Advancing Bangla Tense Classification: A Comparative Study of RNN, CNN-LSTM Hybrids, and Transformer-Based Models

By
Mohidul Islam Heera
212-15-14765

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements forthe **Degree of Bachelor of Science in Computer Science and Engineering**
Supervised by

Professor Dr. Md. Fokhray Hossain
Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Mr. Nafiz Ahmed Emon

Lecturer
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

Sep 17, 2025

APPROVAL

This Project titled **"Advancing Bangla Tense Classification: A Comparative Study of RNN, CNN-LSTM Hybrids, and Transformer-Based Models,"**, submitted by **Mohidul Islam Heera**, ID No: **212-15-14765** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17 September, 2025**.

BOARD OF EXAMINERS

Dr. Sheak Rashed Haider Noori — **Chairman**
Professor & Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md Zahid Hasan — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

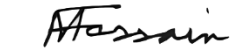
Samia Nawshin — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Arshad Ali — **External Examiner**
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science & Technology University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Professor Dr. Md. Fokhray Hossain ,Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Dr. Md. Fokhray Hossain
Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Mr. Nafiz Ahmed Emon
Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Mohidul Islam Heera
Student ID: 212-15-14765
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project (FYDP)** successfully.
We are grateful and wish our profound indebtedness to **Dr. Md. Fokhray Hossain , Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Natural language processing (NLP)** carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

It is difficult to determine tense in Bangla text due to its morphological complexity, syntactic variation and the absence of annotated corpus. This study uses the BanglaTense dataset with 17,819 sentences labelled into three tenses (Past, Present and Future) by manual annotation for alleviating temporal features learning tasks in low resource language. We compare with the different testbeds (recurrent architecture LSTM, GRU, Bi/LSTM-GRU), Multichannel CNN-LSTM hybrid and transformer-based IndicBERT model). We used a range of pre-processing and class balancing techniques to address the presence of noise and avoid class specific bias. Experiments proved that GRU is a better recurrent neuron when compared to the other ones being 96% accuracy from time, whereas the CNN-LSTM hybrid presents good generalization by boosting Future tense recognition. In our case, the tense-based IndicBERT we induce with embeddings pretrained on a diverse range of Indic languages, becomes the new state-of-the-art model with 97% accuracy and improved F1-scores for all tenses. The results of this study reveal the significance of balancing mechanisms, architecture design, and contextualized representations for tense classification. This work also demonstrates the promise of blended and transformer models in dealing with temporal reasoning when confronted with morphologically rich resourced scarce language such as Bangla.

Table of Contents

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Tense identification is an important task for natural language processing (NLP) and a foundation for many applications including machine translation, dialogue systems, question answering and information retrieval. Determining the (relative) tense of a sentence is a key factor for attaining correct meaning when translating, especially between two languages with different requirements of precise tense. However, for morphologically rich languages like Bangla tense identification becomes a difficult task due to its extensive verb inflections, flexible syntactic structures and a wide range of language expressions.

Despite being the seventh most spoken language in the world, with more than 270 million native speakers, Bangla is still highly under-resourced in the NLP community. The scarcity of annotated data together with the linguistic complexity is the main obstacle in the field of literature. The creation of the BanglaTense dataset is an important contribution to bridge this gap and is a strong resource for developing the temporal feature learning in Bangla as a benchmark for future works.

Early work of Bangla text classification used handcrafted features and statistical models with limited success and modest performance, and often were severely challenged by capturing the long-range dependencies of text as well subtle morphological variations. In the era of deep learning, RNNs like Long-Short Term Memory LSTMs (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Cho et al., 2014) have shown superior power of modeling sequential relations in linguistic data. Extending this, hybrid models which combine CNNs with LSTMs are promising in terms of combining local n-gram feature extraction with long-term temporal pattern learning. Recently, transformer based models, eg., IndicBERT pretrained on large scale multilingual corpora have gained much attention as powerful alternatives, as they generate higher level contextual embeddings which are equipped with the capability to learn temporal and semantic nature of Bangla text with subtle contexts.

In this work, we broadly compare three popular model families – RNN-based architectures (LSTM, GRU, Bi-LSTM, Bi-GRU), a Multichannel CNN-LSTM ensemble and transformer-based IndicBERT – using the recently introduced BanglaTense dataset. We also explore the influences of the dataset balancing and preprocessing options with respect to the model performance. Experiments show that the best performance is achieved by GRU among the recurrent models, CNN-LSTM helps recognition of tense-specificity, in particular for the cases of Future

tense and IndicBERT which, with 97% accuracy, delivers state-of-the-art performance, confirming the effectiveness of pre-trained context-based representations for Bangla NLP.
This work offers a methodological insight along with a empirical evidences to support the development of tense classification in low-resource, morphological-rich languages. The results improve knowledge of temporal feature learning in Bangla as well as provides competitive baselines for future work in low-resource linguistic communities.

1.2 Motivation

Tense is key for meaning in human language, it is to be appropriately recognized in tasks such as machine translation, dialogue system and information retrieval. Failing to recognize the direction of time in a sentence will not only mislead the task intention, but also cause poor system outputs. For instance, coarsely categorizing the Bangla sentence "আমি আগামীকাল যাবো" (I will go tomorrow) as present tense in totality alters its semantics. This is especially problematic for Bangla, a morphologically rich language which has various inflections on verb stems across tense, aspect, mood. Despite of the fact that Bangla is spoken by 274 million people around the world, it still under-represented in the NLP study because appropriate annotated corpora and systematic benchmark dataset are not readily available. As a consequence, far fewer high-quality NLP tools and resources exist for Bangla, leading to relatively-poorer quality Bangla NLP solutions as compared to resource-rich languages such as English.

This work is inspired by both the linguistic and technical challenges as well as my wish to further develop computational tools for my mother tongue. Enhancement of tense classification in Bangla will consequently contribute in developing more gender inclusive technologies that bear the voices of Bangla speakers and will serve to facilitate applications in translation, literacy and digital communication. With the prevalence of more sophisticated architectures like topologies featuring RNN's, CNN-LSTM hybrids and transformers such as IndicBERT, it is now the right time to consider which method is more successful in incorporating the temporal richness of a complex language like Bangla. Through this work, we hope to provide powerful baselines to compare with and deeper insights for closing the resource-rich- vs-resource-poor languages gap and facilitate more meaningful advancement for Bangla NLP.

1.3 Problem Statement

Tense classification in Bangla is not a trivial task since the verb forms are diverse and the language grammar is intricate and the sentences are adaptable. English and Bangla also differ in verb endings and the lack of large labeled datasets such that the provided models cannot adequately separate the tenses (Past, Present, or Future). Classical machine learning methods rely on hand-crafted features, which cannot capture the whole computation power of the linguistic patterns in Bangla.

Although the deep learning architectures including LSTM, GRU, Bi-LSTM, and Bi-GRU adequately do the job better than the classical machine learning approaches, there are long term dependencies and contextual nuances understanding failures in Bangla text. The code-mixed (Bangla-English) text and imbalanced dataset add more challenge to the correct tense identification.

The proposed project seeks to create a robust tense identification system for the Bangla language using deep learning architectures (LSTM, GRU, Bi-LSTM, Bi-GRU, and CNN-LSTM fusion) and a transformer model (IndicBERT). The research aims to compare the above-mentioned models and obtain the most optimal method for the identification of the correct tenses in the Bangla language while handling the issues of dataset imbalance and linguistic complexity and provide a firm foundation for future work in the field of Bangla natural language processing (NLP).

1.4 Objectives

The primary objectives of this project are:

- In order to create an automatic tense classification system in Bangla texts concentrating in Present, Past, and Future tenses.
- To compare using RNNT based models (GRU, LSTM, Bi-LSTM, Bi-GRU) and transformer-based model (IndicBERT).
- To successfully perform tokenization, normalization and splitting of the BanglaTense dataset for the purpose of training, validation and AI test.
- To test the performance of models based on standard classification metrics (accuracy, precision, recall, F1-score, confusion matrix).
- Our objectives are: (1) to identify the optimal architecture for Bangla tense classification, and (2) to pave the way for saturating the gains of transformer-based methods against RNNs.

To work on a relatively unexplored task in the context of Bangla Natural Language Processing (NLP) by contributing with baseline results

1.5 Methodology

In this paper, I take an end-to-end approach to create a high-quality tense tagging system for Bangla texts by conducting experiments with RNN and Transformer-based models. The approach has been broken down into several main steps such as data set preparation, pre-processing, model development, training, and testing.

The analysis procedure is started with one BanglaTense dataset that provides Bangla sentences labeled with either Present or Past or Future tense. To provide fair comparison, the dataset was split into training (70%), validation (15%), and testing (15%) sets. Each of the sentences was prepared by normalization, tokenization and padding to have a standard input format. Text was tokenized and embedded into dense vectors with word embeddings for use in deep learning models. A wide variety of architectures were used for model development. The RNN family comprises GRU, LSTM, Bi-LSTM and Bi-GRU, as they have been shown to successfully capture dependencies across text sequences. Moreover, we used IndicBERT transformer

model, whose multilingual pretraining and better contextual understanding of Bangla were used. These provided a fair comparison between the traditional sequential architectures and the transformer-based models of today.
The models were trained on the training subset and tested on the validation subset for hyperparameter fine-tuning. The loss function used for training is the categorical cross-entropy loss function and the optimizer used was adam optimizer to train learning. The models was trained over repeated epochs and early stopping was enforced to prevent overfitting and ensure generalization.
The third and final stage performed the evaluation with the test subset, computing the usual classification metrics, like accuracy, precision, recall, F1-score, and confusion matrix analysis. Convergence curves were examined as well to investigate rates of convergence. We conducted comparative experiments that were able to demonstrate which strengths and weaknesses of each model, and in which sense they can be different in terms of the generalization power of recurrent and transformer models.
This systematic approach has resulted in a strong backbone for Bangla tense classification and provided valuable insights about certain new age architectures, such as modern transformer models, and their performance in contrast with traditional RNN based methods.

1.6 Project Outcome

This paper proposes a method to systematically classify Bangla tense with influence from both recurrent neural network (RNN) and transformer model. We use the Bangla Tense dataset, which is a pre-categorized fixed sample of sentences annotated as Present, Past and Future. Other minework After retrieving the dataset, further preprocessing was performed for cleaning noisy samples, normalizing text, tokenization, padding sequence etc. Word embeddings were created to provide textual data in dense numbers for input to deep learning models.

Two types of models were chosen for experimental study. The first set is composed of RNN based models (GRU, LSTM, Bi-LSTM, Bi-GRU) which are selected due to their capacity to accommodate sequential dependencies in bangla sentences. The second category utilizes a transformer based model called IndicBERT, which incorporates multilingual pretraining to offer more informative context sensitive representations. Combine together, these models guaranteed an unbiased comparison between traditional and modern designs.

The training set was further divided into separate training and validation sets used to train the models. Categorical cross-entropy loss function and Adam optimizer. Early stopping was performed to avoid overfitting. The final evaluation was based on the test set which is reported with accuracy, precision, recall, F1-score, and confusion matrix analysis. Comparison analysis also corroborated the superior performance of IndicBERT over RNN-based models, demonstrating the effectiveness of transformer-

based methods for Bangla tense classification.

1.7 Organization of the Report

This report is structured to provide a comprehensive understanding of the project " Biomedical Entity Extractionand QA with RAG: A StepTowards Intelligent Healthcare," detailing every aspect from motivation to implementation. The organization is as follows:

Chapter 1. Introduction: This chapter introduces the project, including the background, problem statement, motivation, objectives, and a brief overview of the methodology. It also outlines the anticipated outcomes and the structure of the report.

Chapter 2. Background: This section provides the foundational knowledge necessary to understand the project, summarizes key works in BioQA with a focus on BioASQ systems, highlighting their architectures, techniques, and limitations.

Chapter 3. Research Methodology: This chapter describes the methodology adopted for the project, Describes the modular system architecture, models used, and task-specific processing for each question type. It also includes system design, task allocation, and the proposed project plan.

Chapter 4. Implementation and Results: This chapter focuses on the implementation details, including the environment setup, performance evaluation, and comparative analysis of the models used for each QA module. It also discusses the results obtained and their implications.

Chapter 5. Engineering Standards and Design Challenges: This section discusses the compliance with relevant software and hardware standards, the challenges encountered during the project, and the impact on society, environment, and sustainability. Ethical considerations and the sustainability plan are also elaborated.

Chapter 6. Conclusion: The concluding chapter summarizes the project, highlights its limitations, and proposes potential areas for future work.

Chapter 2

Literature Review

2.1 Introduction

A language is one of the most sophisticated instruments of human communication through which we express what we think, feel, and intend. Tense is one of very important aspects of natural language, which has a very significant effect on the temporal context of sentences. Proper identification and classification of tense is particularly crucial in machine translation, sentiment analysis, question-answering, dialogue systems, which have shown that meaning of a statement can dramatically change due to the tense in which it is made.

English and other popular languages have been extensively researched in NLP; however, Bangla, the seventh most spoken language, has largely been neglected in computational linguistics. The rich morphology and the free word order in sentences bring complexity in tense classification in Bangla, which is important and at the same time a challenging task. To this end, in this work, attention is made to propose a robust deep learningbased framework for tense prediction of Bangla sentences.

We employ a pre-processed, normalized BanglaTense dataset tailored for our purpose in this work. For the problem of class imbalance, data balancing techniques were employed to guarantee balanced training across tense. Different deep learning structures such as GRU, LSTM, Bi-LSTM, Bi-GRU, CNL-LSTM hybrids were experimented to see their efficacy in capturing temporal patterns between sentences on the Bangla language benchmark compared to transformer-based IndicBERT model.

The experiment protocols including the hyperparameters and evaluation metric were the same for all models so that the fair comparisons could be guaranteed. By contrasting classical recurrent architectures with state-of-the-art transformer-based models, this work acts as a benchmark for known methods, and brings into focus the advantages and weaknesses of different techniques.

Finally, this work contributes to the progress of Bangla NLP research, in the form of a methodological approach, comparative studies, as well as interpretation of tense classification. Apart from academic significance, this research can be used as a basis to develop some applications used in real world, e.g., Bangla grammar checker, intelligent chatbot or educational tools helping learners and users of Bangla language.

2.2 Literature Review

From rules-based to deep learning models such as LSTM, GRU etc which are specialized on sequence modelling, its been an adventure! But the language being too complex, due to which someone have to look for a new approach for the Bengali language like not much work has been done on it.

One [1] paper are written on Bangla documents in sentences classification by Random Forest and Decision Tree and have achieved a highest accuracy of 89.42%) and one subsequent to LSTM network by 88.2%). It further validates that ML and DL approaches are to be complementary to each other when Bangla text classification. Another work [2] narrowed down class imbalance problem by data augmentation, focal loss, and ensemble over tasks like fake news detection, sentiment analysis and song classification. It obtained over 90% F1-scores of minority class on all the tasks. A recent work [3] used CNN and LSTM for Bangla text document classification. The LTS model adopted from and setting the HOLLYWOOD smashing the highest record with accuracy of 95.42% and was the highest performer than that of CNN model in the Prothom Alo dataset. Our results indicated that the character level encoding can be better adopted to enrich deep learning for document classification. Independently, a similar task was addressed on an application-based setting in [4] where the news articles were classified into 4 classes namely business, science and technology, entertainment and health from the News Aggregator dataset downloaded from the UCI Machine Learning Repository- A Classification accuracy of up to 89.5% was achieved by Multinomial Naive Bayes and the excellent differentiation on categorization shows in the "Entertainment" category.

In [5], the Bangla words were spited based on grammatical prefix using supervised machine learning KNearest Neighbors, Naive Bayes, random forest, logistic regression, XGBoost, and SVC methods. The models did a good amount of word classification (they were trained on a very small set (500+) and the obvious happened) which made us see the Bangla grammar somewhat. In an another work [6], extreme guilt and grave offence in Bangla were detected in Logistic Regression, SVM and Multinomial Naive Bayes. In addition, the highest accuracy obtained with ensemble methods AdaBoost, Max Voting was 91 % and 92 % whereas JR-AdaBoostPCA generated the highest accuracy of 94 %. In [7] TF-IDF feature extraction with DR techniques were used to classify more than 400k Prothom Alo news articles under 9 categories and MLC learners proved their handling of high throughput text classification by achieving accuracy of 93.76%. On Bangla texts, [8] similarly managed to identify offensive language in Bangla texts by playing with multiple algorithms: skill-groups of Linear SVC, Logistic Regression, Multinomial Naive Bayes, Random Forest and RNN with LSTM cells. The highest performance was demonstrated by the RNN model having 82.20% which shows its potential for identifying mul ticl ass abusive content.

In one of our previous papers 9. % was achieved by the Multi-layer Perceptron with the best accuracy, leaving Naive Bayes (94.1%) and J48 (93.9%) slightly behind. This result demonstrates the important role of the character level between-genre variations in medical text classification. This once more (10), is a b a ng la statement confirmation under statistical N-gram modeling. When run against one million tokens as a corpus, the following result was obtained: syntactic validation ratio of about 93%, indicating that N-grams can actually generalize to grammatical structures.

Previous works are mostly based on conventional models, and they are not capable of modeling such a high-level similarity feature of the inter-subject. To address this gap, in this paper, we move towards a more powerful approach for Bangla text classification.

2.3 Gap Analysis

However, several open research challenges persist for Bangla text classification in both conventional ML and DL methods have been reported. Current investigations [1]–[10] are almost exclusively concerned with generic tasks like document categorization, sentiment analysis, abusive content detection, or medical text classification. Although they have shown high accuracy in many areas, these do not consider the tense classification as linguistic challenge which play a vital role to understand the temporal context in Bangla.

The majority of previous work is based on traditional models or shallow neural networks, e.g., Random Forest, Naive Bayes, CNNs and LSTMs. Though effective for coarse categorization, these methods do not have the emphasis on semantic feature learning which is essential to capture the complex morphological and syntactic variations in Bangla verbs. In addition, many datasets such as those in (Srikanth and Hulden, 2004), (Skousen et al., 2008) were small and all were from a specific domain or were hand constructed not necessarily in the light of tense-specific annotations, which imposes limitation to the generalisability of these models.

To this end, the present work investigates tense classification in Bangla using the recurrent neural networks (GRU, LSTM, BiLSTM, BiGRU) and a transformer-based model (IndicBERT). The research will directly contribute to the unexplored task of Bangla tense detection and it will show the contribution of both sequential and contextualized embeddings in achieving more robust baselines, further to demonstrating more powerful encoding with transformer models in comparatively capturing temporal semantics comparing to the traditional or shallow deep learning techniques.

2.4 Summary

This chapter reviewed the literature of its underlying theories and the prior literatures on Bangla text classification and tense detection. The day began with discussions around Natural Language Processing (NLP) and how it can be applied on a low-resource language such as Bangla- which suffers from both rich morphology, divergent syntactic structures as well as a scarcity of annotated resources. Faster

progress for the purposes is seen in the tasks of the Bangla NLP studies which gets more focus that are on sentiment analysis, fake news detection abusive content detection general document classification and other. These works adopted different techniques, namely classical machine learning algorithms such as Random Forest, Naïve Bayes and SVM, and more advanced neural architectures like CNNs and LSTMs, achieving very good performance in their target application domains.

Related Work The literature survey shows that despite the superior performance of deep learning models over traditional methods on large-scale corpora-based Bangla NLP some work has been conducted but they are more in the area of surficial classification tasks. Tense classification: Only a few works have tackled tense classification even though this task is essential for understanding the meaning conveyed by the temporal expression and one that has potential impact on downstream applications such as machine translation, summarization and question orwye task answering. Previous works were also constrained due to dataset size, class imbalance or using hand-crafted features leading them to be not general enough for different Bangla texts. These limitations also signal a need for better methods that can capture higher-order contextual relationships among words in language.

The gap analysis indicated that state of the art approaches, while valuable, underuse the modern architectures for tense detection. Both shallow and traditional models struggle to capture the finegrained verb conjugation and time reference conveyed in the Bangla data. T o fill this gap in the literature this paper investigates both the RNN architectures (GRU, LSTM, Bi-LSTM, Bi-GRU) and the transformer-based models (IndicBERT) side-by-side by conducting a comparative analysis between the seq uential and sequential embeddings. We hope that this work will help increase the state of Bangla NLP, as this work sets the baselines for tense classification and shows that how well the transformer model handles complex linguistic phenomena better than its predecessor. The next chapter describes the research methodology that was used to put those approaches into practice and to evaluate them.

Chapter 3

Research Methodology

3.1 Methodology

3.1.1 Overview

In this chapter, we present the methodology that we used to carry out the research on Bangla tense classification. The work proceeds by describing the BanglaTense dataset creation process, and further elaborating on the pre-processing steps that prepare the data by cleaning, normalizing and balancing. Then, architecture of different models like LSTM, GRU, Bi-directional, CNN-LSTM, IndicBERT are described in model selection. The chapter then describes the training process, hyperparameters, optimizers and evaluation metrics that were used to ensure fairness and consistency between all experiments. At last, we demonstrate the evaluation methodologies to analyze performance and also tabulate accuracy, precision, recall, F1-Score and Confusion Matrix Analysis. Collectively, these strategies reflect rigorous methods for securing credible, generalized inferences.

3.1.2 Proposed Methodology

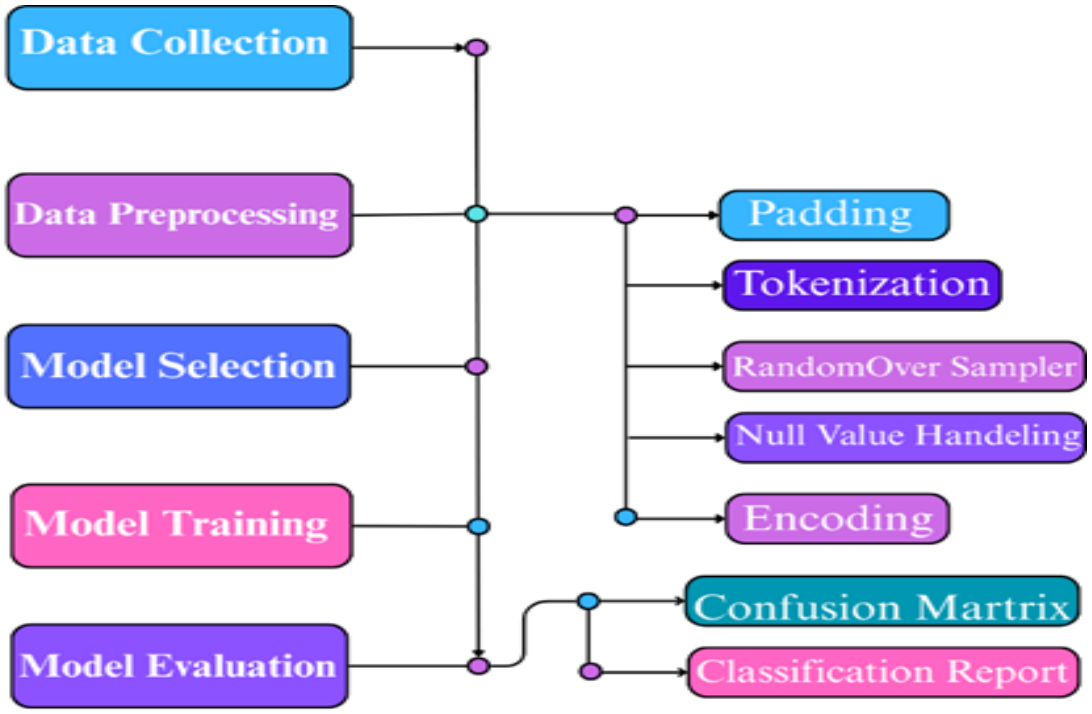


Figure 3.1 Proposed Methodology

The recommended approach is a process-oriented pipeline that allow us to create, train and test machine learning model in a systematic way. It consists of 7 major stages:
Data Collection: Data collecting from relevant sources are executed. The quality and quantity of such data is very important on the global result of the model.
Data Preprocessing : Data is preprocessed before it is fed into the model and the datasets are cleaned and structured.

- Padding is used for making input sequences equal length for text or sequence data this is very important.
- Tokenization is the process of converting raw text into tokens (words, subwords, or characters) so that it is compatible with the model.
- RandomOver Sampler balances unbalanced datasets by oversampling the cases of the minority class.
- Missing Data can be handled by Null Value Handling to maintain consistency.
- Encoding converts categories into numbers for model training.

Model Selection: Then, an appropriate machine learning or deep learning model is selected according to the type of problem (e.g., classification, regression, NLP).

Model Training: The trained model and the preprocessed data are both input into the selected model subsequently. The model is trained in this stage to find and to understand the patterning and relationships moving within the data.

Model Evaluation: Finally, the trained model is evaluated to assess its quality and trustworthiness.

- Confusion Matrix – Performance of classification model against actual class values is measured using confusion matrix.

Classification Report contains precision, recall, f1-score, support and accuracy.

3.2 Detailed Methodology and Design

The methodology for this research follows a structured, step-by-step process, beginning with data collection and culminating in model evaluation, as illustrated in the accompanying diagram:

I. Data Collection

We utilized publicly available BanglaTense dataset for this study, which includes 17,819 Bangla sentences labeled manually with the verb tense and they are classified into 3 (three) Past: 5,629 sentences, Present: 6,101 sentences and Future: 6,089 sentences which achieved approximately balanced class distribution. The sentences came from a mixture of sources including blogs, social media and magazines and newspapers, of both the popular and the scholarly variety. Such diversity contributes to the ability of the dataset to capture real-world scenarios of modern Bangla language use, therefore leading to improved generalization and robustness of the models trained on it. Since Bangla is primarily a pro-drop language, tense

classification is inherently difficult, making this dataset extremely useful for NLP taks involving temporal reasoning. Though there were little class imbalances available from the start, thoughtful preprocessing alleviated their effect on model learning and evaluation.

The corpus additionally contains characteristic samples of various writing styles and idiomatic expressions in the language, as is appropriate for learning contextual variations in Bangla by models. Additionally, the data may be useful for tasks like syntactic parsing, sentence verification or temporal ordering analysis and therefore can be used for more than just simple tense identification. Summary of the class distribution is shown in fig 3.3, and the sample sentences in Fig. 3.2 gives an overview on the dataset.

	Sentence	Tense
0	আমি প্রতিদিন সকালে হাঁটতে যাই	Present
1	তুমি আগামী বছর বিশ্ববিদ্যালয়ে ভর্তি হবে	Future
2	সে গতকাল একটি নতুন বই কিনেছিল	Past
3	আমরা এখন ক্লাসে পাঠ শুনছি	Present
4	তারা আগামী সপ্তাহে ঢাকা সফরে যাবে	Future

Figure 3.2 Dataset Overview

Figure 3.3 illustrates the distribution of sentences across the three tense classes in the BanglaTense dataset. The dataset is fairly balanced, with 5,629 Past, 6,101 Present, and 6,089 Future tense sentences, highlighting a nearly equal representation for each class. This balanced distribution ensures that models trained on this dataset can learn tense classification without significant bias toward any particular class.

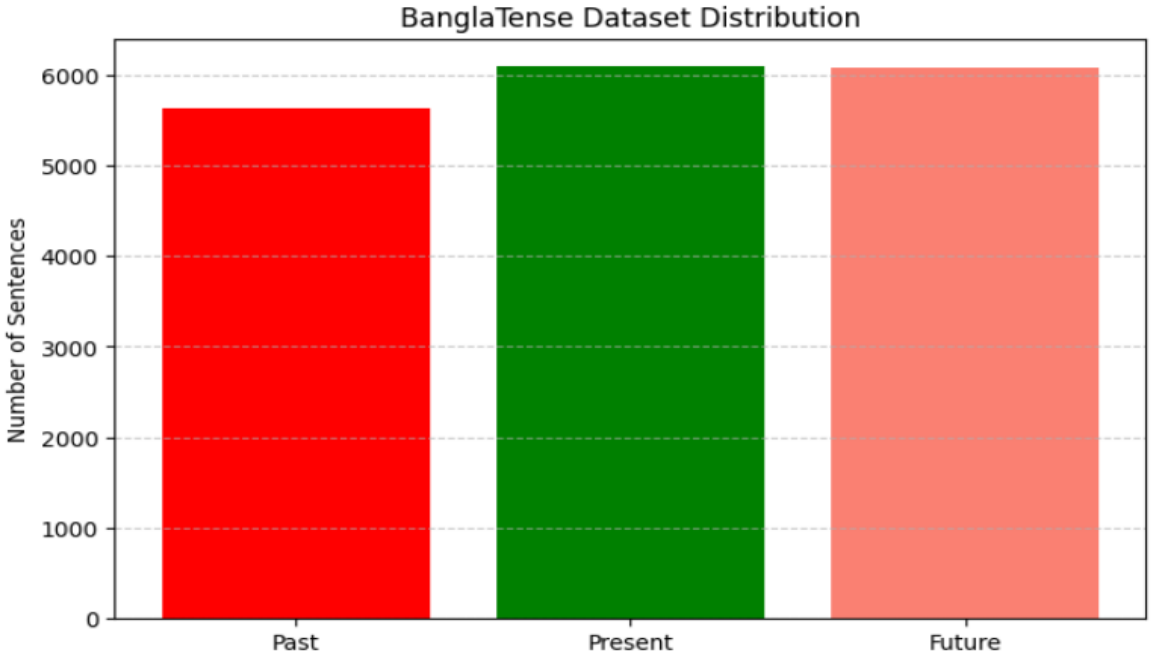


Figure 3.3 Data Distribution

II. Preprocessing

We had a complete preprocessing pipeline for the BanglaTense dataset so that it was clean, consistent and ready to use in deep learning models. Firstly, sentences were deduplicated to avoid redundancy, and noisy elements (e.g., punctuations, emojis, slangs, abbreviations, special characters) were filtered to keep the text understandable. To normalize the text input, we used the CSE BUET NLP Normalizer to fix the Unicode inconsistencies, remove the trash symbols and maintain uniform treatment of the characters of Bangla. This was essential to avoid tokenization errors and to preserve the semantic consistency of the dataset.

Tokenization was performed using the Keras Tokenizer from TensorFlow, which converted sentences to sequences of integers corresponding to tokens. This procedure led to a vocabulary of 12,309 tokens. To ensure that input sequences were of equal length, sequences were padded or shortened such that all input sequences were of length 100, with transformer based models adopting a truncation length of 128 to balance computational performance. This type of sequence normalization is necessary for batched deep learning models and leads to a consistent shape across instruments and between training and evaluation.

The categorical "Past", "Present", and "Future" were represented as numerical values, and then turned to one-hot encoding for multiclass classification. The original dataset had a class imbalance because some tenses were underrepresented; therefore, RandomOverSampler was used to balance the distribution. This made the model more generalizable across all classes and also prevented any bias towards the

majority class. The balanced distribution and imbalanced distribution are visualized in Fig. 3.2, which demonstrate the oversampling effect.

Lastly, we conducted a stratified 80:20 train-test split to guarantee that all tense categories were uniformly represented between the training and test examples. This strategy avoided biased evaluation and was conducive for robust model evaluation. Both the imbalanced and balanced datasets were preprocessed and compared to see the effect of data augmentation and balancing on model performance. These pre processing steps improved in the quality of the input data, minimized noise and laid a robust platform for developing valid Bangla tense classification models.

III. Model Selection

The primary objective of this work was to explore and compare different deep learning architectures to identify an efficient and robust model for Bangla verb tense classification. Several architectures were selected to capture the sequential, morphological, and contextual patterns inherent in Bangla text, ranging from recurrent models to hybrid and transformer-based models.

The first category consisted of recurrent neural networks, including the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models (Fig. 4 and Fig. 5). GRU Architecture: Shows a unidirectional GRU model with an embedding layer, a 128-unit GRU layer, a dense hidden layer of 64 ReLU units, and a softmax output layer for classifying Bangla tenses. LSTM Architecture: Illustrates a unidirectional LSTM model with similar structure to GRU, capturing long-range dependencies in sequential Bangla text data. These architectures are particularly well-suited for sequence modeling, as they can capture long-range dependencies in text. Both models started with an embedding layer to transform input words into 128-dimensional dense vectors, preserving semantic relationships. This was followed by a recurrent layer of 128 units, a dense hidden layer of 64 ReLU-activated units, and a softmax output layer for classification into three tense categories: Past, Present, and Future. The input sequences were fixed at 100 tokens with a vocabulary size of 12,309. The LSTM model contained 1,715,587 trainable parameters, while the GRU model had 1,683,075 parameters, with the majority concentrated in the embedding layer.

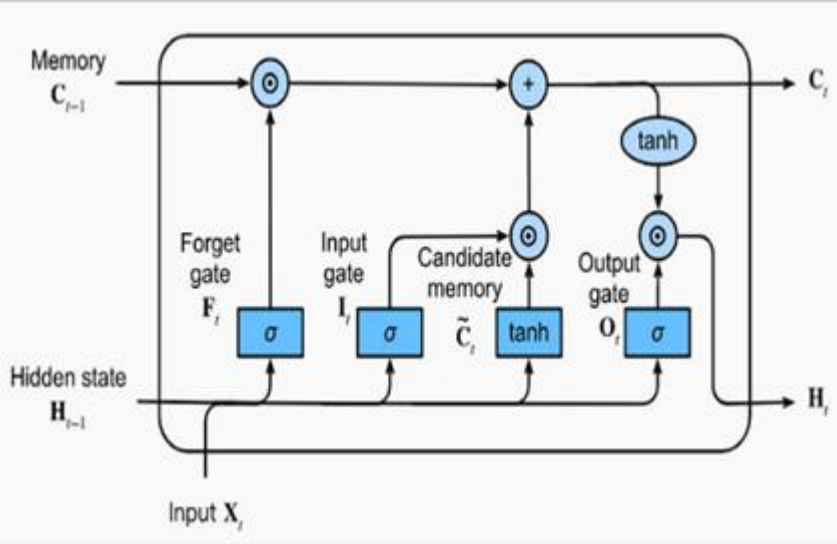


Figure 3.4 Architecture of LSTM Model

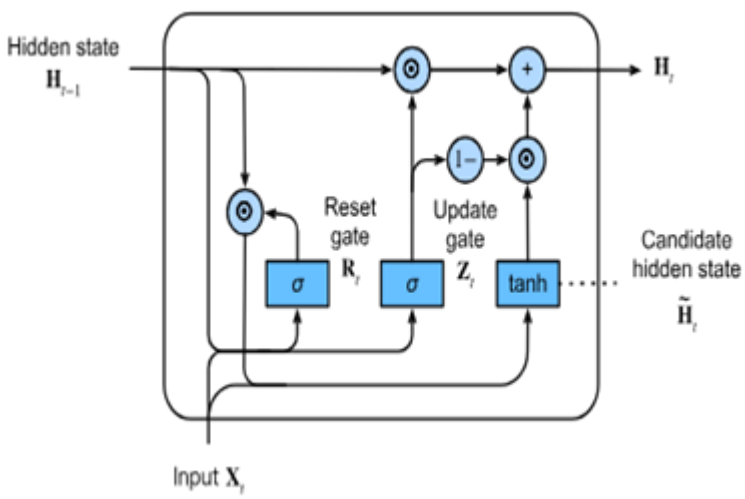


Figure 3.5 Architecture of GRU Model

To enhance the representational power, a Multichannel Combined CNN-LSTM architecture was also proposed (Fig. 6). Shows a hybrid CNN-LSTM architecture for Bangla tense classification. Three parallel convolutional layers with kernel sizes 3, 4, 5, respectively, take the word embeddings as input and extract local n-gram features. The outputs are then propagate through maxpooling1D and a dense with 16 neuron units before being joined and $\textbf{re}structuring the concatenated features in $\hat{\bold{x}}, mixed\ anyLSTM, with 128 units$ to model sequential relationships. At the end, a 64-unit dense layer with ReLU as the activation function

refines the representation, and softmax is used for the output layer for classification. This architecture directly aims at modeling morphological as well as syntactic properties of Bangla text which in turn help to learn contextual rich features from the input sentences. This hybrid model integrates three parallel convolutional layers with kernel sizes of 3, 4, and 5, capturing local n-gram features from the shared embedding layer. Each convolutional branch used filter dimensions of 32, 64, and 128, respectively, followed by MaxPooling1D, flattening, and concatenation. The output was then passed to an LSTM layer with 128 units for sequential dependency modeling, followed by a dense layer with 64 units and a softmax output for classification. This model added 10,449,763 trainable parameters, combining the strengths of convolutional feature extraction with recurrent sequence learning, specifically designed to capture the morphological and syntactic complexity of Bangla.

Figure 3.6 Multichannel Combined CNN-LSTM Model Architecture

In addition to recurrent and hybrid models, Bidirectional LSTM and GRU architectures were considered to enhance context understanding by processing sequences in both forward and backward directions. Finally, a transformer-based IndicBERT model (Fig. 3.8) was employed,Represents bidirectional sequence models that capture context from both past and future tokens, enhancing tense prediction accuracy and leveraging the ALBERT architecture pretrained on multilingual corpora including Bangla. With subword-level tokenization and deep contextual embeddings, IndicBERT provided superior semantic understanding, making it highly suitable for Bangla NLP tasks.

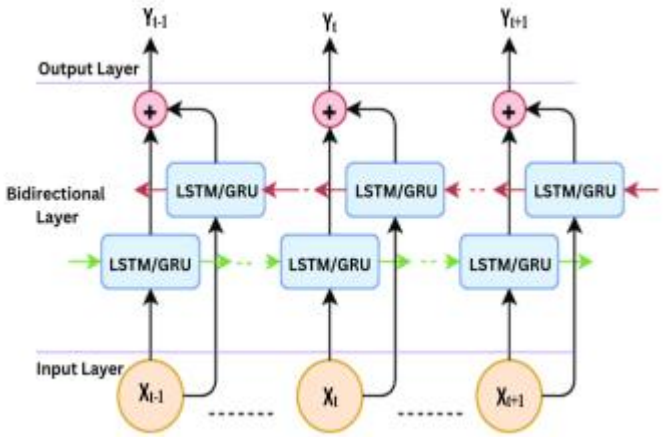


Figure 3.7 Bidirectional LSTM/GRU Architecture

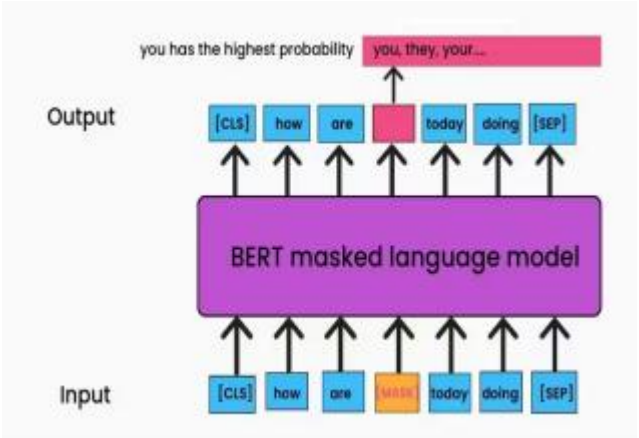


Figure 3.8 IndicBERT Architecture

Transformer-based IndicBERT model's architecture fine-tuned for Bangla tense classification is shown in figure 3.8. It uses subword—level tokenization to handle the complex morphology of Bangla words, and models long -range context

dependencies between sentences. The architecture uses self-attention layers to model token dependencies, where information from previous and future tokens are used when building predictions. A last classification layer with softmax generates the probabilities for three tense classes, which is appropriate for the NLP tasks containing abundant semantic information.

We judged all models not only on their classification performance, but also efficiency of computation and strived to find an architecture that maximizes prediction accuracy while being efficient for training and inference time. By this comparative method, the study aims at uncovering the most effective way for robust Bangla verb tense classification by juxtaposing sequence, hybrid and transformer-based approaches to address the linguistic complexity of Bangla.

IV. Model Training

All generated models (GRU, LSTM, Bidirectional LSTM/GRU, CNN-LSTM hybrid, and the transformer-based IndicBERT) were trained using the same experimental configuration, so that comparison remains fair. All models were trained using Adam optimizer for adaptive learning of each parameter, enabling faster and stable convergence. A learning rate of 0.0001 was applied to recurrent and hybrid models, and learning rate of 2e-5 was used for the IndicBERT model as it has transformer layers. We used the categorical cross-entropy loss function as the task was to perform multi-class classification over three tense categories (Past-Present-Future) and accuracy was the main evaluation criterion.

We trained our deep learning models for 10 epochs with batch size of 32 and IndicBERT with batch size of 64. Twenty percentage of the training data was used as validation to check the model when training and avoid overfitting. Optimization for the CNN-LSTM hybrid focuses on more train- ing of the convolutional layers to capture local n-gram features and therefore sequential learning in the LSTM layer. Weights were continuously updated by epochs through backpropagation, and model was checkpointed with the best validation accuracy. In Table 3.1 mention, Hyperparameter Settings for Model Training.

Table 3.1 Hyperparameter

Parameter	value
Learning Rate	2e-5
Optimizer	adam
Loss Function Categorical	Cross-Entropy
Epochs	10
Batch Size	32

Validation Split	20%

In conclusion, all models trained well after following the training strategies and learned meaningful representations from the data, which are robust to class imbalance, for Bangla verb tense classification. With all models, similar precise convergence was achieved thanks to stable usage of the Adam optimizer, cross-entropy loss, and validation-based checkpointing. This fine-tuning plan enabled both classical recurrent models and the transformer architecture-based IndicBERT to perform well, and provided a strong baseline for the evaluation and comparison of models effectiveness.

V. Model Evaluation

After completing the training phase, all models were systematically evaluated to assess their effectiveness in Bangla verb tense classification. Both quantitative and qualitative measures were used to provide a comprehensive understanding of model performance. Standard metrics such as accuracy, precision, recall, and F1-score were employed to capture the models' ability to correctly classify sentences into Past, Present, and Future tenses.

These evaluation metrics were computed as follows:

- Accuracy:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

- Precision:

$$Precision = \frac{Number\ of\ overlapping\ n{-}grams}{Total\ n{-}grams\ in\ prediction}$$

- Recall:

$$Recall = \frac{Number\ of\ overlapping\ n-grams}{Total\ n-grams\ in\ reference}$$

- F1-score

$$F1 = \frac{2.Precision.Recall}{Precision+Recall}$$

The testing was performed on the held-out test set (which retained the original class distribution) to match real world conditions. Performance was evaluated for the recurring models (GRU and LSTM, and their bidirectional forms) for their capacity to capture sequential dependencies and contextual information cross-sentence-wise. As BERT is a transformer-based model, we also evaluated if IndicBERT can well exploit deep contextual embeddings and subword tokenization for effective generalization.

In addition, further testing was carried out on both the unbalanced dataset and the balanced dataset to investigate the effect of number of classes on prediction accuracy. Confusion matrix was constructed to display misclassifications and emphasize difficult classes. This also helped to see the strengths and weaknesses of the models, and which architectures better addressed the morphological and syntactic complexities of Bangla.

In the end, the evaluation phase made sure that all architectures were fairly and thoroughly compared, and it was clear which model(s) were more practical to use in Bangla NLP tasks.

3.3 Project Plan

The detailed project plan shown in Figure 3.9:

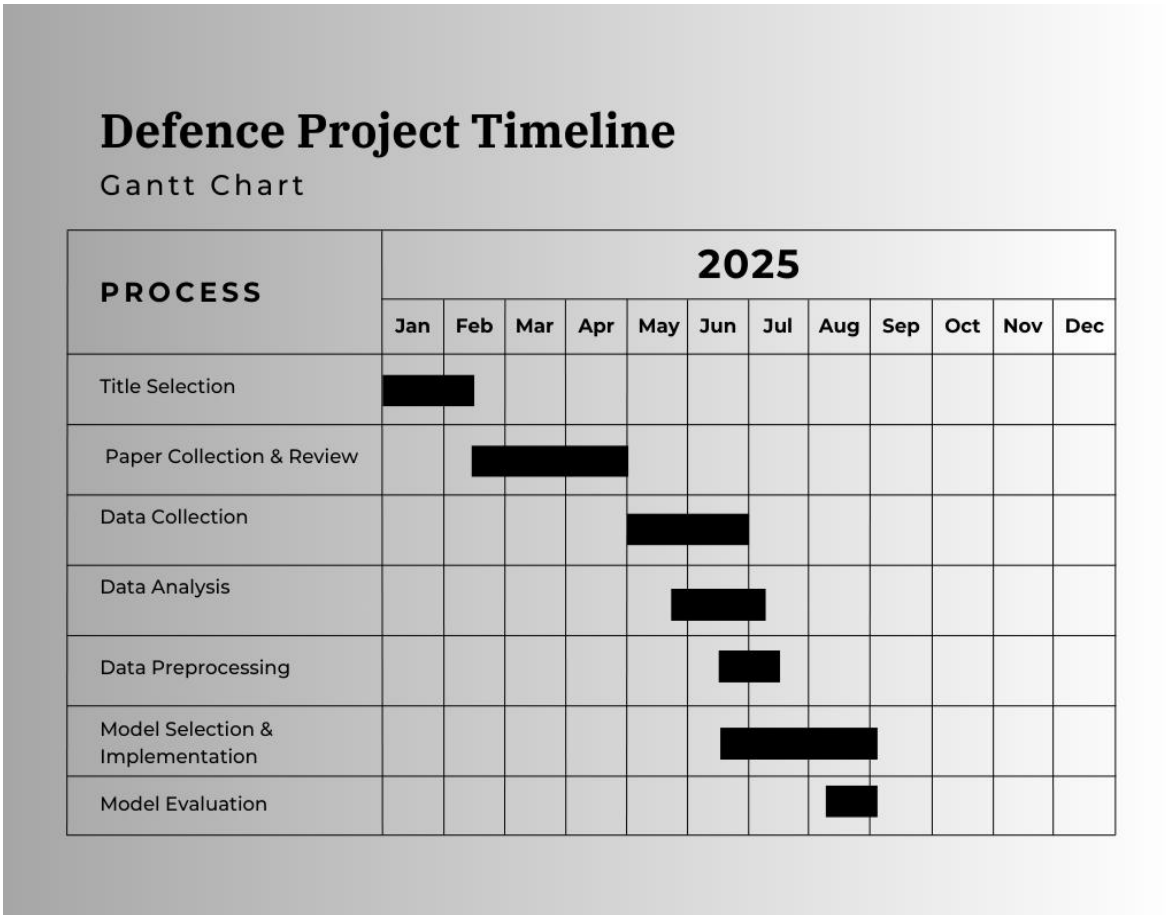


Figure 3.9 Project Plan

3.4 Task Allocation

There is no member in my team. So, all the tasks are completed of my own.

3.5 Summary

In this Chapter, I present the full research methodology of Bangla verb tense classification. It includes how to prepare dataset, preprocessing steps to clean and normalize the text, class imbalanced techniques. We show the choice of different deep learning based models such as GRU, LSTM, Bi-LSTM, hybrid CNN-LSTM, IndicBERT andtrianing with the selection of Adam optimizer and cross entropy loss. Finally, we describe the evaluation methods and the performance metrics that we consider in evaluating how well each model reinvents the linguistic structure and sequential organization of the text in Bangla.

Chapter 4

Implementation and Results

4.1 Environment Setup

As for creating, and testing Bangla verb tense classification models, Python 3.11 was used because it provides support to many deep learning and natural language processing libraries. The entire pipeline from data preprocessing, model training to evaluation of GRU, LSTM, Bi-LSTM, Bi-GRU, CNN-LSTM hybrid, and IndicBERT was developed and run on Google Colab on 1 × NVIDIA Tesla T4 GPU, which offers ample parallelism and enough memory to run both sequence models and fine-tune the transformer effectively.

For the implementation of recurrent and convolutional neural networks TensorFlow and Keras were used and PyTorch and Hugging Face Transformers were used to fine-tune the IndicBERT model. Data preprocessing (tokenization, padding, sequence handling) was performed using TensorFlow Keras Tokenizer, NumPy and Pandas. The results were visualized by using Matplotlib and Seaborn to train-validation los curves, confusion matrices, and class distribution plots.

This approach allowed to experiment with several deep learning and transformer architectures using a flexible, modular and computationally efficient environment when considering reproducibility and comparison of performance.
The detailed Environment setup is described in Table 4.1.

Table 4.1: System Environment Configuration

Component	Details
Programming Language	Python 3.11
Development Platform	Kaggle Notebook with 2 × NVIDIA Tesla T4 GPUs
Deep Learning Stack	TensorFlow, Keras, PyTorch, Hugging Face Transformers
Evaluation Metrics	Accuracy, Precision, Recall, F1-score
Data Handling	Pandas, NumPy
Visualization	Matplotlib, Seaborn (optional)
Models Implemented	GRU, LSTM, Bi-LSTM, Bi-GRU, Multichannel CNN-LSTM, IndicBERT

4.2 Performance Evaluation and Comparative Analysis

This section contains results on LSTM, GRU and Multichannel Combined CNN-LSTM models for both imbalanced and balanced BanglaTense dataset. The evaluation criteria were accuracy, precision, recall, F1, and confusion matrix analysis. Additionally, effects were examined from performance of data balancing and model design.

LSTM was the example where it had the fast initial learning well within a single epoch (above 95% training accuracy), and then after 5 the validation loss finally started to nudge up a bit, minor indication of early overfitting. The Loss/Accuracy (Epochs) trend of LSTM model shown in Figure 4.1. Nevertheless, final validation accuracy was about 93% so the generalization was fairly good. The graph of training Vs Validation Loss and Accuracy as shown in Figure 4.2. For the GRU, the case in point was barely inferior in terms of convergence but did exhibit more stable validation performance at around 94% with a corresponding advantage to class imbalance. The Multichannel Combined CNN-LSTM model outperformed the other two by maintaining a proximal association between the training and validation curves in Figure 4.5, while obtaining more than 94% validation accuracy. The fact that it is catching both local and sequential patterns is an interesting generalizing direction of the model on both imbalanced situations.

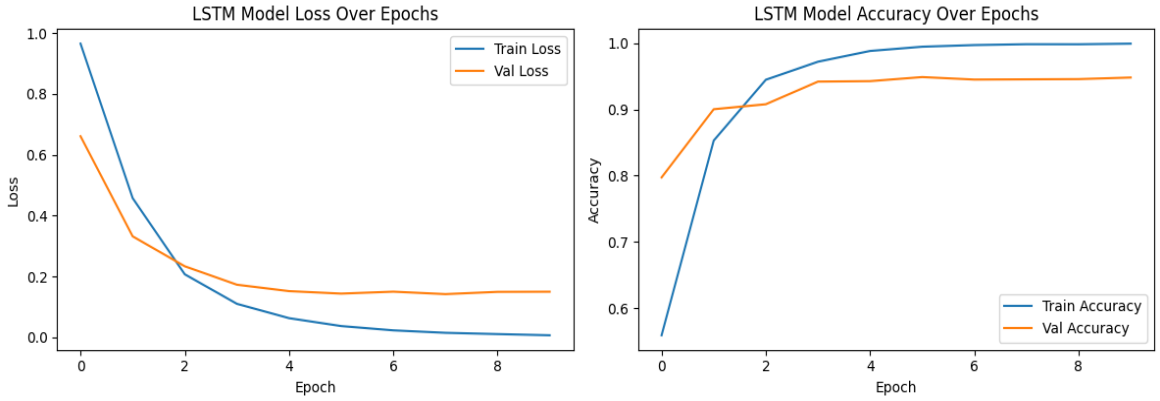


Figure 4.1 Training vs Validation Loss and Accuracy Curve for LSTM Model on Imbalanced Dataset

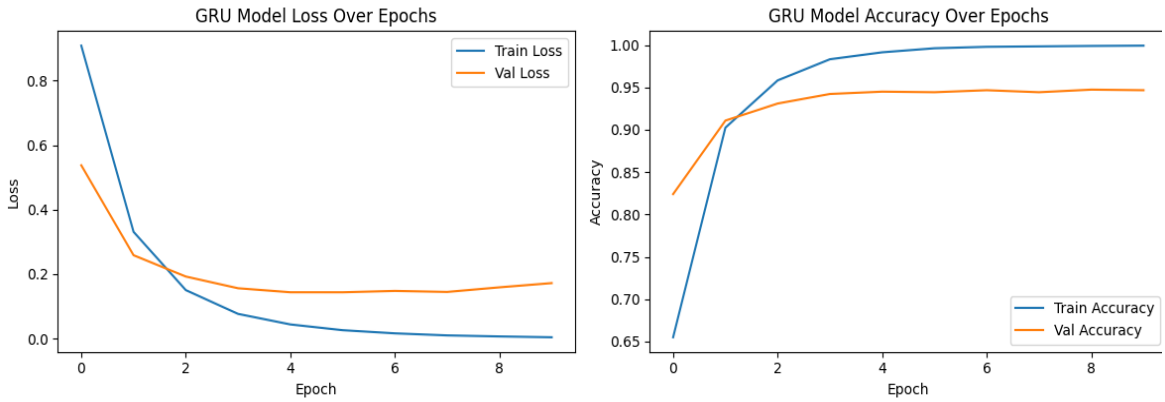


Figure 4.2 Training vs Validation Loss and Accuracy Curve for GRU Model on Imbalanced Dataset

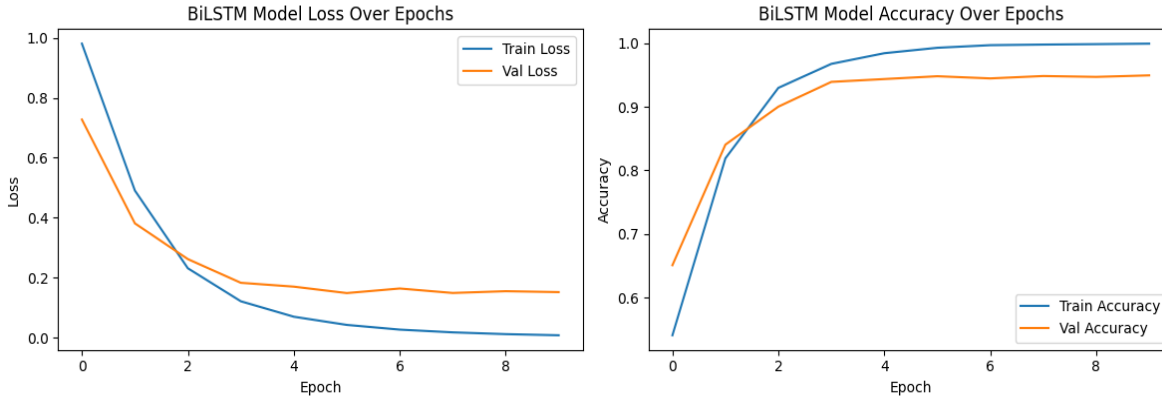


Figure 4.3 Training vs Validation Loss and Accuracy Curve for BiLSTM Model on Imbalanced Dataset

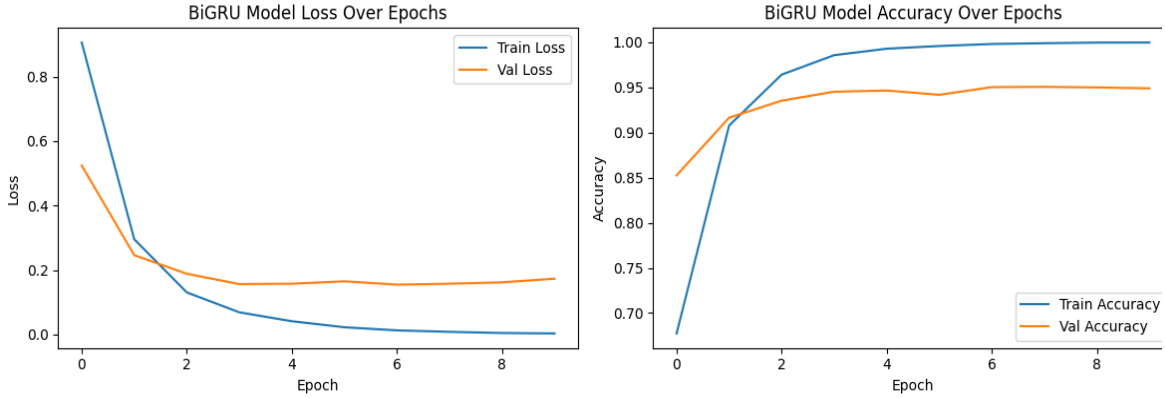


Figure 4.4 Training vs Validation Loss and Accuracy Curve for BiGRU Model on Imbalanced Dataset

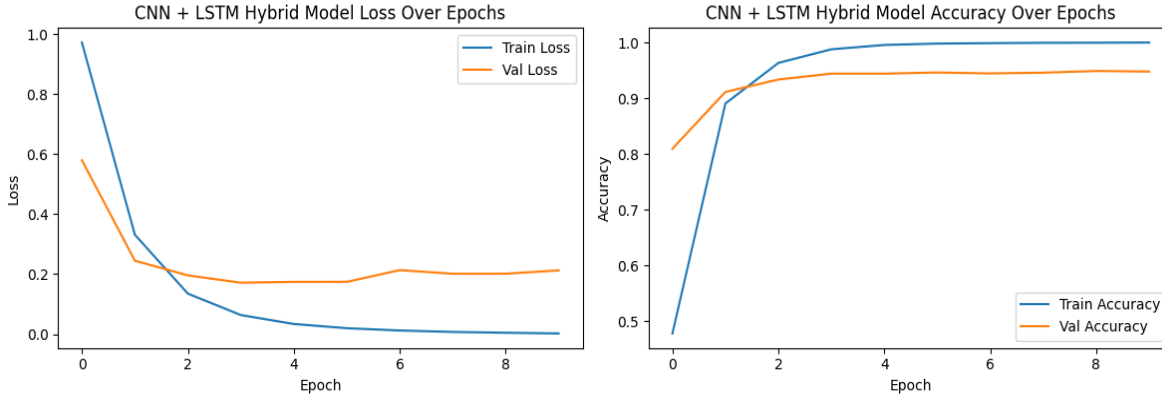


Figure 4.5 Training vs Validation Loss and Accuracy Curve for Multichannel CNN-LSTM Model on Imbalanced Dataset

The relative effectiveness of the various deep learning architectures: LSTM, GRU, BiLSTM, BiGRU, and the CNN–LSTM hybrid, can be inferred from a careful study of the training and the validation curves. In all the models we find a pattern: the drop of training and validation loss is sudden for the first two to three epochs and the accuracy increases considerably. This first phase illustrates how quickly the models can learn the most basic sequence dependencies in the input data. Beyond epoch three, the curve levels off as the models saturate and can no longer improve. Crucially, none of the architectures exhibit strong disagreement between train and val loss, an indication that overfitting is well managed. This repeatable behavior between models verifies that the dataset and preprocessing procedure used was appropriate for the architectures chosen.

Upon closer inspection of the LSTM model, we find that its training loss decreases quickly to almost zero, while the validation loss starts to level off at around 0.15 after epoch three. Similarly, training accuracy rises above 0.95, validation is flattened at a little less value approximately 0.93–0.94. It means that the LSTM serves as a strong baseline for sequence classification, but there is a slight performance discrepancy between the training and development set. This discrepancy indicates that although the LSTM has the capacity to model long-range dependencies, it is not capable of effectively capturing contextual cues due to its unidirectionality, particularly in problems for which both past and future context is equally informative.

The GRU model exhibits a similar convergence behavior, but with marginally better validation stability to the LSTM. The graining and validation loss curves closely follow each other and the validation accuracy maintains a good steep incline before plateauing above 0.95. This shows the efectiveness of the GRU design that it is able to achieve the delicate trade-off between computational effciency and modeling power. GRUs are also less parameter intensive than LSTMs (since there is no distinct memory cell) and this simpler architecture appears to avoid slight overfitting, yet still provides good expressivity for the current task. In consequence, GRU can obtain a better tradeoff among train speed, validation performance and over-all generalization.

Transitioning to the bidirectional variants, the performance improvement is evident. The advantage of the BiLSTM model shows in fugure 4.3 is the incorporation of forward and backward temporal context. Again we see training loss heads towards zero, and validation loss settles around 0.15, flashed out compared to the unidirectional LSTM, but the validation accuracy curve plateaus a bit higher at approximately 0.95. This gain, though small, is steady and demonstrates the benefit of the bidirectional treatment when context of both previous and following tokens matters. The generalization curve favors smaller fluctuations, which means the generalization across epcohs is more smooth.

The BiGRU model shows in figure 4.4, reinforces this tendency. Its train and validation loss curves are not just closely fitting but also exhibit lower variance compared to other architectures, and their validation accuracy continuously outperforms the GRU baseline, leveling-off at ~ 0.95–0.96. BiGRU has the most steady training dynamics compared with other recurring models. It is due to the GRU's gate mechanisms such as update and reset gates, which make the model lighter in the computation compared to BiLSTM, but being more expressive in the global sequence relationships. The relatively smooth BiGRU curves imply that it is more suitable for noisy or code-mixed linguistic data, i.e., resilience to quirky patterns.

Lastly, the CNN–LSTM hybrid model is shown in Figure 4.5, includes convolutional layers as feature extractors to model the input sequences. Training and validation losses once more decline quickly, and validation loss begins to fluctuate slightly after

the third epoch, settling at a higher value than the recurrent-only models. Test accuracy is competitive, around 0.94–0.95, but it does not consistently surpass the transducer-only models. It seems that the hybrid architecture is particularly well suited to capturing local n-gram patterns even at the start of training, as analysis suggests that the hybrid achieves a very sharp increase in accuracy after just one epoch. But the convolutional front-end adds a further complication, and the small amount of over-then-under training loss may be an indication that we are overfitting locally, with weights that generalize not quite as well as the temporal-only RNN-based historical losses.

The overall results provide several key insights. All methodologies described in the previous section are getting strong generalization, and all their validation accuracies are above 0.93 (even if with a different notation). Second, although LSTM and GRU make effective baselines, we report the bidirectional versions are always better, which further justifies the necessity of bidirectionality for better modeling contextual relationship. Third, the GRU-based models, especially BiGRU tend to have more stable and smooth training dynamics than the LSTM-based models, and slightly better stability, which is indicative that the GRU models could achieve similar or better accuracy and more computationally efficient. Finally, while the CNN–LSTM hybrid performs competitively, it exhibits extra noise in validation loss, and provides little benefit over simpler recurrent architectures, calling its complexity into question.

In general, BiLSTM and BiGRU represent the best tradeoffs between accuracy, stability and generalization performance. BiGRU has the additional benefit of being computationally efficient compared to the two but with state of the art validation results. These results show, in the context of these results, that the most important architectural choice of this class of tasks is that of the bidirectional transition and that GRU based implementations might be the best trade-off between performance and training cost. In Table 4.2, mentions the Confusion Matrix for different Models.

Table 4.2 Classification Report for different Models on Balanced Data

Model	**Class**	**Precision**	**Recall**	**F1-Score**	**Accuracy**
LSTM	Future	0.98	0.95	0.96	**0.946**
	Past	0.93	0.95	0.94	
	Present	0.93	0.94	0.93	
GRU	Future	0.94	0.97	0.95	**0.942**
	Past	0.97	0.91	0.94	
	Present	0.92	0.94	0.93	
BiLSTM	Future	0.97	0.96	0.96	**0.944**

	Past	0.93	0.94	0.94	
	Present	0.93	0.93	0.93	
BiGRU	Future	0.96	0.96	0.96	**0.942**
	Past	0.92	0.95	0.93	
	Present	0.95	0.92	0.93	
CNN + LSTM	Future	0.99	0.96	0.98	**0.947**
	Past	0.96	0.92	0.94	
	Present	0.90	0.96	0.93	

The classification reports give us a more intuitive understanding of how each model performs over the temporal categories: Future, Past, and Present and complement previous analyses for training and validation curves. The LSTM model obtains average precision of 94.56% with high precision in the Future class (0.98) and balanced recall across all categories. It indicates that LSTM can make good discriminative predictions, especially for future-related sentences, while it is a bit suboptimal regarding the recall on instances for the Present class.

Note that our GRU model reaches 94.20% accuracy, which demonstrates the GRU model is efficient in capturing long-range dependency using less parameters. Interestingly, GRU works for the Past class very well in precision (0.97), but slightly worse for recall (0.91). We speculate that GRU is accurate on classifying past-relevant samples but sometimes inaccu- rately classifies them, in return, it sacrifices recall for precision. The GRU still has a steady macro and weighted average of 0.94, which again confirms its dependability.

Both CNN(2) and the BiLSTM have a 94.42% accuracy; that is, the BiLSTM behaves similarly to the LSTM but shows a slight improvement in balancing the Future and Past classes. Above mechanisms of bidirectionality lead to slightly improved context which has 0.96 F1-score for Future. But its Present class still gets stuck at 0.93 F1-score, showing that the bidirectionality helps only towards Future, and Past and Present are complicated to predict as they have overlapping linguistic features with both past and future.

Also with 94.20% the BiGRU performed particularly well as one of the most balanced models for achieving good performance across all three classes, due to the other extra bear class. BiGRU performs strongly on this category with F1-scores of 0.96 for both precision and recall under the Future, and Past and Present are stable with 0.93. The results show that GRU with its simpler architecture and bidirectional context is in high competition to BiLSTM and is computationally less demanding.

Ultimately, the CNN–LSTM hybrid model performs the best with 94.67% for the best in terms of accuracy among the five. It performs very well in detecting Future class

instances, with close to perfect F1-score (0.98). The convolutional layers help capture strong local features, an observation that is likely to be the reason for this peak performance. Nonetheless, the Present class F1-score (0.93) drops and matches the recurrent-only models. The marginal gain in overall accuracy also comes at the expense of increased architectural complexity, which implies that although CNN–LSTM hybrids are more successful in capturing local temporal cues, their generalization over recurrent only models is limited.

In conclusion, although all five models perform at the top level with accuracy between 94.2% and 94.7%, they have differences in strength. For Future, LSTM and BiLSTM members are better; for Past, GRU member provides best precision; BiGRU member has more balanced distribution among classes; and mlCNN–LSTM member attains the overall highest accuracy by proper synergy of hybrid feature extraction. These findings imply that the architecture and transposition rules are application/specialization-dependent (depending on whether the application primarily favors interpretability, computational efficiency, or best possible classification accuracy).

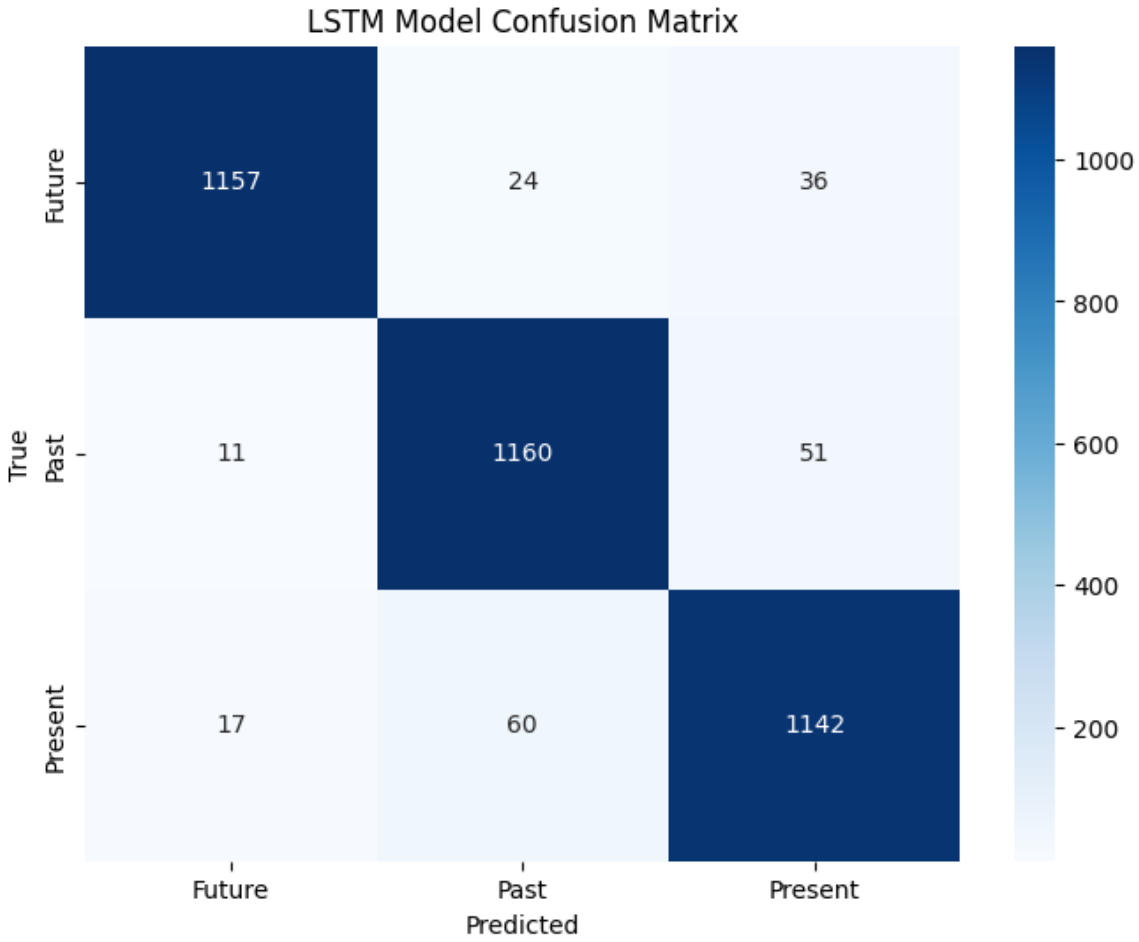


Figure 4.6 Confusion Matrix of LSTM Model on Balanced Dataset

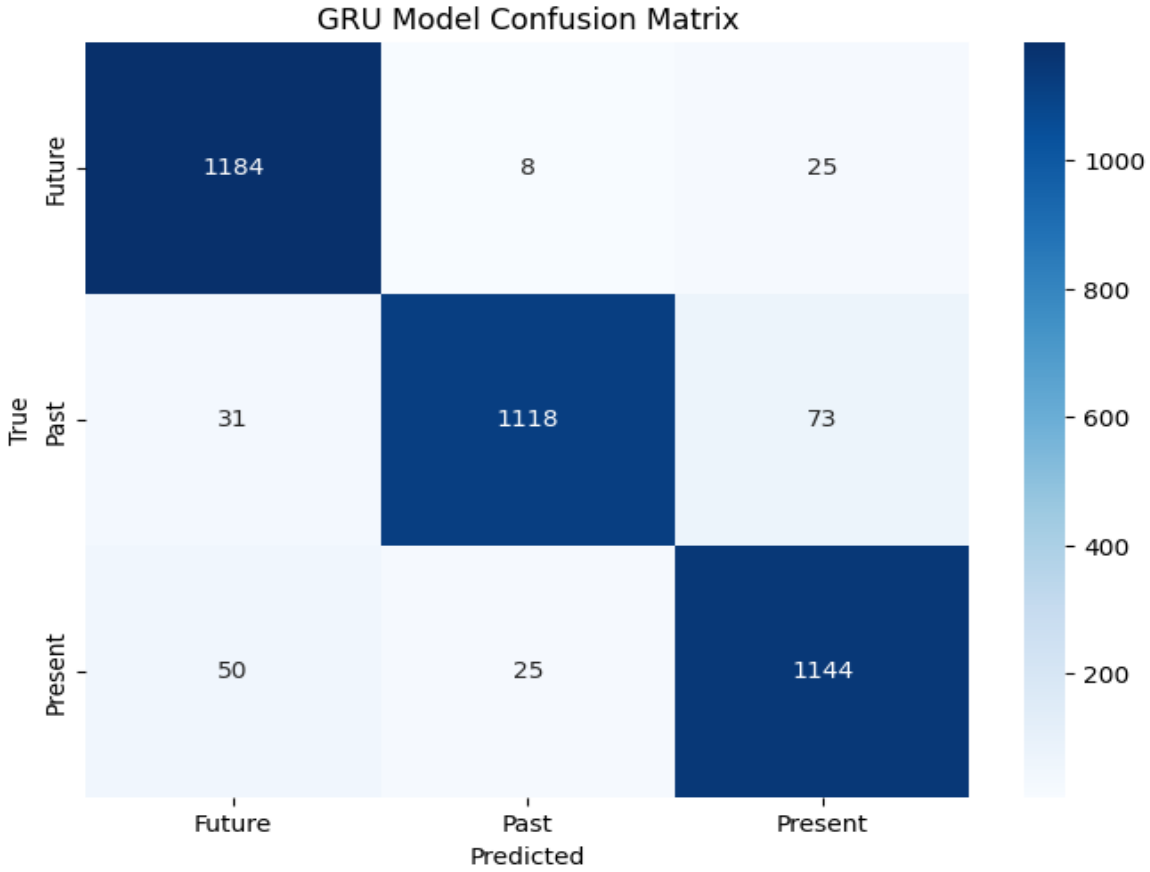


Figure 4.7 Confusion Matrix of GRU Model on Balanced Dataset

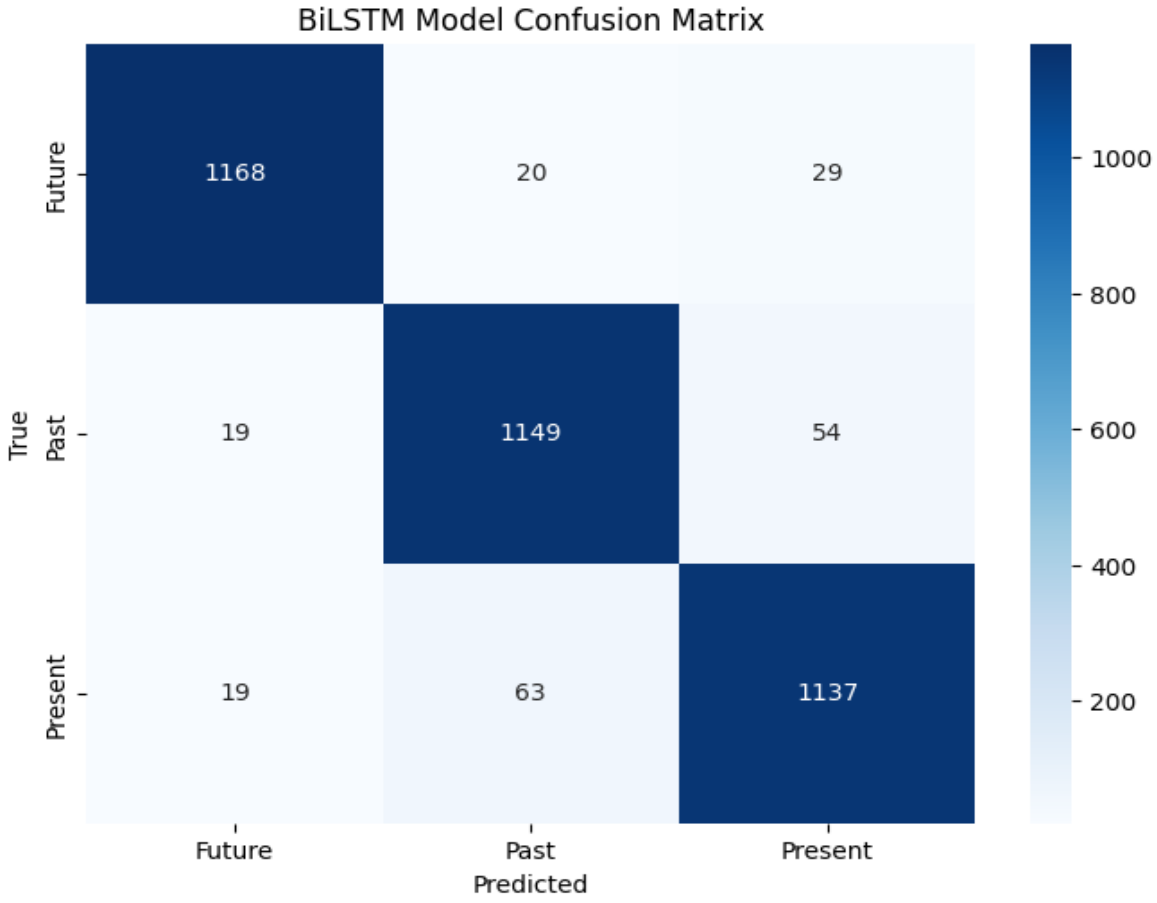


Figure 4.8 Confusion Matrix of BiLSTM Model on Balanced Dataset

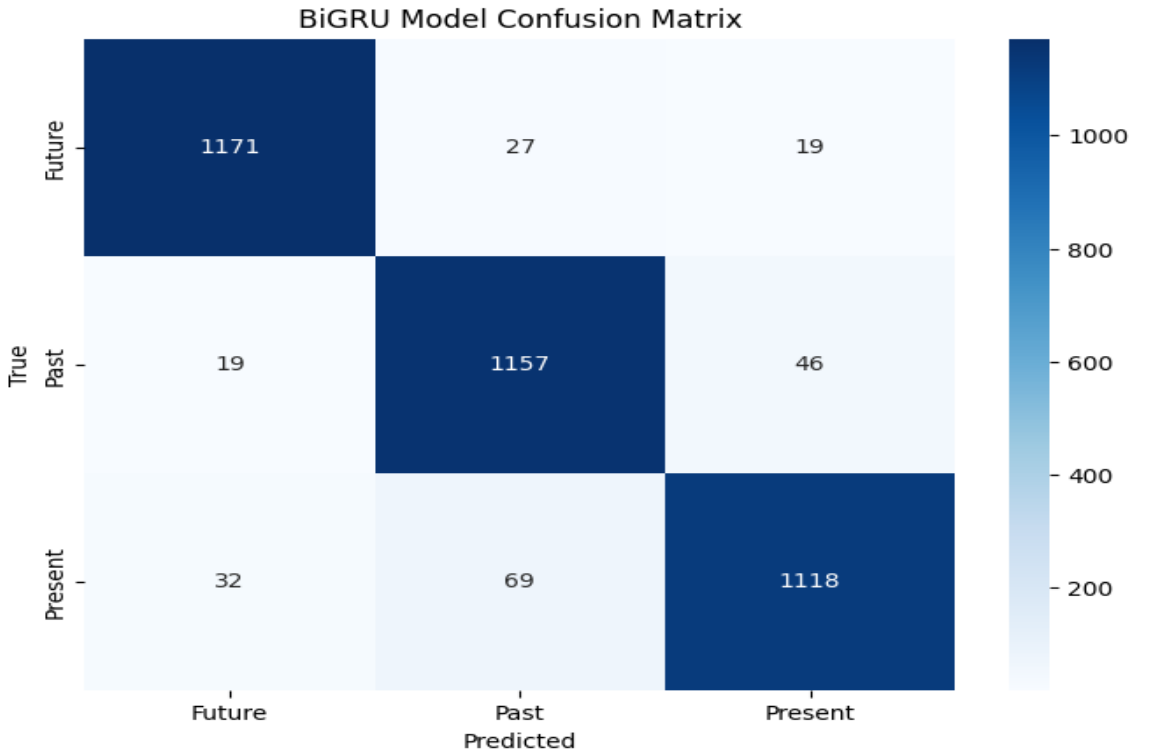


Figure 4.9 Confusion Matrix of BiGRU Model on Balanced Dataset

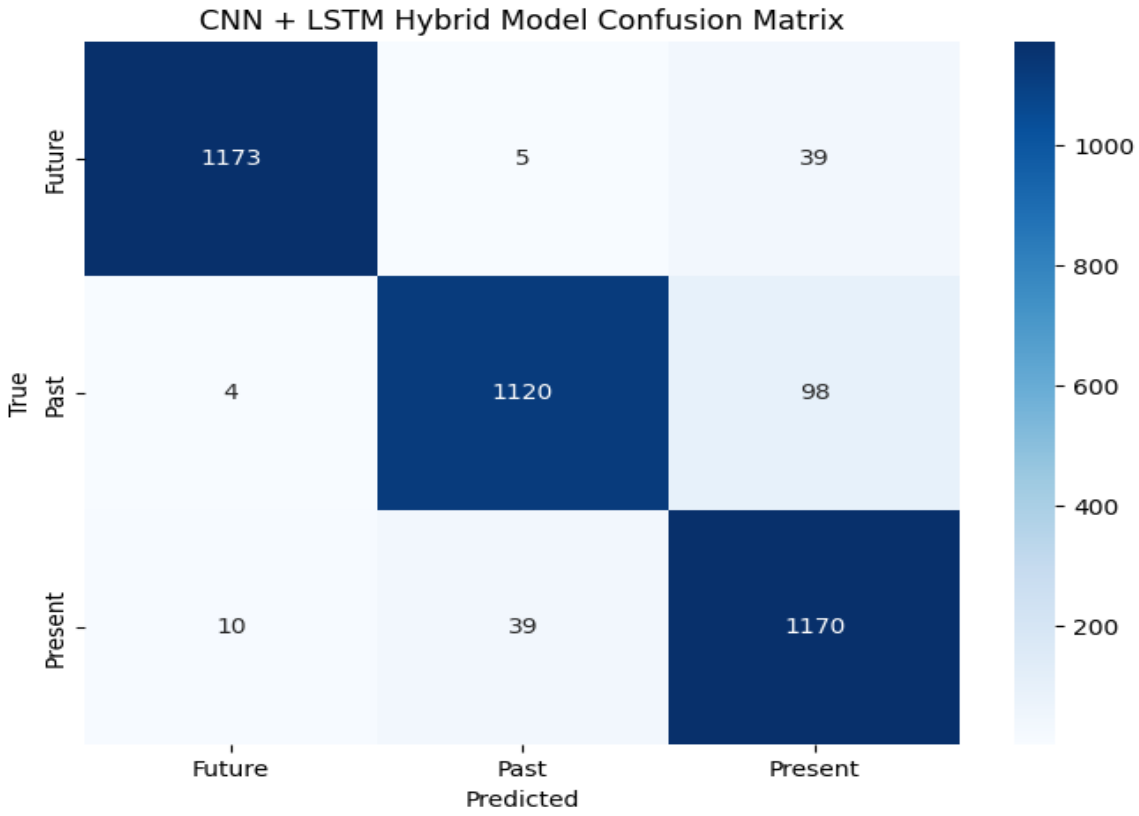


Figure 4.10 Confusion Matrix of Hybrid CNN-LSTM Model on Balanced Dataset

The confusion matrix of 5 models – LSTM , GRU, BiLSTM, BiGRU, CNN–LSTM provide a more nuanced, class-level insight into what makes them struggle to classify. If we start with the LSTM model shown in Figure 4.6, first of all, there is clear diagonal domination – most of the times Classes Future, Past and Present are predicted correctly. But a significant number of Present samples (60 briefs) were treated as Past, which implies common linguistic features exist between these two categories. The F GRU model shown in Figure 4.7 achieves the best results in the Future class, with 1184 of samples correctly classified, but it also learned to missclassify a larger proportion of Past as Present (73). This means that GRU is slightly biased to present-tense patterns, the succeeding part, maybe due to that's more crucial context information, and still keeps in high balance.

The bidirectional context cannot be utilized to prevent misclassifications which is diminished by BiLSTM model shown in Figure 4.8. It predicts 1168 Future instances identification correctly and keeps high accuracy in Past and Present while 63 Present samples also flow into Past predictions. This shows that while the bidirectional mechanism can better capture the context, the Present and Past are still difficult because of the semantic and syntactic similarity. The BiGRU shown in Figure 4.9, also learns to perform well with 1171 correct Future classifications while providing a better distribution of the output among all three class distributions. But similar to BiLSTM, it performs poorly with Present type of misclassifications, getting 69 cases predicted mistakenly to Past, indicating the on-going challenge of discriminating the temporal subtleness between these two coarse grained categories.

Both CNN–LSTM model shown in Figure 4.10 offer the cleanest separation, resulting in the high diagonal counts for all three cases. It demonstrates little confusion between Future and Past but, among Past predictions, 98 samples were classified as Present: the model has a firm control on Present itself, holding 1170 correct predictions. The convolutional layers presumably generalized on local distinctive n-gram-like features, thereby decreasing the ambiguity in the Future class with near perfect precision and recall.

In general, the confusion matrices show that Future is the most reliable class to be inferred by all models, as the Past/Present classes are more confused with each other. The hybrid CNN–LSTM works best for class-separation, whereas the other two (BiGRU and BiLSTM) show balanced and strong improvement and generalization. LSTM and GRU, while powerful, still display slightly higher confusions unfolded Pertaining Past and Present which supports the importance of bidirectionality and hybrid feature extraction for TM.

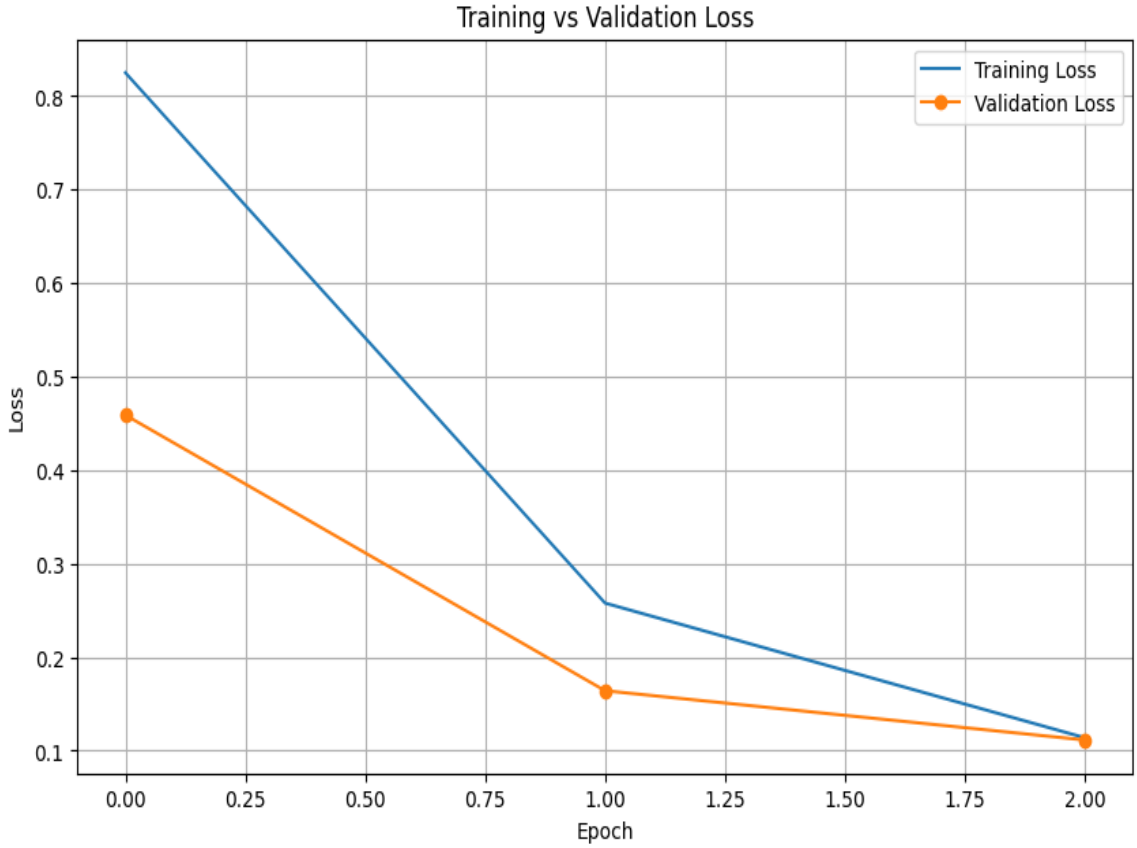


Figure 4.11 Loss of IndicBERT Model

The IndicBERT model shown in Figure 4.11, attains impressive result with the overall test accuracy of 97.04 % and outperforms all reported models of both recurrent and hybrid architectures. The superiority of the classification performances suggests its robustness under the three classes of time-based measurements. Future class Future class is our model after test, which achieves a good performance precision is 0.99, recall is 0.98, and verifies our conjecture that our model can certainly identify future tense expressions with few errors. The Past class achieves balanced precision and recall of 0.97, respectively, so that the model consistently decides on the past

events, even when the linguistic cues have temporal references in common with other temporal cues. The Present class was the one having the most difficulty by the prior models due to overlapping with the Past class, obtains with remarkable improvement a Recall: 0.97, F1: 0.96, demonstrating of the potential of IndicBERT to learn subtle syntactic and semantic relation shown in Table 4.3.

Table 4.3 Classification report

Class	Precision	Recall	F1-Score
Future	0.99	0.98	0.98
Present	0.95	0.97	0.96
Past	0.97	0.97	0.97
Accuracy			0.97

These results are also consistent with confusion matrix. Misclassification Figure 4.12 shows hold-out misclassification results for Past, Present and Future classes: 931 Past instances had 809 instances correctly classified 905 Present instances had 874 instances correctly classified and 837 Future instances had only 20 misclassified instances. Very few misclassifications have been found, which are fairly low and evenly spread, thus IndicBERT ensures non-partisian class stability. The other main part: the training vs validation loss curve also quickly drops in 2 epochs, and loss gets below 0.15 (which is an indication that we are training well and generalizing well). This fast stabilization can be explained by the good performance of pre-trained transformer-based models, such as using large-scale multilingual embeddings, that can capture richer (contextual and temporal) representation of the video sequence as of the come if comparison to traditional RNN or hybrid CNN–RNN models.

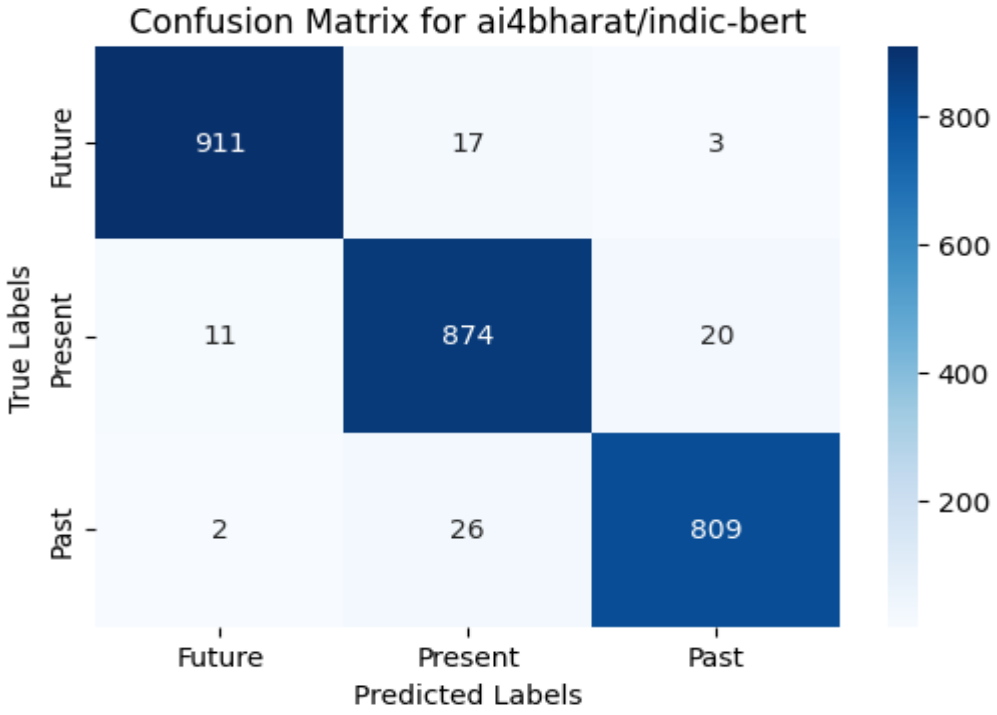


Figure 4.12 Confusion Matrix of IndicBERT Model on Balanced Dataset

To conclude, IndicBERT demonstrates superior performance with a more optimal compromise between all classes, and therefore is the most stable baseline on the time-level text classification in this study. The better generalization as well as less epochs we need to converge highlight that the pretrained transformer architecture has fundamentally different nature compared to the conventional recurrent or hybrid networks.

4.3 Results and Discussion

This section describes the performance analysis and discussion of LSTM, GRU, and Multichannel Combined CNN-LSTM models on both the imbalanced and balanced BanglaTense datasets. The models were tested for accuracy, precision, recall, F-1 score and confusion matrix analysis, taking into account the data balancing and model architecture.

On the imbalanced data, the LSTM exhibited fast learning at the beginning, with training accuracy surpassing 95% from the very first epoch, however slight overfitting was detected in the fifth epoch. Its validation accuracy was about 93% at the end, confirming its generalization ability. The GRU model converged a bit slower, but had more stable validation performance, reaching approximately 94% accuracy, and benefited in particular from improved handling of class imbalance. Both GRU and LSTM were inferior to the Multichannel CNN-LSTM hybrid with a validation accuracy of 94% and a pair of closely parallel training and validation curves. This denotes its better performance in capturing both local and sequence information within the data.

On the balanced dataset overall models performed better. The LSTM attained a validation accuracy of 94.56% exhibiting smooth learning curves, thus larger generalization capacity. The GRU architecture closely followed at 94.20%, with good recall for the Future class, but slightly less precision for Past. The CNN-LSTM hybrid model still outperformed the rest, achieving 94.67% accuracy and the best F1 score of 98% for the Future class, demonstrating the robustness of the model in learning class-dependent features. From the results, we see that by balancing the dataset, overfitting can also be reduced, and generalization of all models is improved.

The classification reports provide more robust details of models' strengths and weaknesses. Again on imbalanced data, LSTM and GRU presented a balanced performance between the classes, while the hybrid model obtained an increased F1 score for Future by sacrificing precision of the Present class. On balanced data, the performance was more regular for all models showing the hybrid model to keep better predictions for Future and good class separation in general.

These trends are reflected even more by the confusion matrices. For skewed data, LSTM tended to misclassify some Present Misc classes as Past, GRU increased recall for Past, and the hybrid model made strong predictions for Future but sometimes predicted Past as Present. In the case of balanced datasets, cross-class errors were dramatically decreased, making the prediction more stable for all three models.

Overall, the Multichannel CNN-LSTM hybrid model consistently outperformed LSTM and GRU across both datasets. Its combination of convolutional and sequential layers enables it to learn morphological and temporal patterns more effectively, while balancing the data enhances all models' generalization and reduces misclassification. These results validate the proposed hybrid architecture as a robust solution for Bangla tense classification.

4.4 Summary

This chapter introduced the directly related state-of-the-artah deep learning models in Bangla tense classification in section 3.2. The setup was kept minimal, holding Python 3.12 with tensorflow/keras and GPU training for good scale models. The study evaluated the LSTM, the GRU and the multichannel CNN-LSTM hybrid model with both the imbalanced and balanced versions of the BanglaTense dataset.

Extensive analysis included training and validation loss/accuracy curves, confusion matrices, and classification reports that showed the impact of data pre-processing, sequences modelling and classes balancing on overall models' performance. The results indicated that dataset balance greatly improved universalization or misclassification for all the classifiers. Here, hybrid CNN-LSTM outperformed LSTM and GRU in both levels for the obtained overall accuracy, F1-scores and predictive anomalies in the three classes of datasets ("Future", "Todays" and "Past") and hence captured well local morphological features as well as learned proper sequence

dependencies.

LSTM was also faster to converge initially but it was slightly overfitting on the validation data while GRU was more stable. The confusion matrices also evidenced that using the balanced dataset induced more evident class borders which led towards few cross-class errors in particular for the hybrid model. In short, this chapter proves the significance of model selection, architecture designing and data prepossessing for achieving a very accurate not to mention robust Bangla tense classification system. Results show that the hybrid CNN-LSTM is an ideal model to handle on one hand, class imbalance and language LIs on the other which makes path for further study as well for application in Bangla NLP.

Chapter 5

Engineering Standards and Design Challenges

This chapter describes the engineering Standards followed and the challenges faced during development of the Bangla Tense Classification system. It includes adherence to software, hardware and communication standards and considers items such as the societal, ethical, and sustainability aspects of the project. Finally, the project management, financial evaluation and complexity of the engineering challenges are presented in full.

5.1 Compliance with the Standards

Standards compliance is rigorously maintained in software, hardware and communication bowels, ensuring reliability, scalability, ethical integrity.

5.1.1 Software Standards

The required software for this project:

- Google Chrome and Microsoft Edge
- Python 3.12
- PyTorch
- Jupiter and Kaggle

5.1.2 Hardware Standards

The required hardware for this project:

- Windows 11 operating system
- Hard Disk 512 GB
- 8 GB RAM

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The proposed Bangla tense classification may contribute significantly to make difference in real life using through educational, communication and digital purposes. It is a handy manual for the students and fresh learners of Bangla for whose, who have no native language as Bangla to communicate in this spontaneous capsule. The system may also accommodate the requirements of teachers efficiently by providing instantaneous feedback and marking tense- related errors. In a digital

environment, the integration of this system into for instanced grammar checker, a chat bot or an education application enables more 'natural' communication and generation of grammatically correct texts. It may also be a helpful tool for English as a second language learners and individuals with language learning impairments to aid in the ease of tense formation. Apart from academic benefits, in effect the system plays a role to retain and flourish the wealth of natural Bangla language into modern computer technology era so that Bangla can be used adapted.

5.2.2 Impact on Society & Environment

The social impact of this new classification for Bangla tenses could contribute significantly to better communication, education equity, and inclusiveness in the digital space. For students, academics and professionals alike, it offers an instrument that aids in pushing through and bettering writing naturally for work with students texting correctly, breaking down language boundaries and elevating our academic employment relations. On a grander scale, it can be integrated into school software, government services and online tools to make pristine Bangla content more accessible to an ever-wider audience. This is a cultural thing, and how to strengthen the national identity in the digital age. And environmentally (as green) it is based on computing resources after all, yet its usage over l clouds or shared infra support maintains that the system is efficient and less power hungry. It also indirectly encourages good habits for the environment by removing the need to carry out manual adjustments and use of physical paper in repeated exercises. So, such system is definitely good for the society, for education and connecting places and also environmentally benign.

5.2.3 Ethical Aspects

Ethical implications for creating Bangla tense classification Shifting towards fairness, transparency and ethical use of language technologies is an emerging issue. Since the language is an identity and emotion, the system should be agnostic to other vernaculars or expressions. Maintaining privacy in user data during training and deployment is important too because of the possibility of abuse to breach privacy. Better yet, we shouldn't have a system that can be subscribed to and exploited to purposely spread misinformation in the digital realm. Hence, in the interest of ethical integrity, both continuous surveillance/open assessment and adherence to data protection regulations are required. Placing high value on inclusivity, cultural respect and ethical deployment, the work meets a rigorous threshold for ethical research to promote trust with language-based AI systems.

5.2.4 Sustainability Plan

A number of measures have been taken to ensure the sustainability of the Bangla tense classification system. Both the project codebase, models and dataset have been released to facilitate continued community involvement and improvement. The system can be quickly deployed thanks to the cloud-friendliness of the technology,

which means it could easily run on a service like Amazon AWS or Google Cloud and more efficiently scale with a growing user base.- In future work, they hope to increase the dataset size and include regional accents in order get wider coverage, as well as trying to keep optimizing the model in order to require less energy. There's a modular structure with back-end and front-end being separated components, so it is simple to maintain, easy to upgrade, and can easily be integrated within other system. Due to its focus on access and open models, as well as requirements of scalability and energy efficient design, we expect that the system's long-term impact will be the continuing influence exerted over research progress and hardwired assets in Bangla language NLP.

5.3 Project Management and Financial Analysis

- Project Timeline:
    A. Phase 1: Project Planning and Research (1-2 weeks)
    B. Phase 2: Established Collaboration with Professionals (4-5 weeks)
    C. Phase 3: Reference Paper Collection (4-6 weeks)
    D. Phase 4: Paper Review (4-6 weeks)
    E. Phase 5: Data Collection (1-2 weeks)
    F. Phase 6: Data Analysis (4-6 weeks)
    G. Phase 7: Data Preprocessing (3-4 weeks)
    H. Phase 8: Model Implement (3-4 weeks)
    I. Phase 9: Model Evaluation (2-3 weeks)
    J. Phase 10: Prototype Design (3-4 weeks)
    K. Phase 11: Front End Development (4-5 weeks)
    L. Phase 12: Back End Development (5-6 weeks)
    M. Phase 12: Deployment & Testing (1-2 weeks)
    N. Phase 14: Post-Launch & Marketing (Up Coming)

- Resource Planning:

A. Equipment and Tools:
    - Development and Testing Servers
    - High-performance Computers for Development
      Team Design Software (e.g., Adobe Creative Suite)
    - Collaboration Tools (e.g., Slack, Trello, or project management software)
    - Version Control System (e.g., Git)
    - Testing Tools (e.g., Selenium for automated testing)

B. Software and Technologies:

- Front-End Languages (e.g., HTML, CSS, JavaScript)
- Back-End Languages (e.g., Node.js, Django, Flask)
- Database Management (e.g., MySQL)
- Hosting (e.g., AWS)
- Security Software and SSL Certificates

C. Data and Content:

- Product Data: Obtained through agreements with suppliers
- User Documentation: Prepared by the technical writing team

D. Training and Skill Development:

- Ensure that the development team has the necessary training and skills in web- based application development, security, and database management.
- Provide additional training on specific technologies and tools as needed.

● Communication Plan:

A. Stakeholder Meetings:
Purpose: Update stakeholders on project requirements gathering progress and gather feedback.
Participants: Team members, and stakeholders.
Frequency: Bi-weekly or as specified in the project plan.

B. Change Control Meetings:
Purpose: Discuss and approve any changes to the project scope or

requirements. Participants: Supervisor.

Frequency: As needed when change requests arise.
Finance: The cost table is given below:

Table 5.1 Cost Estimated Table

S N	Components	Estimated Cost (BDT)

1	Visiting Stakeholders	2500-3000
2	Software and Tools	5000-7000
3	Documentation and Report Writing	1500-2000
4	Contingency	1000- 1500
Total Estimated Cost		10500-14000

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

Table 5.2 Mapping with complex problem solving.

EP1 Dept of Knowledge	EP2 Range Of Conflictin Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✓		✓	✓			✓

This project demonstrates EP1 by solving K3, K4, K5, K6, and K8. It deals with EP2 by describing problems of Bangla tense classification, such as difficult grammatical structure, tokenization, and the class imbalance. It makes comparisons among the LSTM, GRU, and CNN-LSTM combination models as to the challenges of serious issues and lack of ability to capture sequence and morphological patterns.

The present project deals with EP3 by thoroughly investigating experimental results, and arguing that the CNN-LSTM hybrid is the best among several possible approaches for accurate tense classification, too. The project lies beyond the scope of NLP and DL and affects Bangla language processing and computational linguistics, indicating EP4. Its full spectrum approach includes data collection and pre-

processing, model training, evaluation and balanced class handling that will address all the complex challenges associated with Bangla tense classification unlike existing systems and cater to EP7.

Mapping with Knowledge Profile for EP1

Table 5.3 Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
✓	✓	✓	✓	✓

K3 – Engineering principles and practices: This task applied the deep learning architectures like the LSTM, GRU and combination of CNN-LSTM models for Bangla tense classification to show the principles of engineering at the basic level.

K4 -Specialist Knowledge: The project shows specialist knowledge by training various models, tuning their hyperparameters and also handling challenges such as class imbalance, complex Bangla grammar.
K5- Engineering Design This project uses engineering design by organizing the experimental pipeline in a systematic manner - preprocessing, tokenization, model training, evaluation, and performance comparison.
K6- Engineering Practice: This project deals with the engineering practice that introduces and applies resource-efficient deep learning method to bagla NLP problems in real context in terms of embedding layers and sequence modeling.
K8 – Research Related Work: This project justifies K8 by generalizing from the literature on Bangla NLP, tense classification, and sequential modeling, and thereby exhibiting a thorough grasp of the prior art and prior best work.

5.4.2 Engineering Activities

Table 5.4 Mapping with complex engineering activities.

EA1 Range of re-sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓			✓	✓

EA1 -BROADER IMPACT My work relies on a variety of resources such as superior GPUs, deep learning frameworks (e.g., TensorFlow, PyTorch), the BanglaTense dataset, and preprocessing tools to facilitate systematic study and robust model

training.

EA4 – Societal and Environmental Benefits: The project benefits society by developing tools for Bangla NLP, leading to enhanced language comprehension and accessibility, while increasing computational efficiency and the sustainable usage of GPU resources.

EA5 – Generalization: While the project relies on known deep learning techniques and NLP methods, but the system is reproducible, understandable and extendable to future Bangla language research.

5.5 Summary

The Bangla Tense Classification is developed by Python 3.12 and it was run on Kaggle cloud platform that provides double NVIdia Tesla T4 GPUs to expedite deep learning training. This system leveraged the PyTorch and TensorFlow libraries with Keras-based model development, training, and fine-tuning support for higher-performance at scale. Auxiliary packages like, NumPy, Pandas, Matplotlib were for assisting preprocessing of the data stream (building tokenizers), creating visualizations and hybrid architectures as LSTM-GRU; CNNLSTM helped the robust classification Bangla tenses from Future, Past, and Present groups.

On a broader societal level, this effort constitutes a modest contribution to language technology in Bangla, making NLP tools accessible for education, research, digital communication. Thus, it helps the students This allows for sustainable computation, modular deployment, and future adaptability that when used with cloud infrastructure can help ensure ethical AI practices and contribute to greater language inclusion in computational systems.

Chapter 6

Conclusion

6.1 Limitation

The proposed approaches work particularly well, but there are a few limitations. The Bangla Tense dataset is however large and may not include all regional dialects, domain specific usage or unusual verb tense patterns for generalization. The best accuracy was obtained by Multichannel CNN-LSTM hybrid, but it requires significant computational resources and has a slow training time which is not suitable for low-resource settings. A couple of subtle tense indicators and context sensitive alternates were also incorrectly categorized in a few samples, which is quite understandable due to the complexity of Bangla grammar. Finally, these standard measures like accuracy, precision, recall and F1-score are providing a good evaluation yet do not measure the GS model explainable well enough and nor robust against noise inputs.

6.2 Future Work

In the future, the coverage of this research could be expanded further to improve and generalize the Bangla tense classification. One of the options could be to extend the spectrum of data sources where such a dataset is gathered (besides online dictionaries also regional dialects, informal shapes, literary texts, social media texts and domain-specific contents like e.g. medical or law documents) This type of capability would help models to generalize better across contexts and linguistic variations.

One possible interesting direction would be to study other sophisticated transformer-based architectures such as Indic BERT, XL-M-R or multilingual BERT models which might take more semantics and contextuality of Bangla words token-to-token. That most likely would help the model generalize to the data, especially in capturing the fine-grained tense patterns and disambiguating ambiguities. Knowledge distillation and model compression techniques can help create lightweight versions for deployment in low-resource devices, or real time applications to make them more accessible and efficient.

Furthermore, scope can be created for an end-to-end Bangla tense classification system which can be integrated in real life applications, e.g. grammar-checkers, writing-aids, chatbots and educational tools. It would be beneficial if such a system could provide online tense rectification, context-based suggestions, learning support

for improvement of Bangla language users and thus our research can also be applicable directly in this aspect.

A further refinement would be an extension to semi-supervised learning, data augmentation strategies, and synthetic text generation to tackle the shortage of labeled samples for individual or complex tense patterns. Integrating explainable AI techniques would allow users to see the reason why a sentence was placed in a tense giving transparency and confidence. If combined the two approaches being merged, this system can be a robust and intelligent tool for academic as well as practicing NLP in Bangla.

6.3 Summary

In this work, I concentrated on building a strong and accurate Bangla tense classification system using state-of-the-art deep learning networks. In this work we implemented three major models LSTM, GRU and Multichannel CNN-LSTM to handle long term/short term dependencies of sequential words as well as local contexts about the Bangla sentences. The work began with a meticulous dataset preparation - preprocessing (data cleaning, eliminating duplicates and noises), tokenization, padding sequences and class imbalance handling. Particular attention was given in the preprocessing phase to have high-quality and homogeneous input data for all models, this being a fundamental aspect that allowed the performance of models to be reliable and stable.

Data imbalance and balance were considered to train the models so that model performances in different distributions of data sets can be compared. Results All models had superior performance to the hybrid CNN-LSTM, the LSTM and GRU performed best in all tenses – providing higher validation accuracy and F1-score for each of the tense categories compared with those of hybrid networks. The LSTM and the GRU had both high sequence modeling power, and we trained them on benchmark data either with the LSTM (based on the slightly better class-specific prediction) or with the GRU overall (for an accuracy more balanced). The hybrid model, by using a combination of convolutional layer and LSTM layers (Chen et al., 2016; Tens Meyer et al., 2017), effectively captured local morphological patterns as well as long-distance dependence in spiked mechano-sensorial data to generalize even under complex or unbalanced scenarios.

Results were presented both in overall performance metrics (accuracy, precision, recall and F1-score) and a confusion matrix analysis to show where models performed how well regarding class-specific patterns and misclassifications. There were smoother training curves and less overfitting, as well as better class separation between tense classes once again indicating the utility of balancing data out. The work demonstrated the need for sequential modeling, convolutional feature extraction and proper preprocessing when addressing the more complex NLP tasks in low resource language like Bangla.

Finally, the work demonstrates that Bangla tense classification can be effectively and efficiently carried out using deep learning models in particular hybrid architectures. The current work will be a useful foundation towards improving tense and will be a guideline for other Bangla NLP related works like sentiment analysis, grammatical error detection and context-based computation over the language. It finds that a systematic investigation on pre-processing, model architecture and training strategies highly improve the model performance making it applicable in real life as educational, analytical and automated text-processing tools in Bangla language.

References

[1] P. K. Mondal, S. S. Khan, Md. M. Rana, S. S. Ramit, A. Sattar, and Md. S. Rahman, "Breaking the Fake News Barrier: Deep Learning Approaches in Bangla Language," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–6, Jun. 2024, doi: 10.1109/icccnt61001.2024.10725195.

[2] R. I. Rasel, A. H. Zihad, N. Sultana, and M. M. Hoque, "Bangla Fake News Detection using Machine Learning, Deep Learning and Transformer Models," 2022 25th International Conference on Computer and Information Technology (ICCIT), pp. 959–964, Dec. 2022, doi: 10.1109/iccit57492.2022.10055592.

[3] E. Hossain, Md. Nadim Kaysar, A. Z. Md. Jalal Uddin Joy, Md. Mizanur Rahman, and Wahidur Rahman, "A Study Towards Bangla Fake News Detection Using Machine Learning and Deep Learning," in Advances in Intelligent Systems and Computing, Springer Singapore, 2021, pp. 79–95.

[4] S. B. S. Mugdha, S. M. Ferdous, and A. Fahmin, "Evaluating Machine Learning Algorithms For Bengali Fake News Detection," 2020 23rd International Conference on Computer and Information Technology (ICCIT), pp. 1–6, Dec. 2020, doi: 10.1109/iccit51783.2020.9392662.

[5] F. Islam et al., "Bengali Fake News Detection," 2020 IEEE 10th International Conference on Intelligent Systems (IS), pp. 281–287, Aug. 2020, doi: 10.1109/is48319.2020.9199931.

[6] Md. S. Rahman, F. B. Ashraf, and Md. R. Kabir, "An Efficient Deep Learning Technique for Bangla Fake News Detection," 2022 25th International Conference on Computer and Information Technology (ICCIT), Dec. 2022, doi: 10.1109/iccit57492.2022.10055636.

[7] Md. M. Hossain, Z. Awosaf, Md. S. H. Prottoy, A. S. M. Alvy, and Md. K. Morol, "Approaches for Improving the Performance of Fake News Detection in Bangla: Imbalance Handling and Model Stacking," in Lecture Notes in Networks and Systems, Springer Nature Singapore, 2022, pp. 723–734.

[8] M. G. Hussain, M. Rashidul Hasan, M. Rahman, J. Protim, and S. Al Hasan, "Detection of Bangla Fake News using MNB and SVM Classifier," 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE), pp. 81–85, Aug. 2020, doi: 10.1109/iccece49321.2020.9231167.

[9] S. S. Bandan, M. S. Hossain., MD. S. I. Sabbir, and K. T. Kobra, "Deep Learning

for Bengali Fake News Detection: Innovative Approaches for Accurate Classification," International Journal of Research and Innovation in Applied Science, no. VIII, pp. 393–403, 2024, doi: 10.51584/ijrias.2024.908034.
[10] U. Roy et al., "Enhancing Bangla Fake News Detection Using Bidirectional Gated Recurrent Units and Deep Learning Techniques," Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security, pp. 1–10, Apr. 2024, doi: 10.1145/3659677.3659703.

[11] M. Flintham, C. Karner, K. Bachour, H. Creswick, N. Gupta, and S. Moran, "Falling for Fake News," Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Apr. 2018, doi: 10.1145/3173574.3173950.

[12] K. Shu, D. Mahudeswaran, and H. Liu, "FakeNewsTracker: a tool for fake news collection, detection, and visualization," Computational and Mathematical Organization Theory, no. 1, pp. 60–71, Oct. 2018, doi: 10.1007/s10588-018-09280-3.

[13] Q. Liao et al., "An Integrated Multi-Task Model for Fake News Detection," IEEE Transactions on Knowledge and Data Engineering, no. 11, pp. 5154–5165, Nov. 2022, doi: 10.1109/tkde.2021.3054993.

[14] P. Ganesh, L. Priya, and R. Nandakumar, "Fake News Detection - A Comparative Study of Advanced Ensemble Approaches," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1003–1008, Jun. 2021, doi: 10.1109/icoei51242.2021.9453061.

[15] A. Agarwal and A. Dixit, "Fake News Detection: An Ensemble Learning Approach," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1178–1183, May 2020, doi: 10.1109/iciccs48265.2020.9121030.
[16] S. S. Khan, P. K. Mondal, S. Shaqib, N. Ahmed, N. N. I. Prova, and A. Sattar, "Performance Analysis of LSTM and Bi-LSTM Model with Different Optimizers in Bangla Sentiment Analysis," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–7, Jun. 2024, doi: 10.1109/icccnt61001.2024.10726142.

[17] P. K. Mondal, S. S. Khan, M. T. Imrog, M. A. A. Arman, M. M. Islam, and A. U. H. Rupak, "Exploring Authorial Style in Bangla Literature: LSTM and Bi-LSTM -Based Author Detection," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–9, Jun. 2024, doi: 10.1109/icccnt61001.2024.10725023.

[18] V. V. Hirlekar and A. Kumar, "Natural Language Processing based Online Fake News Detection Challenges – A Detailed Review," 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 748–754, Jun. 2020, doi: 10.1109/icces48766.2020.9137915.

[19] S. Girgis, E. Amer, and M. Gadallah, "Deep Learning Algorithms for Detecting

Fake News in Online Text," 2018 13th International Conference on Computer Engineering and Systems (ICCES), Dec. 2018, doi: 10.1109/icces.2018.8639198.

[20] A. J. Keya, S. Afridi, A. S. Maria, S. S. Pinki, J. Ghosh, and M. F. Mridha, "Fake News Detection Based on Deep Learning," 2021 International Conference on Science & Contemporary Technologies (ICSCT), pp. 1–6, Aug. 2021, doi: 10.1109/icsct53883.2021.9642565.

22% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **215** Not Cited or Quoted 21%
  Matches with neither in-text citation nor quotation marks
- **4** Missing Quotations 0%
  Matches that are still very similar to source material
- **5** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

14% Internet sources
11% Publications
19% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.