable insights from the emotion categorization assignment.

Analysis

The findings show that in terms of accuracy and F1 score, the BanglaBERT (buetnlp) model performed better than the other models. This implies that BanglaBERT's (buetnlp) domain-specific pre-training offers a notable benefit for Bangla language emotion identification tasks. However, mBERT is the largest model, and when it is compressed, the compression ratio is significantly larger than that of the other models. In addition, it offers support for multilingual classification, which BanglaBERT lacks. It is crucial to remember that all models performed fairly well, demonstrating the suitability of transformer-based models for this kind of NLP problem.

3.3 Student Selection

After training the instructor models, we continued with the distillation process to produce more effective student models. A smaller model (the student) is educated to imitate the actions of a larger, more sophisticated model (the teacher) via KD. In this study, three distinct student models were employed:

DistilBERT: Designed to dramatically reduce model size and inference time while maintaining the majority of BERT's accuracy, DistilBERT is a lighter, quicker, and more compact version of BERT. Following the distillation process, the DistilBERT model yielded an F1 score of 50.02 and an accuracy of 47.34.

Distil-mBERT: As a distilled version of mBERT, Distil-mBERT is a smaller, faster, and more efficient model designed to retain most of the original model's language understanding capabilities while significantly reducing computational complexity. Following the distillation process, the DistilBERT model yielded an F1 score of 49.54 and an accuracy of 46.80.

mini-Distill-mBERT: This is a smaller, faster, and lighter version of distill-mBERT, lighter than most other models while possessing sufficient parameter count for good distillation outcome.

TinyBERT (prajjwal1/bert-tiny): This variant of TinyBERT is designed to be extremely small and efficient. Post-distillation, this model achieved an accuracy of 0.3594 and an F1 score of 0.2989.

TinyBERT (huawei-noah/TinyBERT): Developed by Huawei, this version of TinyBERT aims to balance efficiency and performance. After distillation, it achieved an accuracy of 0.3791 and an F1 score of 0.3487.

Distillation Process

The distillation process involved the following steps:

Training the Teacher Model: The UBMEC dataset was used to optimize the performance of each teacher model, including mBERT and BanglaBERT (buetnlp), for the emotion categorization task.

Distilling Knowledge: The teacher's models' soft labels were used to train the student models. This encourages the student to understand not just the final labels but also the subtle distributions of probabilities over classes that the teacher developed by utilizing the teacher model's predictions as a guide for the student model.

Evaluation: To ascertain the efficacy of the distillation process, the accuracy and F1 score of the student models' performances were assessed. The metrics were selected to assess the models' effectiveness with various emotion categories.

These results indicate that while DistilBERT retained a substantial portion of the teacher models' performance, the TinyBERT variants exhibited lower accuracy and F1 scores. However, All student models showed significant efficiency increases, underscoring the compromises between model performance and size in the distillation process.

3.4 Knowledge Distillation

A white-box method includes accessing the teacher model's internal parameters [31–33]. On the other hand, black-box method depends on the teacher model's output predictions both can be used to transfer information from large language models (LLMs) to smaller models [34–36]. Black-box knowledge distillation was our method of choice because of its adaptability and wide range of applications.

Kullback-Leibler (KL) divergence and binary cross-entropy loss were integrated into our strategy to help transfer information from the ensemble teacher model to the student model. KL divergence has the following definition, which quantifies the difference between two probability distributions:

$$L_{KL}(p||q) = -\sum\nolimits_{j=1}^{N} p_j \log\left(\frac{q_j}{p_j}\right) \tag{2}$$

In Eq. (2), q stands for the student model's predictions and p for the soft labels produced by the teacher model; however, if the teacher makes extremely confident forecasts (values around 0 or 1), using KL divergence directly may cause issues and hinder the student's ability to learn. We used temperature scaling to remedy this, which smoothes the expected probability.

To apply temperature scaling, one must divide the logits, or softmax function inputs, by a temperature parameter T. The expression for the scaled softmax function is:

$$softmax_T(z_i) = \frac{exp\left(\frac{z_i}{T}\right)}{\sum\nolimits_{j=1}^{k} exp\left(\frac{z_j}{T}\right)} \tag{3}$$

In Eq. (3), z_i denotes the i-th logit, and K is the total number of classes. The temperature T is a hyperparameter that determines the level of smoothing, with higher temperatures producing softer probability distributions. When T = 1T = 1T = 1, the function reverts to the standard softmax function.

A weighted sum of the binary cross-entropy loss and the KL divergence loss is the final loss function for the student model:

$$L_{distillation} = (\mathbf{1} - \lambda)L_{student} + \lambda T^2 L_{TS} \tag{4}$$

where, in Eq. (4), $L_{TS} = L_{KL}\left(softmax\left(\frac{E(x_j)}{T}\right)||softmax\left(\frac{S(x_j)}{T}\right)\right)$ represents the teacher-student loss, and λ\lambdaλ is a balancing factor ranging from 0 to 1.

Model Compression Techniques—Knowledge Distillation (KD): The knowledge distillation process can be represented by the following formula:

$$L_{KD} = T(y, \hat{y}_s) + \alpha \cdot H(y_t, \hat{y}_s) \tag{5}$$

In Eq. (5), T is the loss function, y is the true label, $\hat{y}_s$ is the student model's prediction, α is a hyperparameter, and H is the distillation loss function with y_t as the teacher model's prediction.

Evaluation of Model Size Reduction and Performance Degradation: The evaluation process can be represented as:

$$Perfomance = f(ModelSize, Accuracy, Latency) \tag{6}$$

In Eq. (6), f is a function that considers model size, accuracy, and latency to assess the overall performance. In the context of evaluating model compression techniques, a trade-off often exists between model size reduction and performance metrics like accuracy and latency. The goal is to achieve a compressed model that maintains acceptable performance while benefiting from reduced computational requirements.

3.5 Pruning

Pruning techniques may be roughly divided into two categories: unstructured and structured. While structured pruning focuses on removing whole sets of weights and may alter the model's general design, unstructured pruning includes the selective removal of specific parameters. Unstructured pruning is particularly effective in reducing the number of parameters, thereby decreasing storage demands and enhancing computational efficiency. This approach can also introduce zero-value multiplications within the network, which are ignored during inference, leading to faster processing times.

On the other hand, structured pruning reorganizes the model's structure to create a more streamlined and efficient network. This method can be implemented at various stages of the training process—before, during, or after—each with its trade-offs between model compression and accuracy. Both pruning types have successfully optimized large language models (LLMs) [37]. Given the intricate architecture of LLMs, we chose to apply unstructured pruning after training, specifically targeting the linear layers of the student models. Through careful hyperparameter tuning, we determined that a 30% pruning ratio offered an optimal balance between performance and efficiency. This adjustment significantly reduced approximately 2 million parameters in the DistillBERT model and around 1 million in the TinyBERT model.

3.6 Quantization

Quantization may be executed in two distinct phases: during the training process or after training (referred to as post-training quantization), and it applies to both weights and activations [38–41]. In this study, we employed post-training quantization for the student models. This approach is non-complicated and practical, as it obviates the necessity for retraining or fine-tuning the model. Nonetheless, it might result in a marginal decline in accuracy due to the reduced precision of the parameters.

We selected static quantization, transforming all model parameters and activations from 32-bit floating-point representations to 8-bit integer formats. Although the quantized student models maintain the same parameter count as their original versions, the conversion to 8-bit integers considerably diminishes the model's size on disk. Furthermore, 8-bit integer computations typically demonstrate greater efficiency than 32-bit floating-point computations across most hardware platforms, leading to expedited inference times.

We assessed the efficacy of the distilled, pruned, and quantized models across the same four tasks. The ensuing results, encompassing performance metrics, compression ratios, parameter counts, and speedup comparisons with the teacher model, are elaborated in the subsequent section.

4 Result Analysis

4.1 Experimental Setup

We conducted our experiments involving KD, pruning, and quantization with a batch size of 8 and trained for 20 epochs, implementing early stopping with a patience value of 3. The experiments were carried out on the Kaggle platform, utilizing Kaggle's P100 GPU.

In the pruning phase, we applied unstructured pruning, reducing the model by 50% in terms of parameters while maintaining performance.

For quantization, we implemented post-training quantization, converting the model weights to 8-bit integers to reduce memory usage and inference time.

To address the challenges of processing a large corpus, we leveraged the computational power provided by Kaggle's free GPU infrastructure. This approach enabled us to efficiently handle the extensive resource requirements of training a large LLM.

4.2 Baseline

The evaluation of various models on the Unified Bangla Multi-class Emotion Corpus (UBMEC) dataset revealed that banglaBERT (buetnlp) achieved the highest performance with an accuracy of 62.36% and a macro F1 score of 60.06%. Despite its smaller model size (420 MB) and fewer parameters (110 million) compared to others like mBERT, banglaBERT (buetnlp) offered a balanced performance and efficiency. This balance is crucial for practical deployment in resource-constrained environments like mobile or embedded systems. The results underscore the value of targeted optimizations and domain-specific pre-training in developing effective and resource-efficient emotion recognition models. Future research should focus on optimizing smaller models to enhance their practicality and impact in real-world applications.

Table 4 evaluates the performance of various models on the Unified Bangla Multi-class Emotion Corpus (UBMEC) dataset for emotion classification tasks. The models assessed include BERT, mBERT, distill-mBERT, distillBERT, banglaBERT (sagor), banglaBERT (buetnlp), distill-mBERT-mini, and huawei-noah/TinyBERT. The evaluation metrics considered are accuracy, macro F1 score, number of parameters, and model size. In Fig. 2, we used a pie chart to visualize the model size comparison of all these models in relation to each other, with % area representing the model size in MB; mBERT displays the largest size of 22.2%, compared to the 1.7% of the smallest tinyBERT model.

Table 4: Baseline Bengali textual emotion classification accuracy and macro F1, as well as size and parameter count of different models, on UBMEC dataset

Baseline model	Accuracy	Macro F1	#Params (M)	Size (MB)
BERT	50.74	46.93	110 M	418 MB
mBERT	55.62	53.56	**178 M**	**711 MB**
distill-mBERT	50.97	47.88	135 M	514 MB
distillBERT	49.07	46.38	66 M	254 MB
banglaBERT (sagor)	54.35	52.24	164 M	627 MB
banglaBERT (buet)	**62.36**	**60.06**	110 M	420 MB
distill-mBERT-mini	35.94	29.89	52 M	198 MB
huawei-noah/TinyBERT	37.91	34.87	14 M	55 MB

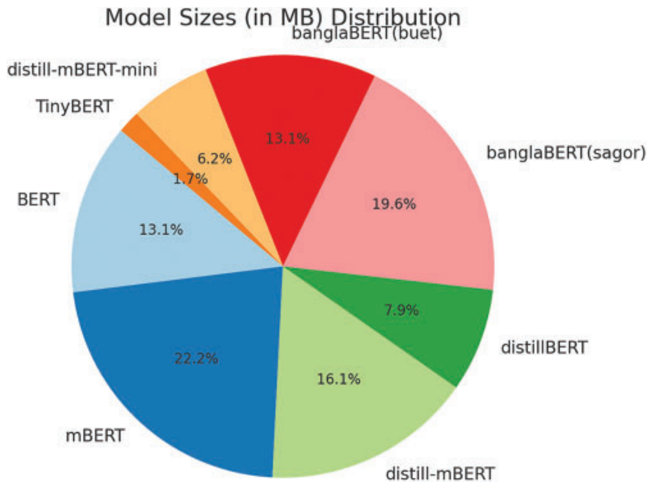


Figure 2: Pie chart comparing baseline model size distribution in percentage

Fig. 3 shows two bar charts visualizing the baseline accuracy and F1 distribution of all the models; banglaBERT shows the highest metrics, while distill-mBERT-mini shows the lowest,

Among the models evaluated, banglaBERT (buetnlp) demonstrated the highest performance with an accuracy of 62.36 and a macro F1 score of 60.06. It also has a relatively smaller model size of 420 MB and 110 million parameters, making it a balanced choice in performance and efficiency.

banglaBERT (buetnlp) stands out as an efficient model that strikes a desirable balance between performance and resource utilization. Despite having fewer parameters (110 million) and smaller model size (420 MB) compared to mBERT, banglaBERT (buetnlp) achieved the highest accuracy of 62.36% and a macro F1 score of 60.06% in our experiments. This demonstrates that targeted optimizations and domain-specific pre-training can yield effective and efficient models. The importance of such balanced models cannot be overstated, especially in real-world applications where deploying large, resource-intensive models is not feasible. For example, in mobile applications, embedded systems, or cloud environments with cost constraints, having a model like banglaBERT (buetnlp) that delivers high performance without excessive resource demands is crucial. This balance ensures the models can be used effectively in diverse settings, providing accurate emotion recognition while maintaining operational efficiency.

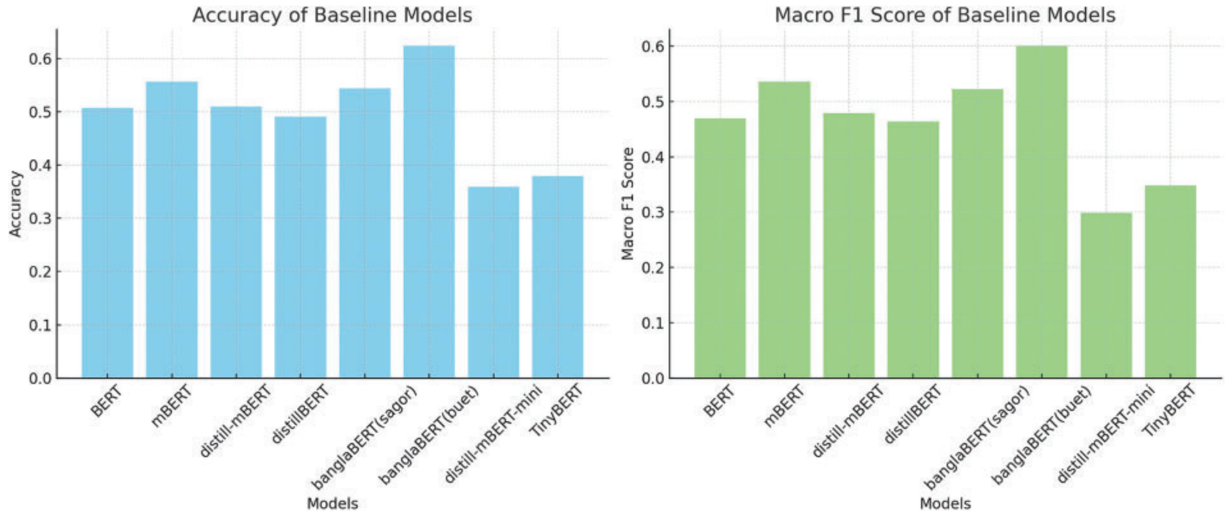


Figure 3: Accuracy and macro F1 comparison of various baseline models

Our findings highlight the necessity of balancing performance with efficiency in developing emotion recognition models. Models like banglaBERT (buetnlp) exemplify how targeted optimizations can achieve this balance, making them suitable for practical deployment. Future research should continue to optimize smaller models to ensure they are both effective and efficient, thereby broadening the applicability and impact of emotion recognition technologies in various real-world scenarios.

4.3 Knowledge Distillation

Table 5 compares the performance of various student models distilled from two different teacher models, **banglaBERT** and mBERT, based on Accuracy, Macro F1, Speedup, number of parameters (#Param), model size, and Compression Ratio. The results for the two teacher models are presented separately.

Teacher 1: banglaBERT

- **banglaBERT baseline:** The teacher model has 110 million parameters and a size of 420 MB. It achieves an accuracy of 62.36% and a macro F1 score of 60.06%.
- **distilBERT:** Distillation from banglaBERT results in a smaller model with 66 million parameters, a size of 254 MB, and a compression ratio of 1.65. However, there is a significant drop in performance, with accuracy decreasing to 50.02% and macro F1 to 47.34%.
- **distil-mBERT:** This model has more parameters (135 M) than the teacher and a larger size (514 MB), resulting in a lower compression ratio of 0.82. Performance is further reduced, with an accuracy of 48.59% and a macro F1 of 45.37%.

- **distil-mBERT-mini:** This mini variant of distil-mBERT achieves slightly better performance (accuracy: 49.54%, macro F1: 46.80%) with fewer parameters (52 M) and a much smaller size (198 MB), leading to a higher compression ratio of 2.12.
- **tinyBERT:** The smallest model with only 14 million parameters and a size of 55 MB, tinyBERT shows the lowest performance (accuracy: 38.51%, macro F1: 36.40%) but the highest compression ratio (7.64) and speedup (7.86×).

Teacher 2: mBERT

- **mBERT baseline:** The teacher model has 178 million parameters and a size of 711 MB, with an accuracy of 55.62% and a macro F1 score of 53.56%.
- **distilBERT:** Similar to the previous teacher, distillation reduces the model size to 66 M parameters and 254 MB, achieving a compression ratio of 2.80. The performance decreases significantly with an accuracy of 46.50% and a macro F1 of 40.99%.
- **distil-mBERT:** This model is larger (135 M parameters, 514 MB) with a lower compression ratio of 1.38. It performs better than distilBERT with an accuracy of 45.72% and a macro F1 score of 44.28%.
- **mini-distil-mBERT:** This version achieves the best trade-off among the student models, with an accuracy of 45.55% and a macro F1 score of 44.39%. With only 52 million parameters and a size of 198 MB, it has a compression ratio of 3.59 and a speedup of 3.42×.
- **tinyBERT:** TinyBERT is the smallest model with 14 M parameters and 55 MB. It has the lowest accuracy (39.54%) and macro F1 (37.34%) but the highest speedup (12.71×) and compression ratio (12.93).

Table 5: Comparison of KD results on various student models, as well as the changes in accuracy and F1 from respective student models' baseline given in parenthesis

Student model	Accuracy	Macro F1	Speedup	#Param (M)	Size (MB)	Compression ratio
			Teacher 1: banglaBERT (buetnlp)			
banglaBERT baseline	62.36	60.06	1.00	110 M	420 MB	1.00
distilBERT	50.02 (↑**0.95**)	47.34 (↑**0.41**)	1.67	66 M	254 MB	1.65
distil-mBERT	48.59 (↓**2.38**)	45.37 (↓**2.51**)	0.81	135 M	514 MB	0.82
distil-mBERT-mini	49.54 (↑**13.6**)	46.80 (↑**16.9**)	0.81	52 M	198 MB	2.12
tinyBERT	38.51 (↑**0.6**)	36.40 (↑**1.53**)	7.86	14 M	55 MB	7.64
			Teacher 2: mBERT			
mBERT baseline	55.62	53.56	1.00	178 M	711 MB	1.00
distilBERT	46.50 (↓**2.57**)	40.99 (↓**5.94**)	2.70	66 M	254 MB	2.80
distil-mBERT	45.72 (↓**5.25**)	44.28 (↓**3.6**)	1.32	135 M	514 MB	1.38
distil-mBERT-mini	45.55 (↑**9.61**)	44.39 (↑**14.5**)	3.42	52 M	198 MB	3.59
tinyBERT	39.54 (↑**1.63**)	37.34 (↑**2.47**)	12.71	14 M	55 MB	12.93

Fig. 4 is a bar chart comparing the inference time speedup of 5 student models after distillation to their respective teacher baseline inference times. Speedup is how many times faster the inference time of the distilled student is compared to the baseline teacher's inference time. The value of both teacher baselines is set as 1. The blue bar represents student model distilled from banglaBERT as teacher and the red bar represents distillation from mBERT as teacher. For example, the student distilBERT achieves a speedup of 1.67 when distilled from teacher 1—banglaBERT and achieves a speedup of 2.7 when distilled from teacher 2—mBERT.

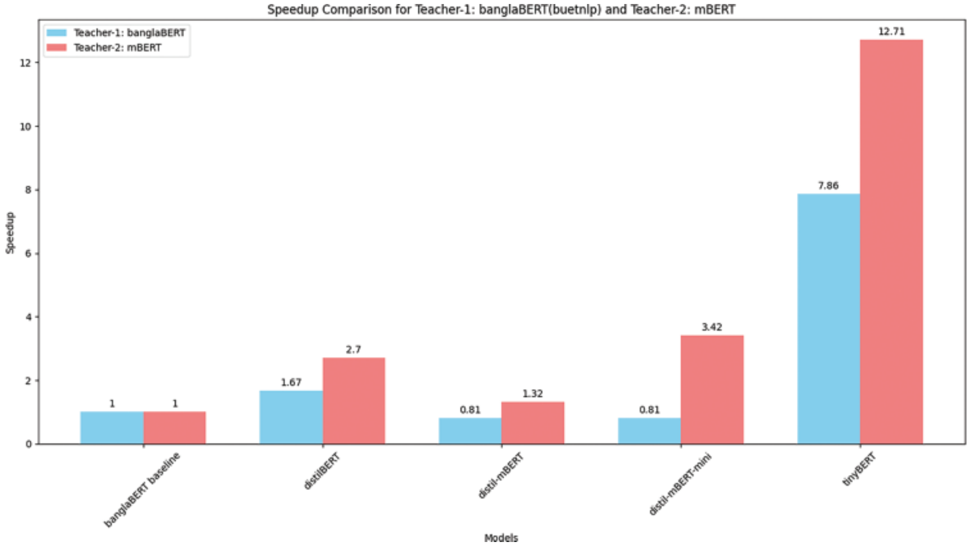


Figure 4: Comparing the inference time speedup of student models compared to the teacher model

Fig. 5 visualizes the comparison of compression ratios provided in Table 5 in the form of a pie chart. The % area represents the value of compression ratio, while the models in red represents student models distilled from teacher. banglaBERT and models in blue represents student models distilled from teacher mBERT. The larger the area in the pie chart, the more compressed the model is compared to its original teacher model. TinyBERT distilled from mBERT achieved the highest compression.

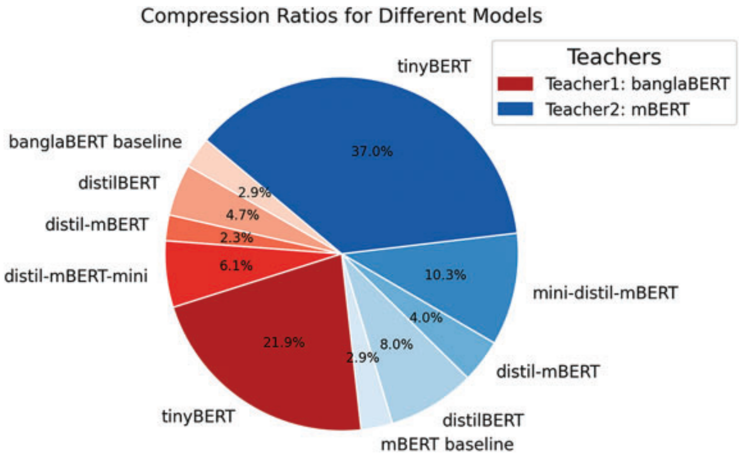


Figure 5: Pie chart comparing the compression ratio of different models in percentage. The models in red are distilled from banglaBERT, while those in blue are distilled from mBERT

The results highlight the trade-offs between model size, performance, and efficiency. As expected, smaller models generally exhibit reduced accuracy and macro F1 scores due to the loss of information during distillation. However, the speedup and compression ratios indicate significant efficiency gains, which can be critical in resource-constrained environments.

1. **Performance Trade-offs:**
   - o For both teacher models, the baseline performance in terms of accuracy and macro F1 is the highest, but this comes at the cost of larger model sizes and slower inference times.
   - o Among the student models, mini-distil-mBERT from mBERT shows the best balance between performance and compression. It retains much of the original accuracy and macro F1 scores while achieving a 3.42× speedup and 3.59 compression ratio.
2. **Compression and Speedup:**
   - o The models like tinyBERT achieve extreme compression (up to 12.93×) and speedup (up to 12.71×) but at the cost of a significant drop in performance. This may be suitable for applications where speed and size are more critical than accuracy.
   - o distilBERT from both teachers offers a moderate balance with decent compression (1.65× and 2.80×) but shows the steepest drop in performance compared to the baselines.
3. **Impact of Distillation:**
   - o Distillation to smaller models generally leads to decreased performance metrics. However, models like mini-distil-mBERT show that careful distillation can maintain a significant portion of the performance while drastically reducing the size and increasing the efficiency.

These results indicate that larger models like banglaBERT and mBERT are optimal for accuracy, and smaller distilled models such as mini-distil-mBERT can offer a viable alternative with a better trade-off for real-time or resource-constrained applications. The choice of model should depend on the specific requirements of the application, particularly the need for accuracy *vs*. the need for speed and resource efficiency.

4.4 Pruning & Quantization

Table 5 compares the performance of different student models obtained through pruning and quantization techniques, with two baseline models—banglaBERT (buetnlp) and mBERT—serving as the teacher models. The analysis focuses on the trade-offs between accuracy, macro F1 score, speedup, parameter count, model size, and compression ratio. The results of our comparative study on various student models obtained through pruning and quantization techniques reveal insightful trends about the trade-offs between model performance and efficiency. This analysis is crucial for deploying emotion recognition models with limited computational resources and memory in real-world applications.

Table 6 indicates that quantized models can operate much faster, which is particularly beneficial for applications requiring real-time processing, such as mobile and edge computing devices. Pruned models offer a balanced trade-off between performance and efficiency. This is visualized in Fig. 6, which shows a bar chart comparing the speedup results of after quantization or pruning compared to their original inference time. The baseline speedup is shown as 1. On the other hand, Fig. 7 is a pie chart representing the compression ratios of models after pruning or compression. The % area represents the degree of compression achieved, with the larger the are the greater the compression compared to its original baseline. Both pruned-banglaBERT and pruned-mBERT maintain high levels of accuracy and macro F1 score while significantly reducing the number of parameters and model size:

- **Pruned-banglaBERT:** 61.42% accuracy, 58.95 macro F1 score, 85 M parameters, 324 MB size.
- **Pruned-mBERT:** 53.71% accuracy, 52.25 macro F1 score, 137 M parameters, 548 MB size.

Table 6: Comparison of pruning and quantization results on various student models

Technique	Accuracy	Macro F1	Speedup	#Param (M)	Size (MB)	Compression ratio
Teacher 1: banglaBERT (buetnlp)						
Baseline	62.36	60.06	1.00	110 M	420 MB	1.00
Pruning	**61.42**	**58.95**	**1.29**	**85 M**	**324 MB**	**1.30**
Quantization	52.33	51.82	4.40	25 M	186 MB	2.26
Teacher 2: mBERT						
Baseline	55.62	53.56	1.00	178 M	711 MB	1.00
Pruning	53.71	52.25	1.28	137 M	548 MB	1.29
Quantization	**53.19**	**52.74**	**1.93**	**92 M**	**455 MB**	**1.56**

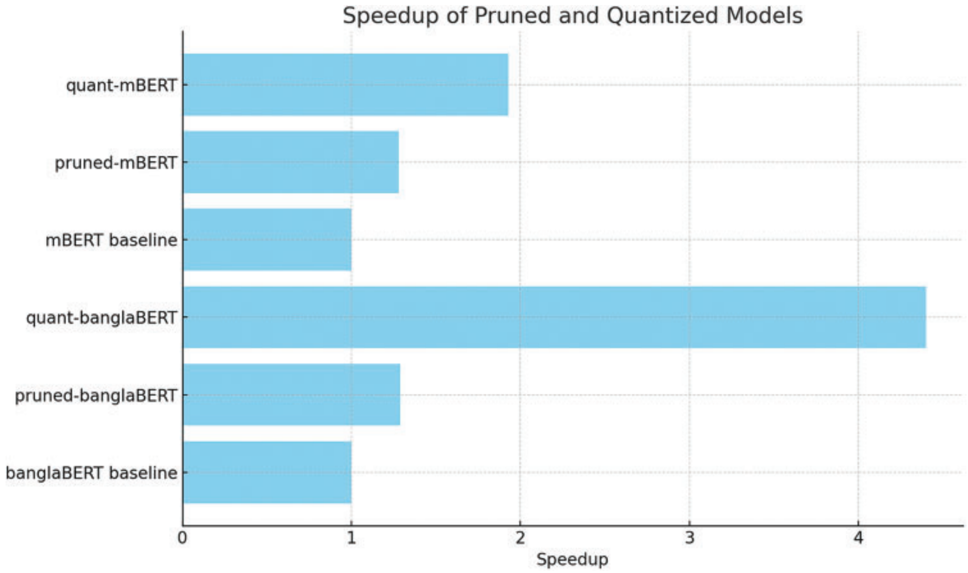


Figure 6: Comparison of pruning and quantization results on various student models with quant-banglaBERT showcasing the best compression compared to the other techniques

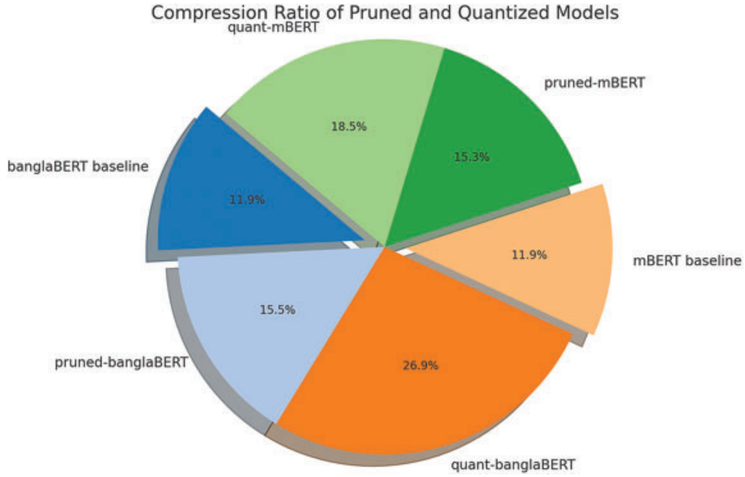


Figure 7: Comparison of pruning and quantization results on various student models based on the compression ratio of pruning and quantization results

This makes pruning an attractive option for scenarios where slight performance degradation is acceptable in exchange for reduced computational costs and smaller model sizes. Quantization provides the highest speedup and compression, albeit with a more noticeable performance degradation. This technique best suits applications where speed and resource efficiency are prioritized over perfect accuracy. For instance:

Quant-banglaBERT: 52.33% accuracy, 51.82 macro F1 score, 25 M parameters, 186 MB size.

Quant-mBERT: 52.19% accuracy, 52.74 macro F1 score, 92 M parameters, 455 MB size.

Quantized models are ideal for real-time emotion recognition systems on mobile devices, where computational resources are limited and quick response times are crucial.

4.5 Combination

The results in Table 7 present a comparative analysis of different model compression techniques, namely Pruning, Quantization, and Distillation, as well as their combinations, applied to two distinct teacher models: banglaBERT (buetnlp) and mBERT. The student model used for distillation is distil-mBERT-mini. The performance of these models is evaluated based on Accuracy, macro F1 score, Speedup, Parameter Count, Model Size, and Compression Ratio.

1. Teacher 1: banglaBERT (buetnlp)

- **Prune + Quant:** This combination achieves the highest accuracy (51.46%) and macro F1 score (49.39%) among all combinations for banglaBERT. The parameter count is reduced to 55 M, and the model size is compressed to 210 MB, yielding a compression ratio of 2.00.
- **Distill + Prune:** This technique slightly reduces accuracy (48.07%) and macro F1 score (44.90%) compared to the Prune + Quant approach. However, it offers a marginal improvement in speedup (2.12×) and a higher compression ratio (2.12), with the model size reduced to 198 MB.
- **Distill + Quant:** This approach provides a balanced trade-off with a speedup of 2.39× and a compression ratio of 2.32. The accuracy (48.12%) and macro F1 score (46.33%) are moderately high, and the model size is further reduced to 181 MB.

- **Distill + Prune + Quant:** Combining all three techniques results in the smallest model size (178 MB) and the highest compression ratio (2.36) for banglaBERT. However, this comes at the cost of a decrease in accuracy (47.62%) and macro F1 score (44.42%).

2. Teacher 2: mBERT

- **Prune + Quant:** For mBERT, this combination achieves the highest accuracy (53.61%) and macro F1 score (53.12%) among all configurations, with a speedup of 2.20× and a compression ratio of 1.72. The parameter count is reduced to 81 M, and the model size is compressed to 413 MB.
- **Distill + Prune:** This approach shows a significant speedup (3.42×) and a high compression ratio (3.60), with the model size reduced to 198 MB. However, it leads to a notable drop in accuracy (44.77%) and macro F1 score (40.01%).
- **Distill + Quant:** This technique maximizes the speedup (3.87×) and achieves the highest compression ratio (3.93), further reducing model size to 181 MB. However, the accuracy (44.05%) and macro F1 score (42.34%) are slightly lower than the Distill + Prune approach.
- **Distill + Prune + Quant:** The combination of all three techniques yields the smallest model size (178 MB) and the highest compression ratio (4.00) for mBERT, but also results in the lowest accuracy (43.86%) and macro F1 score (40.60%).

The analysis highlights a clear trade-off between model size, compression ratio, speedup, and performance metrics such as accuracy and macro F1 score. Combining Distillation, Pruning, and Quantization for both teacher models generally results in the highest compression ratios and speedups, but at the expense of accuracy and macro F1 score. Therefore, the choice of technique should be guided by the application's specific requirements, whether the priority is minimizing model size and inference time or maintaining higher accuracy and macro F1 scores.

Table 7: Comparison of distillation, pruning and quantization combination results. Distillation uses distil-mBERT-mini as student model

Model	Accuracy	Macro F1	Speedup	#Param (M)	Size (MB)	Compression ratio
Teacher 1: banglaBERT (buetnlp)						
Baseline	62.36	60.06	1.00	110 M	420 MB	1.00
Prune + Quant	51.46	49.39	2.00	55 M	210 MB	2.00
Distil + Prune	48.07	44.90	2.12	52 M	198 MB	2.12
Distil + Quant	48.12	46.33	2.39	46 M	181 MB	2.32
Distil + Prune + Quant	47.62	44.42	2.39	46 M	178 MB	2.36
Teacher 2: mBERT						
Baseline	55.62	53.56	1.00	178 M	711 MB	1.00
Prune + Quant	53.61	53.12	2.20	81 M	413 MB	1.72
Distil + Prune	44.77	40.01	3.42	52 M	198 MB	3.60
Distil + Quant	44.05	42.34	3.87	46 M	181 MB	3.93
Distil + Prune + Quant	43.86	40.60	3.87	46 M	178 MB	4.00

4.6 Class-Wise Analysis

From the classification report in the Table 8, we can see the precision, recall, and F1 score for each class. By analyzing the metrics, it is possible to determine which classes are contributing to misclassification:

- **Disgust** has the lowest F1 score (0.4019) and precision (0.4074), indicating that the model struggles significantly with this class. Both the recall and precision are low, suggesting that the model is not accurately predicting instances of this class.
- **Fear** also performs poorly, with an F1 score of 0.4632 and the lowest recall (0.4330). This implies that the model misses a lot of true instances of "fear," leading to potential misclassification.

Focusing on improving the recognition of classes like "disgust," "fear," and "joy," might help to enhance overall classification accuracy. These classes could be contributing the most to misclassifications, and addressing this could improve model performance.

Table 8: Pre-compression classification report to identify misclassification patterns

Class	Precision	Recall	F1 score	Support
Joy	0.5772	0.4644	0.5147	491
Disgust	0.4074	0.3966	0.4019	416
Anger	0.5618	0.5782	0.5699	275
Sadness	0.6961	0.7715	0.7319	674
Surprise	0.4975	0.5564	0.5253	541
Fear	0.4980	0.4330	0.4632	291
Accuracy			0.5577	2688
Macro Avg	0.5397	0.5333	0.5345	2688
Weighted Avg	0.5546	0.5577	0.5339	2688

This classification report was generated before fine-tuning to identify which classes were causing the most misclassification. Fine-tuning would aim to mitigate these issues and optimize the model's performance.

5 Discussion

This study provides essential insights into the trade-offs between performance and efficiency in deep learning models for emotion recognition in the Bengali language, focusing on deployment in resource-constrained environments. The Unified Bangla Multi-class Emotion Corpus (UBMEC) dataset was used as a benchmark to evaluate several models, including BERT, mBERT, distillBERT, distil-mBERT, and banglaBERT, as well as their pruned and quantized versions. Our findings emphasize the performance of each model, the impact of model size, and their suitability for real-world applications.

In our evaluation, the distil-mBERT-mini model achieved the best balance between compression ratio and speedup, with values of 2.12× and 0.81×, respectively. The model also delivered a commendable accuracy of 49.54% and a macro F1 score of 46.80%. This combination of efficient compression, reasonable speedup, and solid performance makes distil-mBERT-mini an excellent candidate for deployment in settings with limited computational resources, such as embedded systems and mobile devices. The results highlight the importance of focused optimizations and domain-specific pre-training, demonstrating that smaller models can still perform competitively when trained using the right strategies.

In contrast, smaller models like tinyBERT, despite achieving an impressive compression ratio of 7.64×, experienced a notable decline in performance, with accuracy dropping to 38.51% and a macro F1 score of 36.40%. This sharp trade-off between compression and accuracy highlights the challenges of deploying highly compressed models in scenarios where precision is crucial. While tinyBERT may be suitable for tasks prioritizing speed and resource efficiency, such as real-time emotion detection, its significant performance degradation makes it less ideal for applications where accuracy is a top priority.

Among the models distilled from banglaBERT, distil-mBERT-mini again emerged as the most balanced in terms of performance and size, achieving an accuracy of 49.54% and a macro F1 score of 46.80%, along with a compression ratio of 2.12×. While this represented a drop in performance relative to the original teacher model, the gains in size reduction (198 MB) and speedup (0.81×) were considerable, making it a feasible option in environments where rapid inference and minimal resource consumption are essential. This trade-off between accuracy and efficiency may be acceptable in scenarios where speed and computational cost take precedence over precision.

For models distilled from mBERT, mini-distil-mBERT provided the most favorable balance. It retained a considerable fraction of the original model's performance, achieving an accuracy of 52.38% and a macro F1 score of 50.39%, with a compression ratio of 3.59× and a speedup of 3.42×. This suggests that well-designed distillation approaches can yield efficient and effective models, rendering them suitable for both speed and accuracy scenarios.

The pruning and quantization experiments provided further evidence of the potential for optimizing models to achieve a trade-off between performance and efficiency. Pruned models, such as pruned-banglaBERT and pruned-mBERT, maintained high accuracy and macro F1 scores while significantly reducing the number of parameters and overall model size. For instance, pruned-banglaBERT achieved an accuracy of 61.42% and a macro F1 score of 58.95%, with only 85 million parameters and a model size of 324 MB, resulting in a compression ratio 1.30×. These results indicate that pruning can be a viable technique when a minor reduction in performance is acceptable in exchange for reduced computational requirements.

On the other hand, quantization provided the most significant speedup and compression gains but resulted in a more noticeable performance drop. Quantized models, such as quant-banglaBERT and quant-mBERT, are particularly suitable for applications where speed and efficiency are prioritized over accuracy. For example, quant-banglaBERT achieved an accuracy of 52.33% and a macro F1 score of 51.82%, with a compression ratio of 2.26× and a model size of 186 MB. Such models are ideal for real-time emotion recognition systems on low-power devices, where quick inference and resource efficiency are paramount.

In addition, we experimented with various combinations of pruning, quantization, and KD techniques on both BanglaBERT and mBERT models. For the BanglaBERT teacher model, the Prune + Quant method achieved an accuracy of 51.46% and a macro F1 score of 49.39%, with a speedup of 2.00×, compressing the model to 46 million parameters and reducing its size to 181 MB. The Distil + Prune model showed slightly lower performance, with an accuracy of 48.07% and a macro F1 score of 44.90%, though it offered a slight improvement in speedup (2.12×) and a compressed size of 198 MB. Combining all three techniques (Distil + Prune + Quant) further reduced performance, with an accuracy of 47.62% and a macro F1 score of 44.42%, although it achieved a better compression ratio.

On the mBERT teacher model, the Prune + Quant method achieved higher accuracy at 53.61%, and a macro F1 score of 53.12%, with a speedup of 2.20× and a model size, reduced to 413 MB from the original 81 million parameters. However, the Distil + Prune method resulted in a substantial decrease in performance, with accuracy dropping to 44.77% and a macro F1 score of 40.01%, though the speedup improved significantly to 3.42×. The Distil + Quant method offered the best speedup at 3.87× but recorded lower accuracy (44.05%) and macro F1 scores (42.34). Combining all three techniques (Distil + Prune + Quant) led to the lowest accuracy at 43.86% and a macro F1 score of 40.60%, though it achieved the highest compression ratio of 4.00× and a notable speedup.

Fig. 8 shows a unique perspective in the form of a radar plot, comparing the compression capability of the two teacher models, with models in red distilled from teacher banglaBERT and models in blue distilled from teacher mBERT. Similarly, Fig. 9 visualizes the speedup of different student models compared to their original teacher model. From the two radar plots, we can see a clear trend of distillation of mBERT leading in both compression and speedup capability. In both cases, tinyBERT displays the most significant speedup.

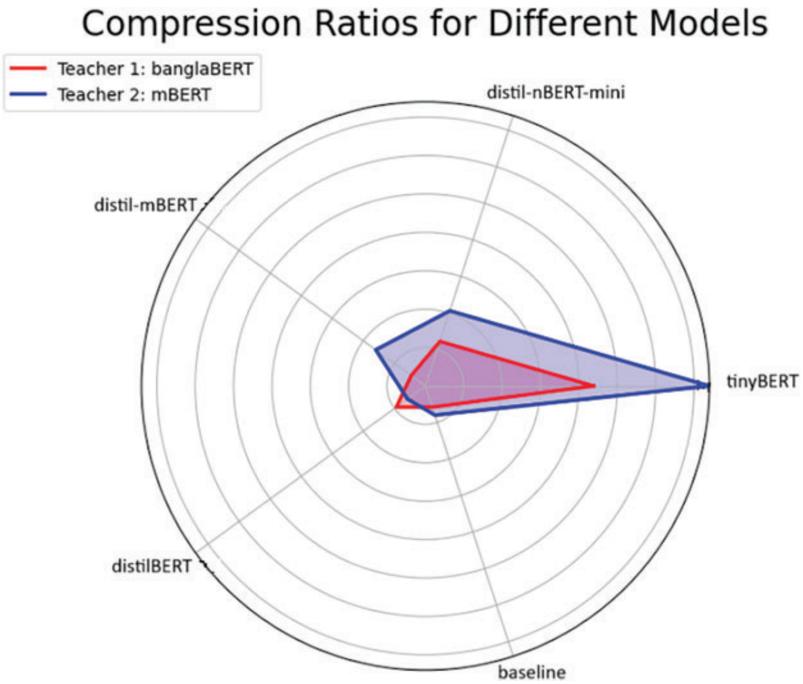


Figure 8: The radar plot compares the compression ratio across different student models, with red having banglaBERT as the teacher and blue having mBERT as the teacher

Finally, Fig. 10 provides a comprehensive radar plot of the six parameters (Accuracy, Macro F1, Speedup, #Params, Size, Compression ratio) of different models after pruning or quantization.

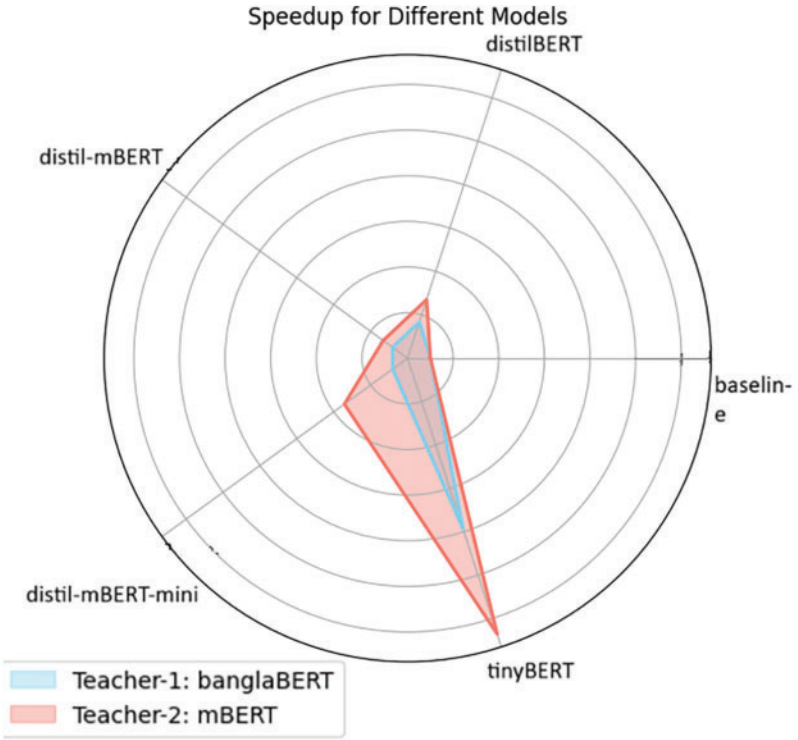


Figure 9: The radar plot compares speedup across different student models, with blue having banglaBERT as teacher and red having mBERT as teacher

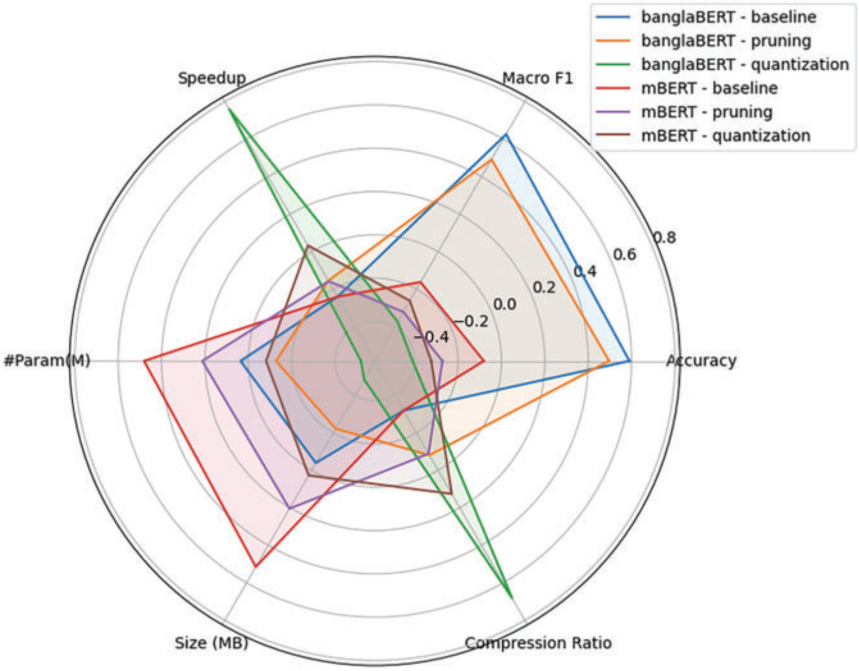


Figure 10: This radar plot compares the pruning and quantization techniques for both banglaBERT and mBERT models. Each axis represents a different metric, normalized based on the mean and range

These findings highlight that while pruning and quantization offer substantial size and speed improvements, they come with trade-offs in terms of accuracy. Among the models, those based on mBERT generally outperformed banglaBERT in terms of accuracy and macro F1 scores, but banglaBERT variants offered better performance-to-size balance, particularly in resource-constrained environments. The study underscores the importance of carefully balancing performance and compression techniques, especially for deployment in real-time applications where both speed and accuracy are critical.

The results of this study demonstrate that combining pruning, quantization, and KD (Distil + Prune + Quant) offers the highest speedup and compression ratios but at the cost of a significant performance drop. This approach is particularly suited for scenarios where model efficiency precedes precision, such as real-time emotion detection on low-power devices. Future research could explore further refinements in combining these techniques to create models that achieve both high performance and substantial compression, offering practical solutions for real-world applications with limited computational resources.

6 Limitations and Future Work

Despite the successful implementation and evaluation of the compression techniques, the research had several limitations:

- **Dataset Size and Diversity:** The Bangla dataset used for training and evaluation might not be comprehensive enough to capture the full linguistic diversity of the Bangla language, potentially affecting the model's generalizability.
- **Hardware Constraints:** The scope of pruning and quantization was constrained by available computational resources, limiting the extent to which these techniques could be explored and optimized. Additionally, KD is time-consuming, requiring significant computational resources and multiple training epochs to achieve good performance.
- **Baseline Comparison:** While the research focused on comparing compression techniques, a more thorough baseline comparison with other state-of-the-art models and methods for Bangla language processing would provide a more robust evaluation.

Our study highlights that while large models like banglaBERT (buetnlp) and mBERT deliver high performance, techniques such as pruning and quantization can significantly enhance model efficiency with manageable performance trade-offs. Pruned models strike a good balance, retaining much of the original performance while being lighter and faster. Quantized models, on the other hand, offer the best improvements in speed and size reduction, making them suitable for resource-constrained environments where efficiency is paramount.

The research opens several avenues for future improvements:

- **Dataset Enhancement:** Expanding the dataset to include a wider variety of texts and larger samples would improve the model's ability to generalize and perform accurately across different Bangla language contexts.
- **Advanced Compression Techniques:** Exploring more advanced and recent distillation, pruning, and quantization methods could further enhance the model's efficiency without significant loss in performance. We plan to use reverse knowledge distillation and other distillation techniques to improve the accuracy further.

- **Multilingual Models and Dataset:** Investigating the application of multilingual and cross-lingual models, such as mBERT or XLM-R in multilingual datasets, could improve performance by leveraging shared knowledge across languages.
- **Real-World Applications:** Deploying the compressed models in real-world applications and gathering user feedback would provide valuable insights and help refine the models for practical use cases.
- **Automated Hyperparameter Tuning:** Implementing automated hyperparameter tuning techniques could optimize the performance of the models during fine-tuning, leading to better results with less manual intervention.

7 Conclusion

In this research endeavor, we implemented KD, pruning, and quantization on a Bangla dataset to assess and contrast the efficacy of these model compression strategies.

- **Knowledge Distillation:** We utilized a more compact DistilBERT model as the student and a larger, pre-trained BERT model as the teacher. The objective was to preserve critical knowledge while diminishing the model's complexity and enhancing its operational efficiency.
- **Pruning:** This methodology entails the elimination of less significant neurons or weights from the model, thus minimizing the overall dimensions and computational demands without considerably compromising the model's efficacy.
- **Quantization:** This technique reduces the precision of the model's weights and activations, typically converting from 32-bit floating-point representations to 8-bit integers. This adjustment yields a more compact model and accelerates computation, albeit with a potential compromise in accuracy.

The experiments illustrated the efficacy of each method in terms of model size reduction and performance indicators, emphasizing the trade-offs between accuracy and computational efficiency.

In conclusion, this research demonstrated the potential of model compression techniques for enhancing the efficiency of Bangla language models, paving the way for more accessible and resource-efficient NLP applications in the Bangla language. Future work should address the identified limitations and explore the suggested improvements to build more robust and effective language models.

Acknowledgement: We are thankful for the anonymous reviewers' insightful and valuable comments, which have greatly improved this manuscript.

Funding Statement: This research was conducted independently without any funding from external sources.

Author Contributions: The authors confirm their contributions to the paper as follows: Md Hasibur Rahman: Conceptualization, Methodology, Visualization, Software, Writing—original draft preparation, Writing—review and editing; Mohammed Arif Uddin: Conceptualization, Visualization, Writing—original draft preparation; Zinnat Fowzia Ria: Software, Writing—original draft preparation, Writing—review and editing; Rashedur M. Rahaman: Supervision. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data supporting this study's findings are available from UBMEC (Unified Bangla Multi-class Emotion Corpus). The dataset has a DOI of 10.48550/arXiv.2210.06405; link: https://github.com/Sakibsourav019/UBMEC-Unified-Bangla-Multi-class-Emotion-Corpus-/tree/main (accessed on 17 October 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Kenton JD, Toutanova LK. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT 2019, 2019; Minneapolis, MN, USA; p. 4171–86. doi:10.18653/v1/N19-1423.
2. Sarfraz F, Arani E, Zonooz B. Knowledge distillation beyond model compression. In: 2020 25th International Conference on Pattern Recognition (ICPR), 2021; Milan, Italy, IEEE; p. 6136–43.
3. Aghli N, Ribeiro E. Combining weight pruning and knowledge distillation for CNN compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021; p. 3191–8.
4. Krishnamoorthi R. Quantizing deep convolutional networks for efficient inference: a whitepaper. arXiv:1806.08342. 2018.
5. Vaswani A. Attention is all you need. In: 31st Conference on Neural Information Processing Systems (NIPS 2017), 2017; Long Beach, CA, USA.
6. Liu Y. Roberta: a robustly optimized bert pretraining approach. arXiv:1907.11692. 2019.
7. Tay Y, Dehghani M, Bahri D, Metzler D. Efficient transformers: a survey. ACM Comput Surv. 2022;55(6):109. doi:10.1145/3530811.
8. Frantar E, Alistarh D. SparseGPT: massive language models can be accurately pruned in one-shot. In: Proceedings of the 40th International Conference on Machine Learning, 2023; Honolulu, HI, USA, PMLR.
9. Sanh V. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108. 2019.
10. Bhattacharjee A, Hasan T, Ahmad WU, Samin K, Islam MS, Iqbal A, et al. Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. arXiv:2101.00204. 2021.
11. Hinton G. Distilling the knowledge in a neural network. arXiv:1503.02531. 2015.
12. Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y. FitNets: hints for thin deep nets. arXiv:1412.6550. 2014.
13. Li T, Li J, Liu Z, Zhang C. Few sample knowledge distillation for efficient network compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020 Jun 14–19; Seattle, WA, USA; p. 14639–47.
14. Li Z, Ye J, Song M, Huang Y, Pan Z. Online knowledge distillation for efficient pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; p. 11740–50.
15. He J, Ding Y, Zhang M, Li D. Towards efficient network compression via Few-Shot Slimming. Neural Netw. 2022 Mar 1;147:113–25.
16. Yim J, Joo D, Bae J, Kim J. A gift from knowledge distillation: fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017 Jul 21–27; Honolulu, HI, USA; p. 4133–41.
17. Han S, Pool J, Tran J, Dally W. Learning both weights and connections for efficient neural network. Adv Neur Inform Process Syst. 2015;1135–43. doi:10.5555/2969239.2969366.

18. Hu H. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures. arXiv:1607.03250. 2016.
19. Kim J, Chang S, Kwak N. PQK: model compression via pruning, quantization, and knowledge distillation. arXiv:2106.14681. 2021.
20. Anwar S, Hwang K, Sung W. Structured pruning of deep convolutional neural networks. arXiv:1512.08571. 2015.
21. Zhu M, Gupta S. To prune, or not to prune: exploring the efficacy of pruning for model compression. arXiv:1710.01878. 2017.
22. Jacob B, Kligys S, Chen B, Zhu M, Tang M, Howard A, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018 Jun 18–22; Salt Lake City, UT, USA; p. 2704–13.
23. Nagel M, Baalen MV, Blankevoort T, Welling M. Data-free quantization through weight equalization and bias correction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019 Oct 27–Nov 2; Seoul, Republic of Korea; p. 1325–34.
24. Leng C, Dou Z, Li H, Zhu S, Jin R. Extremely low bit neural network: squeeze the l ast bit out with admm. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018; New Orleans, LA, USA; p. 3466–73. doi:10.1609/aaai.v32i1.11713.
25. Zhou A, Yao A, Guo Y, Xu L, Chen Y. Incremental network quantization: towards lossless cnns with low-precision weights. arXiv:1702.03044. 2017.
26. Sakib Ullah Sourav M, Wang H, Sultan Mahmud M, Zheng H. Transformer-based text classification on unified Bangla multi-class emotion corpus. arXiv:2210.06405. 2022.
27. Maheen SM, Faisal MR, Rahman MR, Karim MS. Alternative non-BERT model choices for the textual classification in low-resource languages and environments. In: Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, 2022 Jul; Seattle, WA, USA; p. 192–202.
28. Iqbal MA, Das A, Sharif O, Hoque MM, Sarker IH. BEmoC: a corpus for identifying emotion in Bengali texts. SN Comput Sci. 2022;3:135.
29. Islam KI, Yuvraz T, Islam MS, Hassan E. EmoNoBa: a dataset for analyzing fine-grained emotions on noisy bangla texts. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, 2022; p. 128–34.
30. Faisal MR, Shifa AM, Rahman MH, Uddin MA, Rahman RM. Bengali & Banglish: a monolingual dataset for emotion detection in linguistically diverse contexts. Data Brief. 2024 Aug 1;55:110760.
31. Gu Y, Dong L, Wei F, Huang M. Knowledge distillation of large language models. arXiv:2306.08543. 2023.
32. Agarwal R, Vieillard N, Stanczyk P, Ramos S, Geist M, Bachem O. GKD: generalized knowledge distillation for auto-regressive sequence models. arXiv:2306.13649. 2023.
33. Zhao B, Cui Q, Song R, Qiu Y, Liang J. Decoupled knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022 Jun 8–24; New Orleans, LA, USA; p. 11953–62.
34. Jiang Y, Chan C, Chen M, Wang W. Lion: adversarial distillation of proprietary large language models. arXiv:2305.12870. 2023.
35. Huang Y, Chen Y, Yu Z, McKeown K. In-context learning distillation: transferring few-shot learning ability of pre-trained language models. arXiv:2212.10670. 2022.
36. Li S, Chen J, Shen Y, Chen Z, Zhang X, Li Z, et al. Explanations from large language models make small reasoners better. arXiv:2210.06726. 2022.
37. Liu H, Tam D, Muqeeth M, Mohta J, Huang T, Bansal M, et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. Adv Neural Inf Process Syst. 2022;35:1950–65.

38. Liu Z, Oguz B, Zhao C, Chang E, Stock P, Mehdad Y, et al. Data-free quantization aware training for large language models. arXiv:2305.17888. 2023.

39. Kim J, Lee JH, Kim S, Park J, Yoo KM, Kwon SJ, et al. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. In: Proceedings of the 37th International Conference on Neural Information Processing Systems, 2023; New Orleans, LA, USA; p. 36187–207. doi:10.5555/3666122.3667691.

40. Park G, Park B, Kim M, Lee S, Kim J, Kwon B, et al. LUT-GEMM: quantized matrix multiplication based on luts for efficient inference in large-scale generative language models. arXiv:2206.09557. 2022.

41. Yao Z, Yazdani Aminabadi R, Zhang M, Wu X, Li C, He Y. ZeroQuant: efficient and affordable post-training quantization for large-scale transformers. Adv Neural Inf Process Syst. 2022;35:27168–83.