DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PORT CITY INTERNATIONAL UNIVERSITY

Detecting and Explaining Bangla Idiom Misuse: A Comparative Study of Neural Approaches with Reason Generation

Submitted by

Md. Ashiful Hoque Chowdhury Tanbin

ID: CSE 02707396

Department of Computer Science & Engineering

Port City International University

This thesis is submitted to the Department of Computer Science and Engineering of Port City International University in fulfillment of the requirement for the degree of Bachelor of Science, Fall 2025.

February 2026

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PORT CITY INTERNATIONAL UNIVERSITY

Detecting and Explaining Bangla Idiom Misuse: A Comparative Study of Neural Approaches with Reason Generation

Submitted by

Md. Ashiful Hoque Chowdhury Tanbin

ID: CSE 02707396

Department of Computer Science & Engineering

Port City International University

Supervised by

Mrs. Farzina Akther

Assistant Professor

Department of Computer Science & Engineering

Port City International University

This thesis is submitted to the Department of Computer Science and Engineering of Port City International University in fulfillment of the requirement for the degree of Bachelor of Science, Fall 2025.

February 2026

DECLARATION

It’s hereby declared that this thesis has not been submitted elsewhere for the award of any degree to the best knowledge of the author’s reference and acknowledgement to other researchers has been given as appropriate. I also confirm that I have just used the resources that have been indicated. All formulations and concepts borrowed directly or in substance from printed or not printed material or the Internet have been cited according to proper scientific procedure with footnotes or other precise references to the original source. I am aware that providing inaccurate information may result in legal ramifications

(Signature of Student)

Md. Ashiful Hoque Chowdhury Tanbin

CSE 02707396

Department of Computer Science & Engineering

Port City International University

RECOMMENDATION

This is to certify that this thesis report entitled "**Detecting and Explaining Bangla Idiom Misuse: A Comparative Study of Neural Approaches with Reason Generation**" submitted by **Md. Ashiful Hoque Chowdhury Tanbin - CSE 02707396** in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Department of Computer Science & Engineering of Port City International University is a record of the candidate own work carried out by him under my supervision. This thesis has not been submitted for the award of any other degree.

(Signature of Supervisor)

Mrs. Farzina Akther

Assistant Professor

Department of Computer Science & Engineering

Port City International University

DEDICATION

The thesis is dedicated to my beloved parents and my honorable teachers.

ACKNOWLEDGEMENT

First and foremost, I express my heartfelt gratitude to almighty for bestowing upon me the glorious blessing of successfully completing my final year thesis. It was such a great start of a journey to the world of research. I had the amazing privilege of studying under the guidance and encouragement of my respected supervisor, **Mrs. Farzina Akther**. Her vast knowledge gives me the opportunity to extended my horizons and make substantial progress.

(Signature of Student)

Md. Ashiful Hoque Chowdhury Tanbin

CSE 02707396

Department of Computer Science & Engineering

Port City International University

ABSTRACT

Idioms are integral to natural language comprehension, yet their contextual validation remains challenging in low-resource languages like Bangla. This paper presents a comprehensive comparative study of neural approaches for automated classification and explanation of idiom misuse in Bangla text. We introduce a manually annotated dataset comprising Bangla sentences with idiom usage labels and corresponding reasoning explanations. We benchmark twelve models across three paradigms – (1) Traditional machine learning (Logistic Regression, SVM, Naive Bayes and their ensemble) – (2) Deep learning architectures (BiLSTM, BiGRU, CNN, DNN and their ensemble) – (3) Transformer-based models (BanglaBERT, mBERT, XLM-R). Our experimental results demonstrate that BanglaBERT achieves superior performance with **94% F1-score** and **94% accuracy** on the classification task. For explainable reasoning generation, we employ BanglaT5 in a novel two-stage training paradigm - pseudo labeling with BanglaBERT predictions followed by fine-tuning on gold-standard annotations [2]. The reasoning generation model achieves a **ROUGE-L F1 score of 0.90** and **BLEU-4 score of 0.82** on the test set, demonstrating strong semantic alignment with human-written explanations. Our end-to-end pipeline moves beyond opaque idiom classification by not only identifying incorrect idiom usage, but also explaining why an idiom is inappropriate in context, making the system directly useful for Bangla language learners, writing assistance tools and linguistically grounded NLP research.

Keywords: Bangla NLP, Figurative Language Understanding, Transformer Models, Neural Language Models, Explainable AI, BanglaBERT, BanglaT5, Idiom Misuse Classification, Reason Generation, Low-Resource NLP, Comparative Model Analysis.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The part presents the setting of the work. The introduction part covers the problem statement, problem description, thesis orientation, motivation and objectives of the study, providing a structured overview of the research framework. It establishes the context and defines the direction of the research work. Language is humanity's most sophisticated tool for communication, encoding not only literal meanings but also cultural wisdom, metaphors and figurative expressions. Idioms fixed phrases whose meanings transcend their literal interpretations represent a particularly challenging aspect of natural language understanding. In Bengali, one of the world's most spoken languages with over 230 million native speakers, idioms play a vital role in everyday communication, literature and cultural expression. The advent of deep learning and transformer-based models has revolutionized Natural Language Processing, achieving remarkable success in high-resource languages like English and Chinese. However, low-resource languages such as Bangla face significant challenges in developing sophisticated NLP systems, particularly for nuanced tasks like figurative language understanding. Idiom classification is determining whether an idiom is used correctly or incorrectly in context. It remains an understudied problem in Bangla NLP. Moreover, modern AI systems increasingly require explainability - The ability to justify their decisions in human-understandable terms. This is especially critical in educational applications, linguistic research and language learning systems where users need to understand why a particular classification was made. This thesis addresses both challenges by developing a comprehensive idiom classification system with integrated explainable reasoning generation.

1.1 PROBLEM STATEMENT

Automated classification of Bangla idioms into correct and incorrect usage remains an open challenge in Bangla NLP due to several interconnected factors. First, idioms are highly context-dependent, creating semantic ambiguity. For example, the expression “অথৈ জলে পড়া” can refer either to literal drowning or to figurative extreme hardship, depending entirely on usage. Second, progress is constrained by severe resource limitations, including the lack of large annotated idiom-specific datasets, pretrained models tailored to Bangla idiomatic language and standardized evaluation benchmarks comparable to those available for English and Chinese. Third, there is no clear consensus on optimal modeling approaches, as rigorous comparative studies across traditional machine learning, deep learning and transformer-based methods are largely absent. Fourth, existing systems suffer from a major explainability gap. Finally, feature representation remains a complex issue, as it is still unclear whether idiom phrases alone, surrounding sentence context or hybrid representations combining idiom-level and sentence-level semantics yield the most robust and generalizable performance.

1.2 Problem Description

Idioms are figurative expressions whose meanings cannot be deduced from the literal interpretation of their constituent words. In Bangla, a morphologically rich and low-resource language spoken by over 230 million people, idioms play a crucial role in everyday communication, literature and cultural expression. However, the contextual validation of idiom usage presents significant computational challenges. Native speakers intuitively understand when an idiom is appropriately used, but non-native learners, automated translation systems and text generation models frequently misuse idioms by applying them in semantically inappropriate contexts. Current Bangla natural language processing systems lack robust mechanisms to detect idiom misuse and provide interpretable explanations for such errors. While substantial research exists for high-resource languages like English and Chinese, Bangla suffers from a scarcity of annotated datasets and benchmark models specifically designed for contextual idiom validation. Existing approaches focus primarily on idiom identification or literal-figurative classification but fail to address two critical gaps: (1) Binary classification of correct versus incorrect idiom usage in sentential contexts and (2) Generation of human-readable explanations justifying the classification decision [8]. These limitations hinder the development of intelligent language learning tools, grammar checkers and computational linguistics applications for Bangla.

1.3 Thesis Orientation

The study is positioned at the intersection of applied NLP, explainable AI, low-resource language processing and educational technology. It addresses two interconnected challenges: first, the automated detection of idiom misuse in context and second, the generation of pedagogically valuable explanations that articulate why particular usages are correct or incorrect. This dual focus distinguishes the work from conventional classification-only approaches and aligns with the growing demand for transparent, interpretable AI systems in educational applications. The thesis is structured to address the research problem through a systematic progression across multiple chapters. Chapter 1 establishes the foundational context through an introduction that presents the research motivation, problem statement, detailed problem description and clearly defined objectives that guide the entire investigation. Chapter 2 presents a comprehensive review of related work in idiom processing, low-resource NLP and explainable AI, establishing the theoretical foundation and identifying critical research gaps that justify this study. Chapter 3 details the methodology, introducing the manually annotated dataset with its annotation procedures and quality assurance protocols, describing the data structure and field specifications, explaining feature extraction strategies for different model families and presenting the architectural details of classification and reasoning generation models. Chapter 4 describes the hardware infrastructure and software frameworks employed, documenting the computational resources, libraries and pre-trained models that enable reproducible experimentation. Chapter 5 analyzes experimental results through quantitative metrics including accuracy, F1-score, ROC-AUC, confusion matrices and classification reports for all twelve models, alongside ROUGE and BLEU scores for reasoning generation, providing comparative evaluation across modeling paradigms. Chapter 6 concludes the thesis by synthesizing key findings, discussing broader implications for low-resource language processing and educational applications, acknowledging limitations of the current work and proposing concrete directions for future research.

1.4 MOTIVATION

The motivation for this research stems from four interconnected dimensions that collectively highlight the urgency and significance of developing automated idiom validation systems for Bangla. Bangla learners, both native students developing advanced skills and second-language learners, struggle with understanding and using idioms correctly. Existing grammar checkers and learning platforms do not validate idiom usage, forcing reliance on dictionaries or human experts. This problem is especially severe in rural and underserved areas with limited access to linguistic support. A real-time automated system that detects misuse and explains errors would enable scalable, self-directed and equitable language learning. Current Bangla NLP systems perform poorly when handling idioms. Machine translation systems often translate idioms literally, producing meaningless output, while text generation models inherit and reproduce idiom misuse from noisy training data. An idiom validation component could function as a post-processing layer, significantly improving the semantic quality of downstream applications such as translation, chatbots, sentiment analysis and content generation. Bangla suffers from a major resource gap compared to high-resource languages like English or Chinese, which have extensive idiom-focused datasets, benchmarks and models. The lack of annotated corpora containing idiom usage labels and explanatory reasoning is a core barrier to progress. This research aims to fill that gap by providing foundational resources that enable future work in low-resource NLP, cross-lingual transfer learning and comparative linguistic studies. Although modern neural models achieve strong performance, they operate as opaque black boxes that offer predictions without justification. In educational and assessment contexts, users such as students, teachers, curriculum designers and policymakers require transparent, human-aligned explanations to trust and effectively use AI systems. This research investigates whether models can not only judge idiom correctness accurately but also explain their decisions in linguistically meaningful terms.

1.5 Objective

The research objectives are organized into 3 thematic clusters that collectively address the identified gaps and advance the state of knowledge in Bangla idiom processing.

The first objective cluster focuses on dataset development as the foundational contribution of this research. We aim to construct a manually annotated dataset of Bangla sentences containing idioms, where each instance is labeled with binary usage correctness indicating whether the idiom is applied appropriately in the given context. Each labeled instance will be augmented with human-written reasoning explanations that articulate the linguistic justification for the correctness judgment. The dataset development process will incorporate rigorous quality assurance through inter-annotator agreement metrics and validation by expert linguists to ensure reliability and validity of annotations.

The second objective cluster centers on comparative model analysis to establish empirical benchmarks for Bangla idiom classification. We will implement traditional machine learning models including Logistic Regression, Support Vector Machines and Naive Bayes, utilizing TF-IDF feature representations to capture lexical patterns. Subsequently, we will develop deep learning architectures encompassing Bidirectional Long Short-Term Memory networks, Bidirectional Gated Recurrent Units, Convolutional Neural Networks and Deep Neural Networks with learned embedding representations [21]. Finally, we will fine-tune state-of-the-art transformer models specifically BanglaBERT, multilingual BERT and XLM-RoBERTa to leverage pre-trained contextual representations. Each model family will be rigorously evaluated using standard classification metrics including F1-score, accuracy, precision, recall, ROC-AUC and average precision, with the best-performing architecture identified for integration into the reasoning generation pipeline.

The third objective cluster addresses explainable reasoning generation through a novel two-stage training methodology. We will employ BanglaT5, a Sequence-to-Sequence transformer model, to generate natural language explanations for idiom usage classifications. The first training stage will utilize pseudo-labeling, where the model learns to generate reasoning based on predictions from the best-performing classifier, enabling exposure to a broader range of usage patterns. The second stage will involve fine-tuning on gold-standard annotations with human-written explanations to align generated text with expert linguistic reasoning. The quality of generated explanations will be assessed using automatic metrics including ROUGE-L and BLEU-4, supplemented by qualitative analysis of semantic coherence and linguistic accuracy.

CHAPTER 2
LITERATURE REVIEW

This chapter presents a comprehensive review of existing research in idiom processing, Bangla natural language processing, explainable artificial intelligence and commonsense reasoning. Section 2.1 provides foundational background on idioms, NLP evolution and the challenges specific to low-resource languages like Bangla. Section 2.2 systematically analyzes current literature, examining six key areas: explainable AI for commonsense reasoning, Bangla specific NLP models, idiom translation and detection, figurative language understanding, transformer-based approaches and evaluation methodologies. The chapter concludes by identifying critical research gaps that motivate this thesis.

2.1 BACKGROUND STUDY

Idioms are multiword expressions whose meanings cannot be derived from the literal meanings of their constituent words [20]. In Bangla, idioms represent a particularly rich linguistic phenomenon, carrying cultural wisdom and metaphorical meanings that transcend literal interpretation. These expressions exhibit non-compositionality, semantic opacity, context dependency and structural diversity, making their computational processing a challenging task in natural language understanding.

The fundamental challenge in idiom processing lies in distinguishing figurative from literal usage. The same phrase can carry different meanings depending on the surrounding context. For example, "অথৈ জলে পড়া" (falling into deep water) can mean facing extreme danger in one context or literal drowning in another. This ambiguity requires sophisticated contextual understanding that goes beyond simple keyword matching or surface-level pattern recognition.

Bangla idioms typically follow several structural patterns including noun phrases, verb phrases, prepositional phrases and comparative constructions. Understanding these structural patterns is crucial for computational processing, as feature extraction strategies must account for this linguistic diversity. The complexity of idiom processing is further compounded by the fact that many idioms permit syntactic variations while maintaining their figurative meaning, whereas others exhibit syntactic fixedness [13].

The field of natural language processing has undergone several paradigms shifts over the past seven decades. Early systems relied heavily on hand-crafted rules and symbolic representations, following the linguistic theories of Chomsky and others [5]. These rule-based approaches, while interpretable, struggled with the inherent ambiguity and complexity of natural language, particularly when dealing with figurative expressions and context-dependent meanings.

The introduction of statistical methods marked the first major revolution in NLP. Hidden Markov Models, Conditional Random Fields and n-gram language models enabled systems to learn patterns from data rather than relying exclusively on manually coded rules. However, these approaches still required extensive feature engineering and struggled to capture long-range dependencies and semantic relationships.

The emergence of neural networks brought distributed representations to NLP. Word2Vec and GloVe demonstrated that meaningful semantic relationships could be learned automatically from large text corpora [14][17], with words appearing in similar contexts clustering together in vector space. Recurrent Neural Networks, particularly Long Short-Term Memory networks and Gated Recurrent Units, enabled models to process sequences of arbitrary length while maintaining memory of previous inputs.

The transformer architecture introduced in 2017 by Vaswani and colleagues revolutionized the field through self-attention mechanisms. BERT [22], introduced by Devlin and colleagues in 2019, demonstrated that pre-training on massive unlabeled corpora followed by fine-tuning on specific tasks could achieve state-of-the-art performance across diverse NLP applications [7]. For Bangla, this evolution culminated in the development of BanglaBERT in 2022 and BanglaT5 in 2023, enabling sophisticated language understanding for this low-resource language [1].

Modern AI systems, particularly deep neural networks, often function as "black boxes" that produce accurate predictions without revealing their reasoning process. This opacity poses critical challenges across multiple dimensions. From a user trust perspective, individuals hesitate to rely on unexplained decisions, especially in domains like education, healthcare and legal systems where understanding the rationale is as important as the decision itself.

The debugging and improvement of AI systems also suffers from lack of explainability. When a model makes an error, developers need to understand why the mistake occurred to improve the system. Without insight into the model's reasoning process, identifying and correcting failure modes becomes a trial-and-error process rather than a systematic engineering effort.

In educational applications, explainability is not just desirable but essential. Language learners studying Bangla idioms need to understand not only whether an idiom is used figuratively or literally, but also why that classification is correct. An unexplained classification provides no pedagogical value, whereas a well-reasoned explanation can teach students to recognize similar patterns in new contexts.

Regulatory frameworks increasingly mandate explainability for automated decision systems. The European Union's General Data Protection Regulation includes provisions for a "right to explanation" for automated decisions. Similar regulations are emerging globally, making explainability not just a technical preference but a legal requirement in many applications.

Three main approaches to explainability have emerged in NLP research. Attention visualization highlights which input words influenced predictions, but research has shown that attention weights may not faithfully represent the model's true reasoning process. Rationale extraction selects specific text spans as evidence for predictions, providing extractive explanations. Natural language generation produces free-text explanations, offering the most human-aligned form of explainability but requiring sophisticated sequence-to-sequence modeling.

2.2 LITERATURE REVIEW

Prior research on explainable AI and commonsense reasoning has shown that while modern neural models can accurately validate linguistic plausibility, generating high-quality natural language explanations remains a challenging problem [3]. Studies such as Ge et al. (2025) demonstrate that structured, two-stage prompting can substantially improve explanation quality [9] but their work is limited to English commonsense explanation generation and does not address classification, idiomatic expressions or low-resource languages. Similarly, the SemEval-2020 ComVE task revealed a clear performance gap between validation accuracy and free-form explanation generation, indicating that classification and explanation require distinct modeling strategies [12][23]. Evaluations of pretrained language models further show that even powerful transformers struggle with implicit and multi-step reasoning, which is critical for understanding idiomatic language. In Bangla NLP, recent advances such as BanglaBERT have achieved strong results on standard tasks, yet empirical studies consistently report failure on idioms and non-compositional expressions [1]. Sentence validation systems with near-human accuracy explicitly struggle with idiomatic usage, confirming that figurative language remains unresolved. Rule-based Bengali–English idiom translation systems demonstrate moderate success but suffer from limited coverage and poor generalization. Across the literature, no prior work systematically addresses Bangla idiom usage classification, combines classification with natural language explanation or evaluates explainability in Bangla. These gaps motivate the present work, which focuses on joint idiom classification and explanation in Bangla using learned representations, multi-field input fusion and two-stage explanation generation to reduce implicit reasoning burden and improve interpretability.

CHAPTER 3
METHODOLOGY

This section presents the overall system design and experimental workflow adopted in this research. The methodology systematically integrates data preparation, feature engineering, model training, reasoning generation and evaluation to enable automated Bangla idiom validation and explanation. An overview of the proposed framework is illustrated in Figure 3.1. The proposed approach follows a two-phase pipeline.

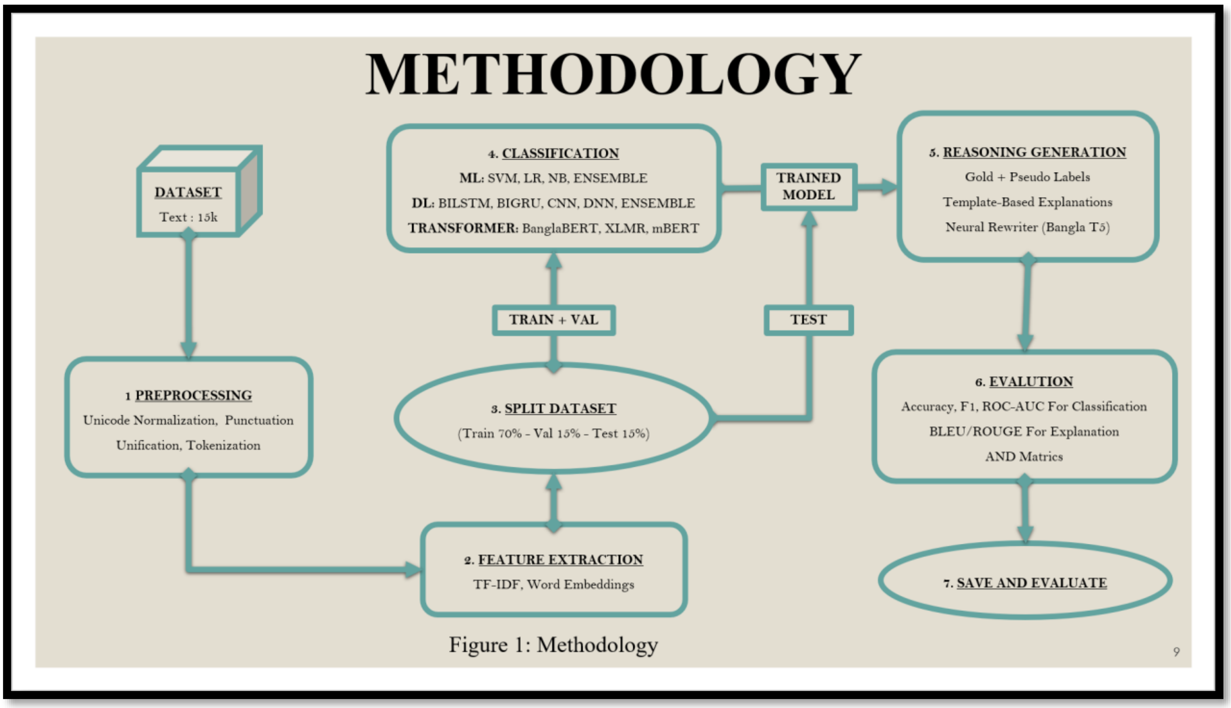

METHODOLOGY
DATASET
Text : 15k
1 PREPROCESSING
Unicode Normalization, Punctuation Unification, Tokenization
2. FEATURE EXTRACTION
TF-IDF, Word Embeddings
3. SPLIT DATASET
(Train 70% - Val 15% - Test 15%)
TRAIN + VAL
TEST
4. CLASSIFICATION
ML: SVM, LR, NB, ENSEMBLE
DL: BILSTM, BIGRU, CNN, DNN, ENSEMBLE
TRANSFORMER: BanglaBERT, XLMR, mBERT
TRAINED MODEL
5. REASONING GENERATION
Gold + Pseudo Labels
Template-Based Explanations
Neural Rewriter (Bangla T5)
6. EVALUTION
Accuracy, F1, ROC-AUC For Classification
BLEU/ROUGE For Explanation
AND Matrics
7. SAVE AND EVALUATE
Figure 1: Methodology
9

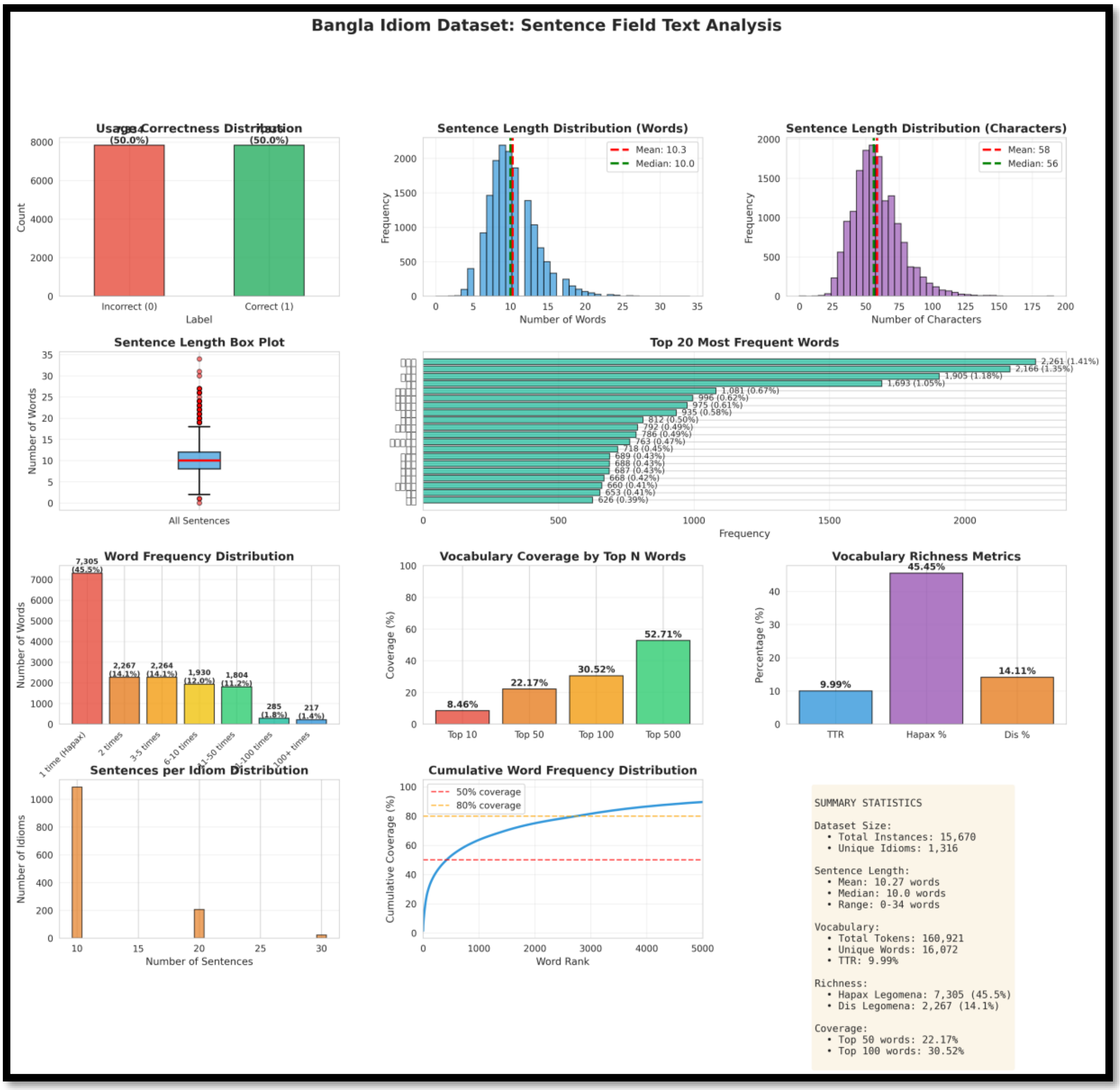
Figure 3. 1: Methodology

In the first phase, a manually annotated dataset of approximately 15,670 Bangla sentences containing idiomatic expressions is constructed. Each instance is labeled to indicate whether the idiom usage is contextually correct or incorrect, along with a corresponding human-written justification. The raw text undergoes preprocessing steps including Unicode normalization, punctuation unification and tokenization to ensure linguistic consistency. Feature extraction is then performed using TF-IDF and word embedding representations for traditional and deep learning models, while transformer-based models operate directly on raw text. The dataset is stratified into training, validation and test splits to maintain label balance. Multiple classification models spanning three paradigms are trained and evaluated: traditional machine learning models (Logistic Regression, SVM, Naive Bayes and ensemble), deep learning architectures (BiLSTM, BiGRU, CNN, DNN and ensembles) and transformer-based models (BanglaBERT, XLM-R and mBERT). Among these, BanglaBERT achieves the strongest performance, reaching **94% F1-score** and **94% accuracy** and is selected as the final classifier for the system.

In the second phase, the focus shifts to explanation generation. A BanglaT5-based neural text generation model is employed to produce human-interpretable reasoning for idiom misuse. Training is conducted in two stages: first, pseudo-labeled explanations are generated using predictions from the best-performing classifier to provide large-scale supervision; second, the model is fine-tuned on gold human-written explanations to improve linguistic fluency, semantic alignment and faithfulness. This hybrid supervision strategy balances scalability with explanation quality. The explanation generator is evaluated using automatic metrics including BLEU-4 and ROUGE-L, complemented by qualitative analysis to assess coherence, relevance and interpretability. Finally, the top-performing classifier and explanation model are integrated into a unified end-to-end system that processes raw Bangla text, detects idiom misuse and generates natural language explanations. The results demonstrate strong generalization and practical applicability for Bangla language learning, writing assistance and text correction tasks.

3.1 DATASET DESCRIPTION

The dataset constructed for this research represents the first manually annotated corpus specifically designed for Bangla idiom usage validation and explainable reasoning generation. The corpus comprises 15,670 sentences spanning 1,316 distinct Bangla idioms (বাগধারা), with an average of 11.91 sentences per idiom to ensure comprehensive coverage of varied contextual applications. Each idiom is represented through multiple sentence contexts that demonstrate both correct and incorrect usage patterns, enabling models to learn nuanced distinctions between appropriate and inappropriate idiomatic applications. The dataset addresses a critical gap in Bangla natural language processing resources by providing the first benchmark for simultaneous classification and explanation generation tasks in the domain of idiomatic expression validation. The construction of this dataset involved approximately 600 person-hours of expert linguistic effort, encompassing idiom selection from authoritative sources, sentence crafting to reflect authentic usage scenarios, meticulous labeling of usage correctness and composition of pedagogically valuable reasoning explanations. The resulting corpus exhibits balanced label distribution with 7,836 instances (50.01%) representing correct idiom usage and 7,834 instances (49.99%) representing incorrect usage, ensuring that classification models do not suffer from class imbalance bias during training and evaluation. An analysis of sentence field is illustrated in Figure 3.2.



```
SUMMARY STATISTICS

Dataset Size:
  • Total Instances: 15,670
  • Unique Idioms: 1,316

Sentence Length:
  • Mean: 10.27 words
  • Median: 10.0 words
  • Range: 0-34 words

Vocabulary:
  • Total Tokens: 160,921
  • Unique Words: 16,072
  • TTR: 9.99%

Richness:
  • Hapax Legomena: 7,305 (45.5%)
  • Dis Legomena: 2,267 (14.1%)

Coverage:
  • Top 50 words: 22.17%
  • Top 100 words: 30.52%
```

Figure 3. 2: Sentence Field Text Analysis

3.2 Data Structure & Field Specifications

Each instance in the dataset is represented as a JSON Lines (JSONL) record containing six structured fields that capture both the linguistic context and annotation information necessary for training and evaluation. The data structure is designed to support two distinct but interconnected tasks: idiom usage correctness classification and reasoning generation.

idiom_word: This field contains the Bangla idiomatic expression in its canonical form as it appears in standard linguistic references and authoritative dictionaries. The idioms are preserved in their dictionary headword form, maintaining consistency with established lexicographic conventions. For example, "অথৈ জলে পড়া" (literally "falling into bottomless water") represents the standardized form of the idiom regardless of inflectional variations that may occur in actual sentence usage. The field facilitates model training by providing explicit identification of the target idiom within each instance.

idiom_meaning: This field provides the figurative interpretation of the idiom in Bangla, representing the semantic content that native speakers understand when encountering the expression in discourse. The meanings are expressed concisely, typically in 3-6 words, capturing the core semantic contribution of the idiom. For the idiom "অথৈ জলে পড়া", the meaning field contains "চরম বিপদে পড়া" (falling into extreme danger/difficulty), which conveys the figurative sense without requiring literal interpretation of the constituent words. This field enables models to access explicit semantic information that may inform usage validation decisions.

sentence: This field contains a complete Bangla sentence in which the idiom appears, constructed to provide sufficient context for determining usage appropriateness. Sentences are crafted to reflect natural language usage patterns observed in informal conversation, formal writing and literary contexts. The sentence length averages 10.27 words (median: 10 words, standard deviation: 3.20 words), reflecting authentic Bangla sentence construction while maintaining focus on idiom-centric contexts. Sentence complexity ranges from simple declarative statements to compound and complex structures incorporating subordinate clauses, ensuring diversity in syntactic patterns. For example, "মার চাকরি চলে যাওয়ায় পরিবার অথৈ জলে পড়েছে" (The family has fallen into extreme difficulty due to mother's job loss) demonstrates appropriate application of the idiom to describe a situation of financial hardship.

i_label: This binary field indicates idiom identification through annotation, marking whether the expression in the sentence constitutes an idiomatic use (1) or a literal compositional phrase (0). This field distinguishes genuine idiomatic uses, where the meaning transcends compositional interpretation of constituent words, from homophonous literal uses where the same word sequence retains transparent compositional meaning. While all instances in the current dataset focus on idiomatic uses (i_label = 1), this field supports potential future extension to include literal-figurative disambiguation tasks.

s_label: This field represents the primary classification target, indicating usage correctness where 1 denotes appropriate contextual application of the idiom and 0 indicates misuse or inappropriate application. The annotation reflects expert judgment regarding whether the idiom's figurative meaning aligns semantically and pragmatically with the sentence context. Correct usage instances demonstrate semantic coherence between the idiom's meaning and the described situation, appropriate register matching between the idiom's typical usage domain and the sentence context and logical consistency in the application of the figurative sense. Incorrect usage instances exhibit semantic mismatches where the idiom's meaning does not fit the context, pragmatic inappropriateness where the idiom occurs in unsuitable registers or discourse contexts or syntactic violations where the idiom appears in structurally non-standard forms. The balanced distribution (50.01% correct, 49.99% incorrect) ensures models learn to discriminate both classes equally well.

reason: This field contains a human-written explanation in Bangla that articulates the linguistic justification for the usage label, describing why the idiom application is deemed correct or incorrect based on semantic coherence, pragmatic appropriateness and contextual alignment. Reasoning explanations average 18.5 words in length, providing sufficient detail for pedagogical utility while remaining concise enough for practical generation by sequence-to-sequence models. For correct usage, explanations typically affirm alignment between figurative meaning and context, such as "উক্ত 'অথৈ জলে পড়া' বাগধারাটির অর্থ হলো 'চরম বিপদে পড়া' এবং এটি সঠিকভাবে ব্যবহৃত হয়েছে।" (The idiom 'অথৈ জলে পড়া' means 'falling into extreme danger' and it has been used correctly). For incorrect usage, explanations identify specific mismatches such as semantic incompatibility, pragmatic violations or logical inconsistencies. This field enables the reasoning generation component of the research, training models to produce linguistically grounded explanations suitable for language learning applications.

3.3 Annotation Methodology

The dataset construction process followed a systematic multi-stage methodology designed to ensure annotation quality, consistency and linguistic validity. The comprehensive annotation workflow encompassed idiom selection, sentence construction, labeling, quality assurance and validation through inter-annotator agreement measurement.

The dataset was annotated by a team of three collaborators, all of whom are native Bangla speakers with strong familiarity with idiomatic usage in both literary and colloquial contexts. The annotators worked closely together to identify correct and incorrect idiom usage and to provide corresponding reasoning explanations. Prior to full-scale annotation, the team jointly annotated a pilot set of 1000 sentences to align their understanding of annotation guidelines and ensure consistency across labels and explanations throughout the dataset construction process.

The 1,316 idioms included in the dataset were collected exclusively from a single publicly available source, namely the Bangla Wikipedia page ‘বাংলা বাগধারার তালিকা - উইকিপিডিয়া’ [24]. This source provides a consolidated and widely referenced list of Bangla idiomatic expressions, making it suitable for constructing a standardized idiom inventory for this study. All idioms listed on the page at the time of collection were included to avoid subjective filtering or selective bias. Rather than applying frequency-based or semantic pre-selection, the dataset construction focused on comprehensive coverage of the source list, ensuring consistency and transparency in idiom selection. Semantic diversity naturally emerged from the Wikipedia compilation itself, which contains idioms spanning emotional states, interpersonal relations, cognitive conditions, social behavior and everyday experiences commonly reflected in Bangla language use. This approach ensured feasibility for manual annotation while maintaining broad representational coverage across idiomatic meanings and usage contexts.

For each selected idiom, annotators constructed multiple sentence contexts following specified guidelines to achieve balanced representation of correct and incorrect usage patterns. The target distribution aimed for approximately equal numbers of correct and incorrect instances per idiom, with actual distribution determined by the natural range of plausible usage scenarios for each expression.

Correct Usage Construction: Correct usage sentences were crafted to demonstrate appropriate semantic and pragmatic application following three criteria. First, semantic coherence required that the idiom's figurative meaning aligned naturally with the described situation, creating logical consistency between the idiomatic sense and contextual scenario. Second, register appropriateness ensured that the idiom occurred in discourse contexts matching its typical usage domain, avoiding jarring combinations of formal idioms in casual contexts or colloquial expressions in formal writing. Third, syntactic naturalness verified that the idiom appeared in structurally standard forms consistent with native speaker intuitions, avoiding forced or artificial sentence constructions.

Incorrect Usage Construction: Incorrect usage sentences were designed to represent authentic error types that language learners might produce, ensuring ecological validity for educational applications. The construction strategy identified four primary error categories. Semantic mismatch errors involved applying idioms to contexts where their figurative meanings did not fit, such as using an idiom expressing joy to describe a sad situation or applying an expression indicating failure to a success scenario. Pragmatic inappropriateness errors placed idioms in unsuitable registers or discourse contexts, such as using highly colloquial expressions in formal academic writing or deploying archaic literary idioms in casual conversation. Logical inconsistency errors created contradictions between the idiom's meaning and other sentence elements, such as temporal mismatches or causal incoherence. Syntactic violation errors presented the idiom in structurally non-standard forms, though this category was minimized to maintain focus on semantic and pragmatic validation.

This deliberate construction of error examples ensures that models learn to recognize genuine misuse patterns reflecting actual learner difficulties rather than artificial contrasts unlikely to occur in authentic language use. The error distribution mirrors empirical findings from pedagogical research on idiom acquisition, prioritizing semantic and pragmatic errors which constitute the majority of learner mistakes.

Each annotator independently labeled sentences for idiom identification (i_label) and usage correctness (s_label), subsequently composing reasoning explanations that articulated the linguistic basis for their judgments. The labeling process followed structured decision procedures to promote consistency.

For idiom identification, annotators determined whether the target expression functioned idiomatically with non-compositional meaning or literally with transparent compositional interpretation. For usage correctness, annotators evaluated semantic alignment between the idiom's figurative meaning and sentence context, pragmatic appropriateness of the idiom for the discourse situation and logical coherence of the application. Borderline cases where usage appropriateness admitted multiple defensible interpretations were flagged for consensus discussion.

The reasoning explanations followed a structured template encouraging annotators to reference the idiom's meaning, describe the contextual situation and articulate the semantic relationship justifying the correctness judgment. For correct usage, explanations typically followed the pattern: "উক্ত '[IDIOM]' বাগধারাটির অর্থ হলো '[MEANING]' এবং এটি সঠিকভাবে ব্যবহৃত হয়েছে।" (The idiom '[IDIOM]' means '[MEANING]' and it has been used correctly). For incorrect usage, explanations identified the specific nature of the error: "উক্ত ‘[IDIOM]’ বাগধারাটির অর্থ হলো ‘[MEANING]’, কিন্তু এটি [ERROR_DESCRIPTION], যা ভুল।" (In this sentence, the idiom '[IDIOM]' has been used incorrectly because [ERROR_EXPLANATION], that is wrong). This structured approach ensures that reasoning texts provide pedagogically valuable feedback suitable for language learning applications.

3.4 BALANCED DATASET OVERVIEW

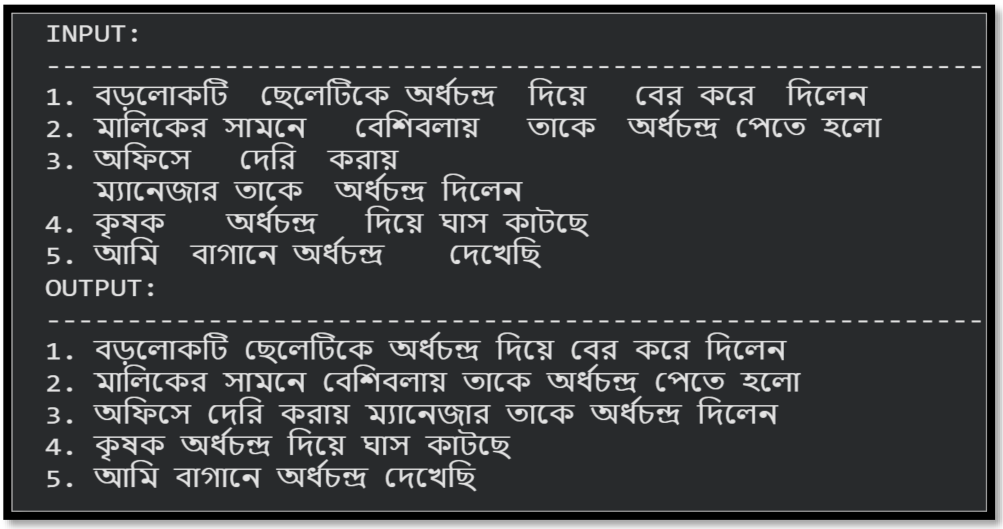
To ensure reliable evaluation and minimize class bias, a carefully curated and balanced dataset was constructed for this study. The corpus comprises Bangla sentences containing idiomatic expressions, annotated for contextual correctness and accompanied by human-written reasoning. Special attention was given to maintaining an equal distribution between correct and incorrect idiom usage across training, validation and test splits. In addition to label balance, the dataset captures a wide range of idioms and sentence lengths, enabling robust assessment of both classification and explanation generation models. Table - 1 summarizes the key characteristics of the dataset, including size, label distribution, split ratios, annotation effort and linguistic properties.

Table 1: Balanced Dataset Overview

Category	Metric	Value
Overall Statistics	Total Instances	15,670
	Total Idioms	1,316
	Avg. Sentences per Idiom	11.91
	Correct Usage (s_label=1)	7,836 (50.01%)
	Incorrect Usage (s_label=0)	7,834 (49.99%)
	Avg. Sentence Length	10.27 words
	Avg. Reason Length	18.5 words
Train Split	Instances	10,969 (70%)
	Correct Usage	~5,484
	Incorrect Usage	~5,485
Validation Split	Instances	2,350 (15%)
	Correct Usage	~1,175
	Incorrect Usage	~1,175
Test Split	Instances	2,351 (15%)
	Correct Usage	~1,177
	Incorrect Usage	~1,174
	Annotation Hours	~600 person-hours
Source	Idiom : বাংলা বাগধারার তালিকা - উইকিপিডিয়া Dataset: Manually Created Dataset	

3.5 PREPROCESSING

Prior to model training, the dataset underwent systematic preprocessing to address Bangla-specific orthographic and encoding challenges while preserving linguistic information essential for idiom processing. Figure 3.3 illustrates the effect of the text normalization process applied during preprocessing.



```
INPUT:
------------------------------------------------------------
1. বড়লোকটি  ছেলেটিকে অর্ধচন্দ্র  দিয়ে   বের করে  দিলেন
2. মালিকের সামনে   বেশিবলায়   তাকে  অর্ধচন্দ্র পেতে হলো
3. অফিসে   দেরি  করায়
   ম্যানেজার তাকে  অর্ধচন্দ্র দিলেন
4. কৃষক    অর্ধচন্দ্র   দিয়ে ঘাস কাটছে
5. আমি  বাগানে অর্ধচন্দ্র    দেখেছি
OUTPUT:
------------------------------------------------------------
1. বড়লোকটি ছেলেটিকে অর্ধচন্দ্র দিয়ে বের করে দিলেন
2. মালিকের সামনে বেশিবলায় তাকে অর্ধচন্দ্র পেতে হলো
3. অফিসে দেরি করায় ম্যানেজার তাকে অর্ধচন্দ্র দিলেন
4. কৃষক অর্ধচন্দ্র দিয়ে ঘাস কাটছে
5. আমি বাগানে অর্ধচন্দ্র দেখেছি
```

Figure 3. 3: Normalized Dataset Sample

Unicode normalization using the NFC (Canonical Decomposition followed by Canonical Composition) standard ensured consistent representation of Bangla characters. Bangla script admits multiple valid Unicode encodings for certain character sequences, particularly for vowel diacritics and consonant conjuncts. For example, the sequence "কি" (ki) can be represented using either precomposed characters or decomposed base characters with combining diacritics. NFC normalization canonicalizes these representations, ensuring that semantically identical text receives identical encoding. Zero-width characters including Zero-Width Non-Joiner (ZWNJ, U+200C) and Zero-Width Joiner (ZWJ, U+200D), which appear in some Bangla text encodings to control rendering of consonant conjuncts but lack semantic content, were systematically removed to prevent spurious token distinctions. These invisible characters, when present, can cause tokenizers to treat identical words as distinct types, artificially inflating vocabulary size and preventing models from recognizing semantic equivalence.

Whitespace normalization collapsed multiple consecutive spaces into single spaces, removed leading and trailing whitespace from all text fields and standardized line breaks. This preprocessing addresses inconsistencies in manual text entry where annotators might inadvertently introduce extra spacing. Consistent whitespace treatment ensures that tokenization procedures produce reliable results unaffected by formatting artifacts.

Validation checks confirmed data integrity by verifying presence of all required fields in each record, detecting and removing any duplicate sentences to prevent data leakage between training and evaluation sets and ensuring label consistency. Quality assurance identified a minimal number of instances with empty sentence fields (indicated by 0-word minimum in length statistics), which were flagged for manual review and correction or removal to maintain corpus quality. Field type validation ensured that numeric labels (i_label and s_label) contained only valid integer values (0 or 1) and that text fields contained non-empty strings of appropriate length. These validation procedures yielded a clean dataset suitable for reproducible experimentation while documenting any data quality issues encountered during preprocessing.

The training set comprises 10,969 instances (70% of total), providing substantial data for model parameter learning across diverse idioms and usage contexts. The validation set contains 2,350 instances (15% of total), enabling hyperparameter tuning, model selection and early stopping decisions without contaminating the test set reserved for final evaluation. The test set contains 2,351 instances (15% of total), held out exclusively for unbiased assessment of generalization performance on completely unseen data. The 70-15-15 split ratio balances competing objectives of maximizing training data for effective learning, providing sufficient validation data for reliable model selection and reserving adequate test data for statistically meaningful evaluation. This partitioning scheme follows established best practices in machine learning research while adapting to the moderate-scale dataset size where larger test sets would excessively reduce training data.

3.6 Feature Extraction

Feature extraction transforms raw text into numerical representations that machine learning models can process. This research employs three distinct approaches tailored to different model architectures: traditional machine learning, deep learning and transformer-based models. All approaches use the same input structure, combining the idiom word (বাগধারা), idiom meaning (অর্থ) and sentence (বাক্য) into a unified text format. This consistency ensures fair comparison while allowing each approach to leverage its unique strengths.

The machine learning models use TF-IDF (Term Frequency-Inverse Document Frequency) vectorization to convert text into sparse numerical features. Unlike deep learning approaches that learn features automatically, TF-IDF is a hand-crafted method that captures statistical patterns in text based on term frequencies and their distribution across documents. The pipeline employs three separate TF-IDF vectorizers, each optimized for different input components. The sentence field uses character-level analysis with 3–6-character n-grams and 5,000 maximum features, effectively capturing morphological variations common in Bengali. The idiom meaning field uses word-level analysis with unigrams and bigrams limited to 500 features, capturing semantic word combinations. The idiom word field uses character-level analysis with 2–4-character n-grams and 300 features to identify distinctive character patterns. These three feature matrices are horizontally concatenated to create a final representation of 5,800 dimensions. The resulting sparse matrix format conserves memory since most features are zero for any given sample. All feature matrices are saved in compressed. ‘npz’ format, while fitted vectorizers are serialized as pickle files for consistent transformation across training, validation and test sets, ensuring reproducibility and preventing data leakage.

Deep learning models use learned embeddings rather than hand-crafted features. The input text undergoes tokenization where each unique word receives an integer identifier from a vocabulary of 50,000 most frequent tokens. An out-of-vocabulary token handles unseen words. These variable-length integer sequences are padded or truncated to a fixed length of 180 tokens, creating uniform input matrices of shape (samples × 180). The actual feature extraction occurs through an embedding layer that maps each integer to a dense 128-dimensional vector. Initially random, these embeddings are learned during training to capture semantic and syntactic relationships between words. Different neural architectures process these embeddings distinctly. Bidirectional LSTM and GRU models capture sequential context from both directions with 128 hidden units. The DNN architecture applies global average pooling, treating text as a bag of embedded words before passing through dense layers [11]. The CNN model uses one-dimensional convolution with 128 filters and kernel size 5 to detect local patterns before global max pooling. Unlike TF-IDF's sparse 5,800-dimensional vectors, these learned embeddings are dense, lower-dimensional and capture richer semantic information through the training process. Raw text data is saved separately in NumPy arrays organized by split and field, allowing models to apply tokenization during training while enabling transformer models to use the same data with their specialized tokenizers. The trained tokenizer is saved as JSON, preserving vocabulary mapping for consistent preprocessing during inference.

Transformer models represent the most sophisticated approach, leveraging pre-trained language models trained on massive corpora. Unlike previous approaches that start from scratch, transformers employ transfer learning with models already containing rich linguistic representations from billions of words. Three transformer models are used: BanglaBERT with WordPiece tokenization trained on 27.5 GB of Bengali text, mBERT with WordPiece tokenization trained on 104 languages and XLM-RoBERTa with SentencePiece tokenization trained on 2.5 TB of multilingual data. These models have vocabularies exceeding 100,000 subword units, far larger than the 50,000-word vocabulary in custom DL tokenizers. Subword tokenization breaks words into smaller meaningful units, allowing models to handle unseen words by composing them from known pieces. Sequences are truncated to 192 tokens to accommodate special tokens like [CLS] and [SEP]. Feature extraction occurs through pre-trained encoder layers using multi-head self-attention mechanisms that compute each token's representation in relation to all other tokens. This produces 768-dimensional contextual embeddings that capture long-range dependencies. Unlike static embeddings, transformer representations are contextual—the same word receives different embeddings depending on surrounding context, making them particularly effective for idiom detection where context determines literal versus figurative meaning. During fine-tuning, the entire pre-trained model adapts to the idiom classification task with a low learning rate (2e-5) to preserve useful linguistic knowledge while specializing for Bengali idiom patterns. This is computationally expensive but allows sophisticated representation learning from limited task-specific data.

3.6.1 TF-IDF Calculation

How often a term appears in a document:

$$TF(t,d) = \frac{count\ of\ t\ in\ d}{total\ terms\ in\ d}$$

For character n-grams:

$$TF = \frac{n\text{-}gram\ count}{total\ n\text{-}grams}$$

How rare a term is across all documents:

$$IDF(t) = log\left(\frac{1+n}{1+df(t)}\right) + 1$$

- n= total documents
- $df(t)$= documents containing the term
- Smoothing avoids zero values

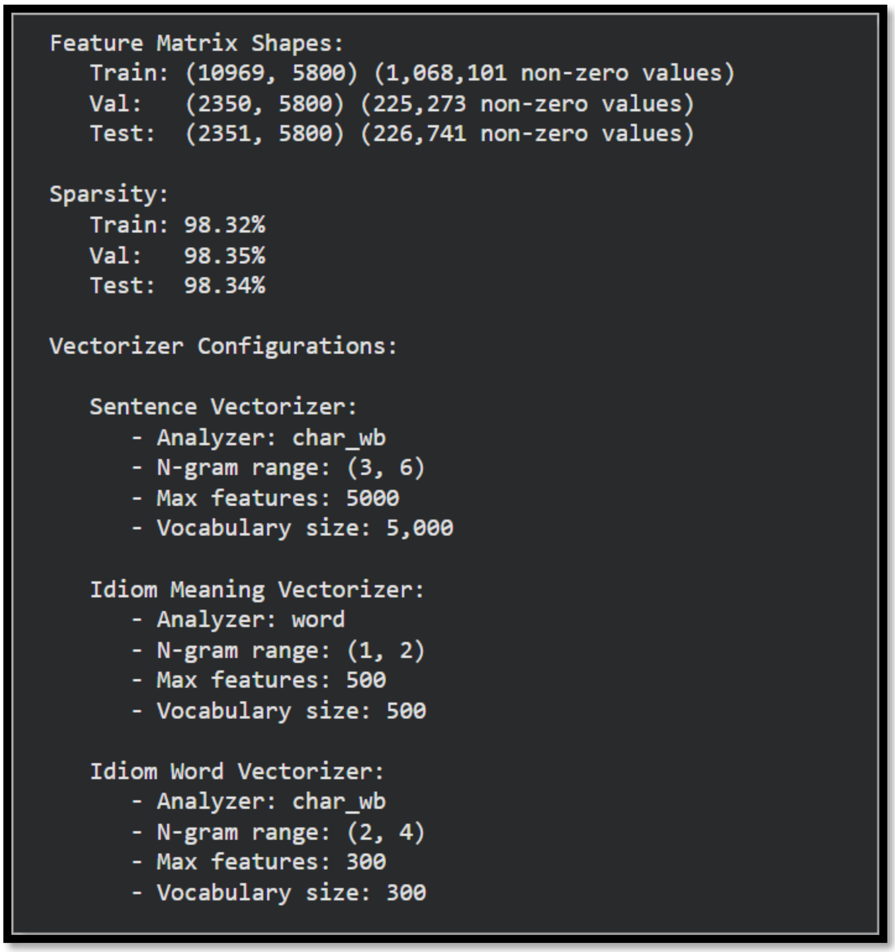
Importance of a term in a document:

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

Scales vectors so document length doesn’t matter:

$$Normalized\ vector\ =\ TF - IDF\ /vector\ length$$

Here, Figure 3.4 demonstrates the numerical feature representation generated from the normalized Bangla text using the Term Frequency–Inverse Document Frequency (TF-IDF) technique.



```
Feature Matrix Shapes:
   Train: (10969, 5800) (1,068,101 non-zero values)
   Val:   (2350, 5800) (225,273 non-zero values)
   Test:  (2351, 5800) (226,741 non-zero values)

Sparsity:
   Train: 98.32%
   Val:   98.35%
   Test:  98.34%

Vectorizer Configurations:

   Sentence Vectorizer:
      - Analyzer: char_wb
      - N-gram range: (3, 6)
      - Max features: 5000
      - Vocabulary size: 5,000

   Idiom Meaning Vectorizer:
      - Analyzer: word
      - N-gram range: (1, 2)
      - Max features: 500
      - Vocabulary size: 500

   Idiom Word Vectorizer:
      - Analyzer: char_wb
      - N-gram range: (2, 4)
      - Max features: 300
      - Vocabulary size: 300
```

Figure 3. 4: TF-IDF Result

3.7 ALGORITHMS

3.7.1 ML MODELS

3.7.1.1 LOGISTIC REGRESSION

Logistic Regression is a linear classification model that estimates the probability of a binary outcome using the logistic (sigmoid) function. To improve generalization and reduce overfitting, Elastic Net regularization is applied, which combines L1 (Lasso) and L2 (Ridge) penalties. The L1 component encourages sparsity by driving less informative feature coefficients to zero, effectively performing feature selection, while the L2 component stabilizes the model by smoothly shrinking coefficient magnitudes. In this study, an l1_ratio of 0.2 is used, assigning 20% weight to L1 and 80% to L2, balancing interpretability and robustness. The training pipeline first fits the base model using the training data, after which probability calibration is performed on the validation set using Isotonic Regression to correct for mis calibrated confidence estimates. Rather than relying on a fixed 0.5 cutoff, an optimal decision threshold is then selected on the validation set to maximize classification performance. This optimized threshold is finally applied to the test-set predictions to ensure fair and unbiased evaluation. Figure 3.5 describes the structural architecture of LR model.

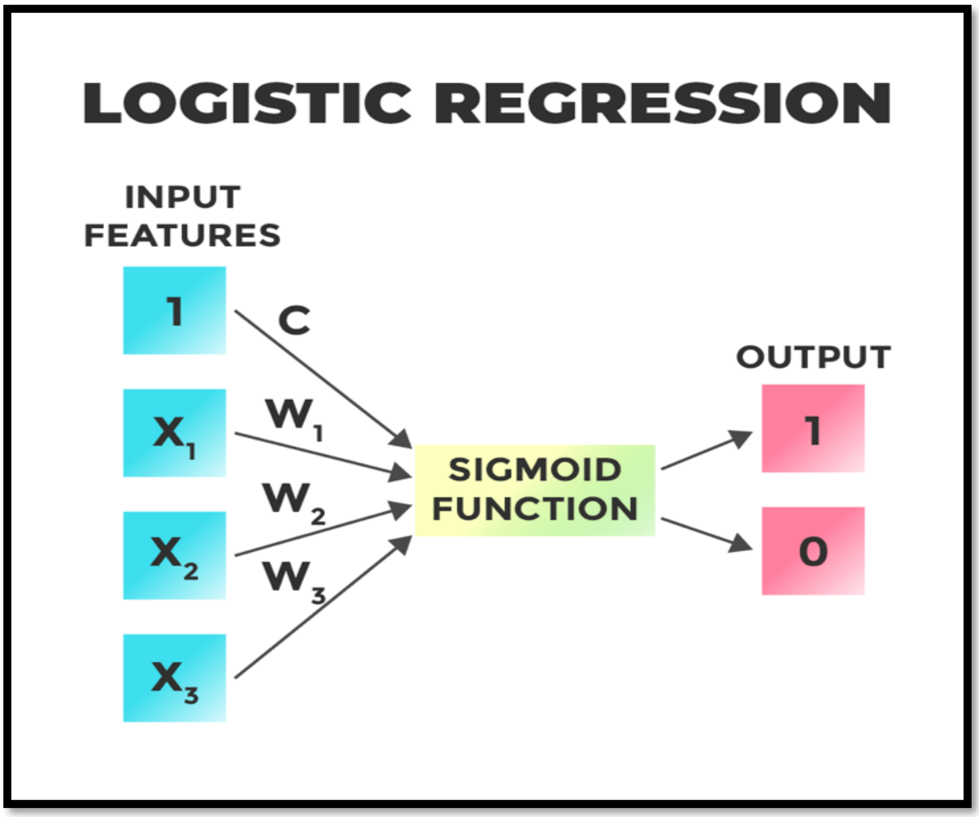


Figure 3. 5: Algorithm of LR

3.7.1.2 Support Vector Machine

Support Vector Machine (SVM) is a discriminative learning algorithm that identifies an optimal hyperplane to separate classes by maximizing the margin between the decision boundary and the closest data points (support vectors). In this study, a linear kernel is employed, which is particularly well suited for high-dimensional and sparse representations such as TF-IDF features, offering both computational efficiency and strong generalization performance. Since LinearSVC does not natively produce probability estimates, it is wrapped with CalibratedClassifierCV using 5-fold cross-validation during training. This procedure fits multiple auxiliary models to learn a mapping from raw decision scores to well-calibrated probability values, enabling reliable probability-based evaluation and threshold optimization. Figure 3.6 describes the structural architecture of SVM model.

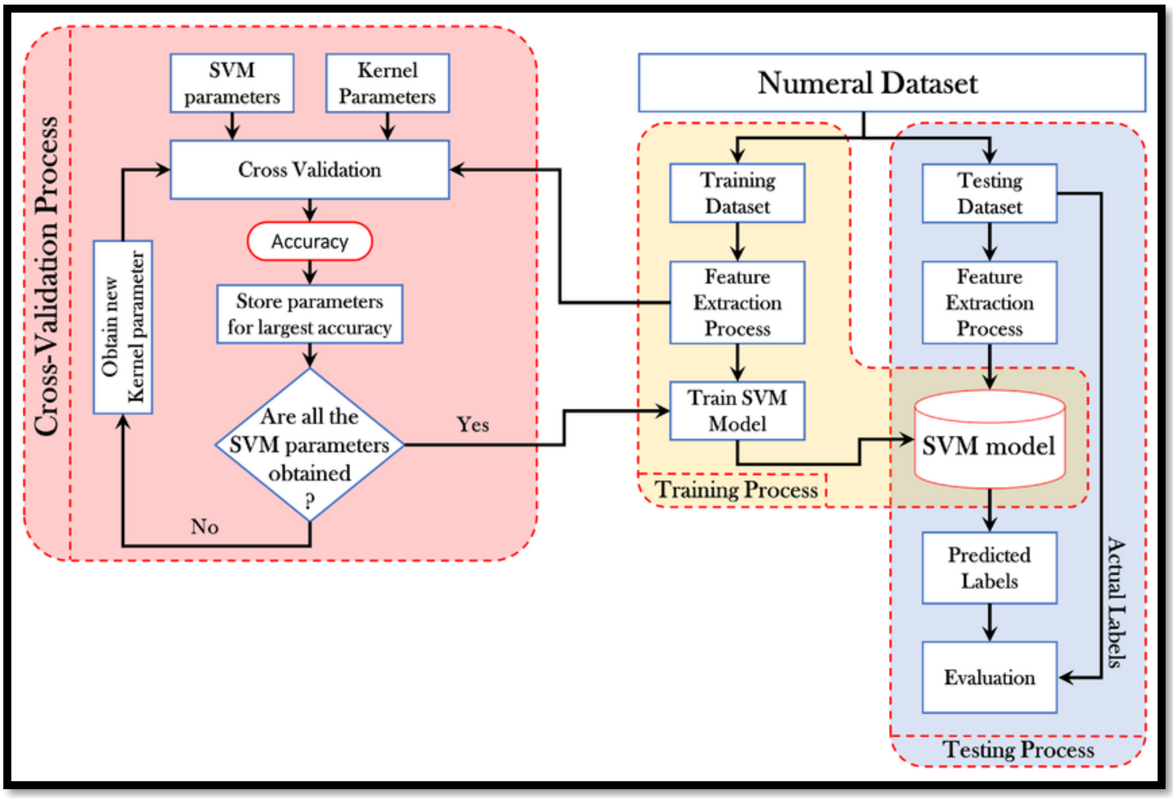


Figure 3. 6: Algorithm of SVM

3.7.1.3 Naive Bayes

Multinomial Naive Bayes is a probabilistic classification algorithm grounded in Bayes' theorem, which estimates the posterior class probability under the simplifying assumption that features are conditionally independent given the class label. The multinomial variant is particularly well suited for discrete frequency-based representations such as TF-IDF and word count vectors, as it models feature occurrences directly. Classification is performed by computing the proportional relationship P (class | features) ∝ P(class) × ∏ P (feature | class), enabling efficient and interpretable text classification. Probability calibration is necessary because Naive Bayes models tend to produce overconfident probability estimates, often pushing predictions unrealistically close to 0 or 1. To address this issue, Isotonic Regression is applied on the validation set to recalibrate the predicted probabilities without altering the underlying classifier. In this setup, the model is first trained on the training data and then calibrated on the validation data using `cv="prefit"`, ensuring a clean separation between training and calibration phases. Figure 3.7 describes the structural architecture of NB model.

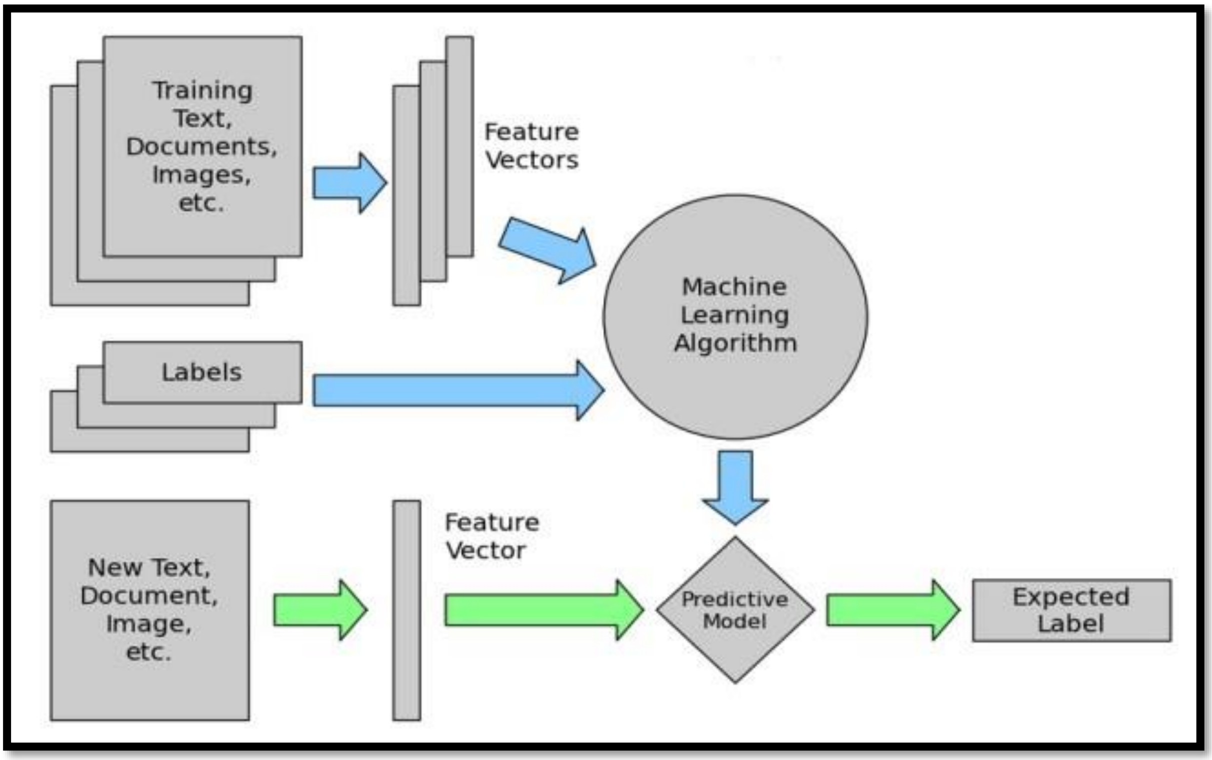

Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

Figure 3. 7: Algorithm of NB

3.7.1.4 ML ENSEMBLE

Soft Voting averages the predicted probabilities from all models—e.g., Final Probability = (P_LR + P_SVM + P_NB)/3—then applies an optimized threshold to determine the final class. Ensemble diversity leverages each model's strengths—LR captures linear patterns with regularization, SVM separates classes via margins and NB models probabilities under independence assumptions—so averaging their outputs reduces overfitting and variance. Figure 3.8 describes the structural architecture of MLENSEMBLE model.

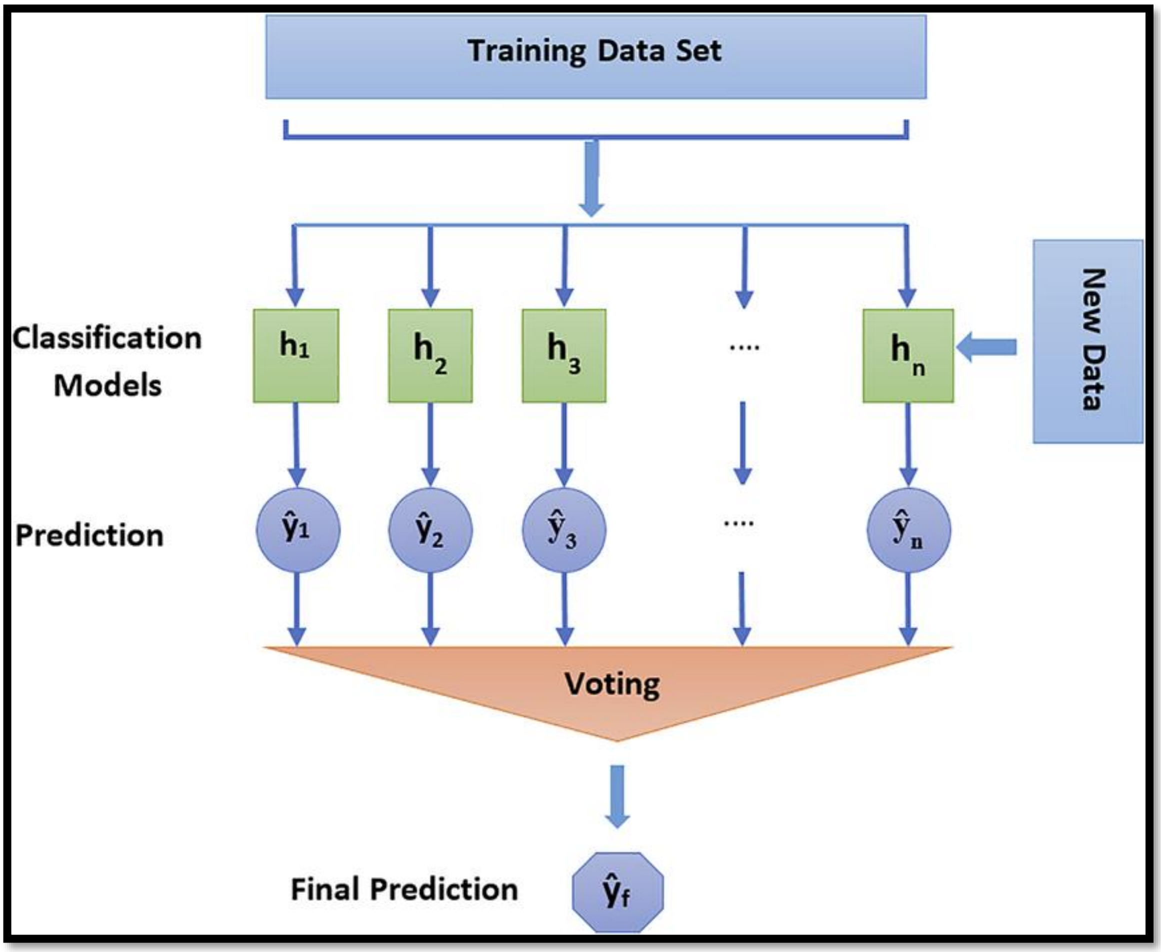


Figure 3. 8: Algorithm of ML ENSEMBLE

3.7.2 DL Models

3.7.2.1 Bidirectional Long Short-Term Memory

Bidirectional Long Short-Term Memory (BiLSTM) is a recurrent neural network architecture that processes sequential data in both forward and backward directions, enabling the model to capture contextual dependencies from past and future tokens simultaneously [10]. The LSTM component addresses the vanishing gradient problem inherent in traditional RNNs through a gated mechanism comprising three specialized gates: the forget gate, which selectively discards irrelevant information from the cell state; the input gate, which determines what new information to store; and the output gate, which controls the information flow to the next hidden state. In this study, the bidirectional configuration is employed to leverage the linguistic structure of Bengali idiom sentences, where both preceding and succeeding words provide critical context for correct usage classification. The model architecture begins with a trainable embedding layer that converts tokenized input sequences into dense 128-dimensional vectors, followed by the bidirectional LSTM layer with 128 hidden units in each direction. Dropout regularization at a rate of 0.45 is applied after the LSTM layer to mitigate overfitting, followed by a fully connected layer with 64 neurons and ReLU activation for feature extraction, an additional dropout layer and a final sigmoid output layer for binary classification. Figure 3.9 describes the structural architecture of BILSTM model.

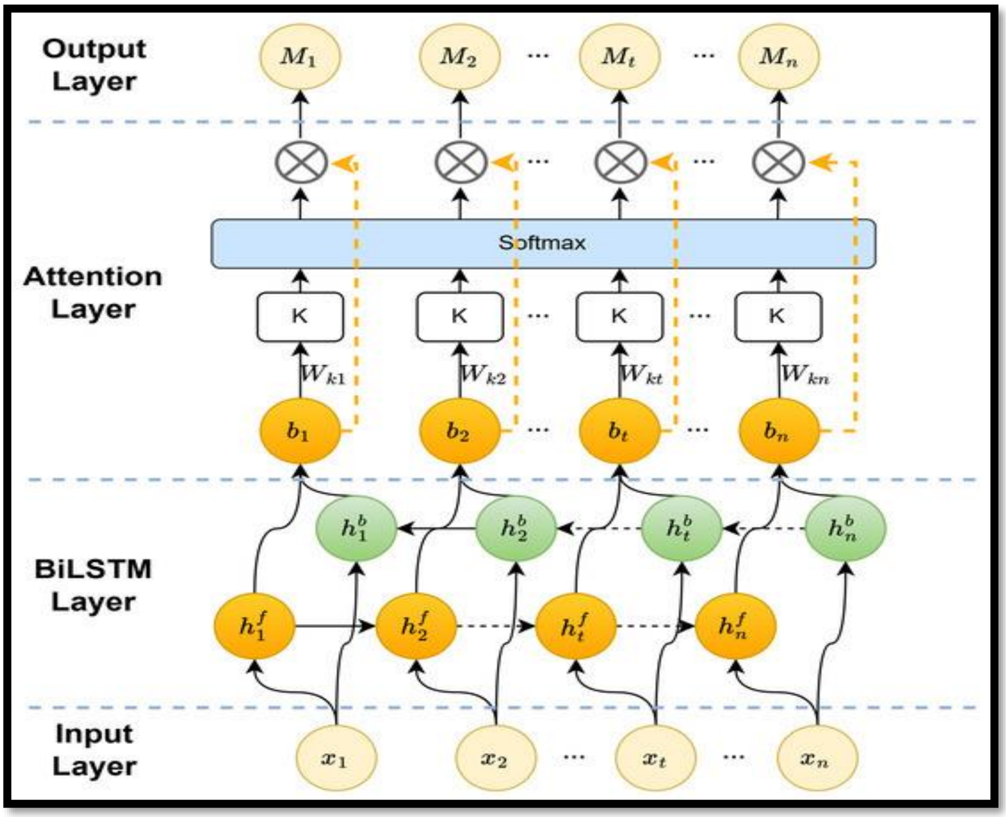


Figure 3. 9: Algorithm of BILSTM

3.7.2.2 BIDIRECTIONAL GATED RECURRENT UNIT

Bidirectional Gated Recurrent Unit (BiGRU) is a computationally efficient variant of the LSTM architecture that employs a simplified gating mechanism with only two gates—the reset gate and the update gate—thereby reducing the number of trainable parameters while maintaining comparable sequence modeling capabilities. The reset gate determines the extent to which previous hidden states should be disregarded when computing the candidate activation, while the update gate controls the balance between retaining information from the previous state and incorporating new candidate values. The bidirectional configuration processes input sequences in both temporal directions, capturing comprehensive contextual information essential for understanding Bengali idiomatic expressions where word order and surrounding context are semantically significant. This architecture demonstrates particular effectiveness in scenarios where computational resources are constrained or training data is limited, as the reduced parameter count mitigates the risk of overfitting. The model follows a similar architectural pattern to BiLSTM, with an embedding layer, bidirectional GRU with 128 units per direction, dropout regularization at 0.35, a dense intermediate layer with 64 neurons and a sigmoid output layer. Figure 3.10 describes the structural architecture of BIGRU model.

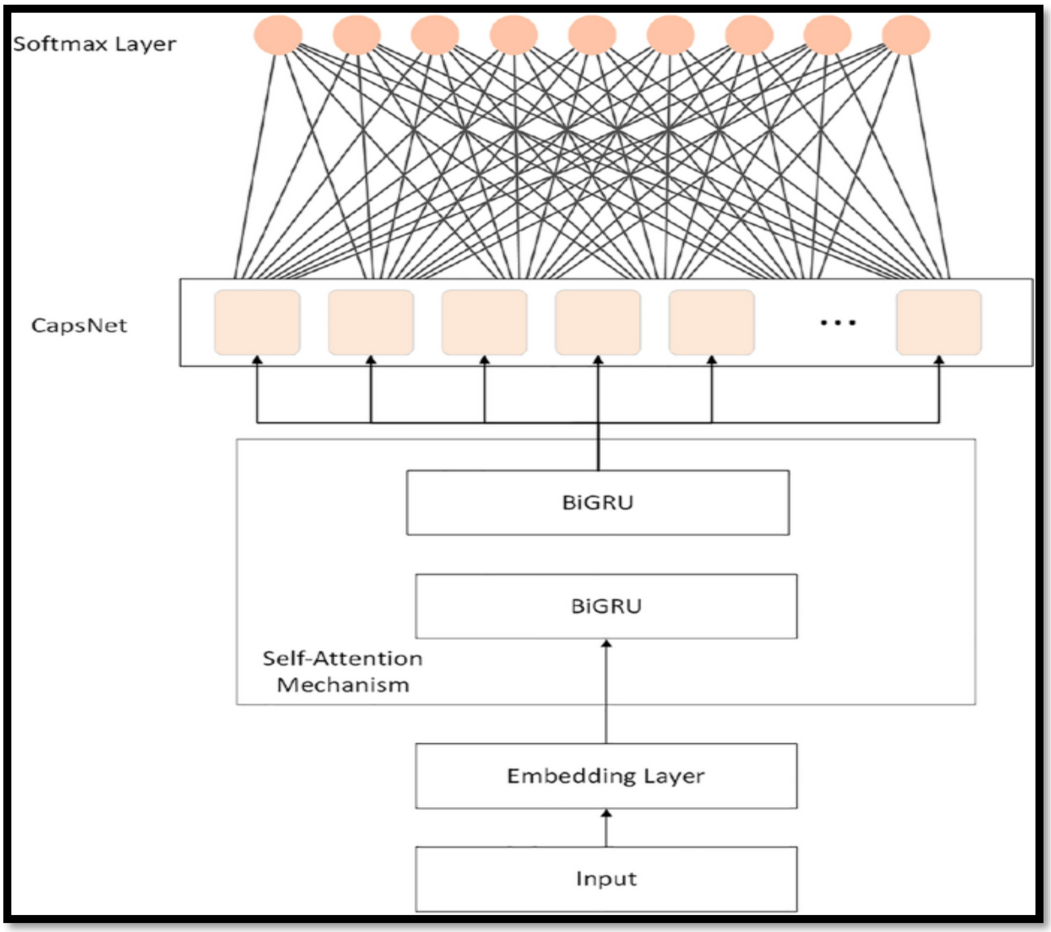


Figure 3. 10: Algorithm of BIGRU

3.7.2.3 Deep Neural Network

Deep Neural Network (DNN) is a feed-forward architecture that processes text representations through multiple fully connected layers without explicit modeling of sequential dependencies. Unlike recurrent architectures, this model treats input sequences as bags-of-embeddings by employing global average pooling over the temporal dimension, which aggregates token-level embeddings into a fixed-length vector representation. This pooling operation computes the element-wise mean across all token positions, effectively capturing distributional semantic information while remaining invariant to sequence length and word order. The resulting feature vector is then passed through a cascade of fully connected layers with progressively decreasing dimensionality—from 256 to 128 neurons—with ReLU activation functions and dropout regularization at 0.40 applied after each layer to prevent overfitting. While this architecture sacrifices explicit sequential modeling, it offers substantial computational advantages including faster training and inference times, reduced memory requirements and strong performance on tasks where global semantic content is more informative than precise word ordering, making it a suitable baseline for comparative analysis. Figure 3.11 describes the structural architecture of DNN model.

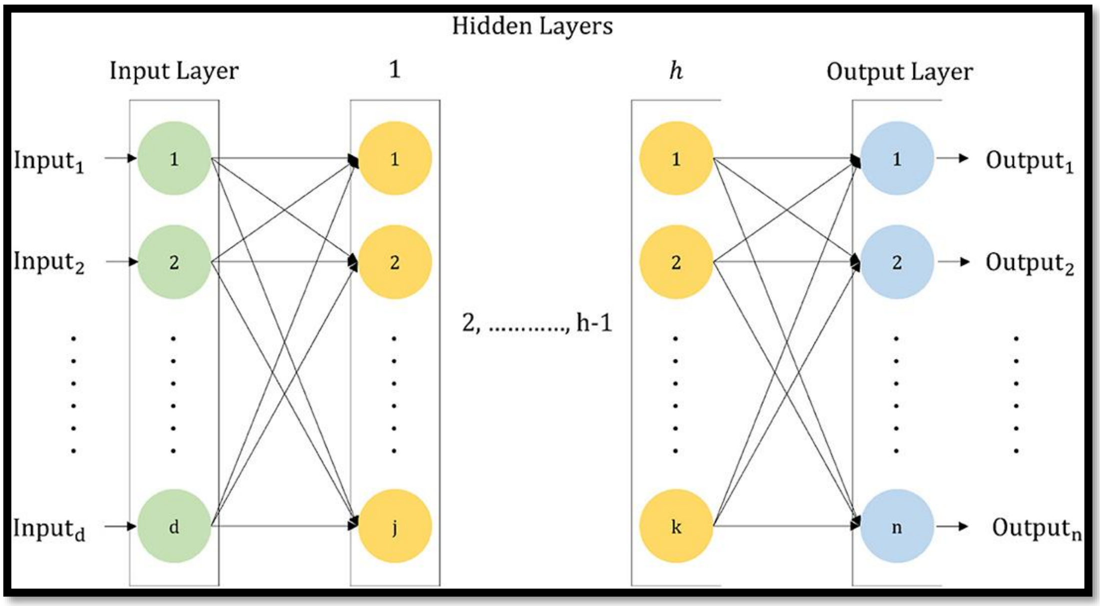


Figure 3. 11: Algorithm of DNN

3.7.2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) is a spatially-invariant architecture that applies learnable filters through one-dimensional convolution operations to detect local n-gram patterns and compositional features within sequential text data. The convolutional layer employs 128 filters with a kernel size of 5, enabling the model to identify salient 5-gram patterns such as common phrase constructions and idiomatic collocations that are characteristic of correct or incorrect usage. Unlike recurrent architectures that process sequences iteratively, CNNs compute activations for all temporal positions in parallel, offering significant computational efficiency advantages while maintaining the ability to capture local linguistic structures. The convolution operation applies same-padding to preserve sequence length, followed by global max pooling which extracts the maximum activation value from each filter across the entire sequence, thereby identifying the most discriminative pattern regardless of its position—a property particularly valuable for text classification where key phrases may appear at arbitrary locations. This max-pooled representation is subsequently processed through fully connected layers with dropout regularization, culminating in binary classification via a sigmoid activation function. Figure 3.12 describes the structural architecture of CNN model.

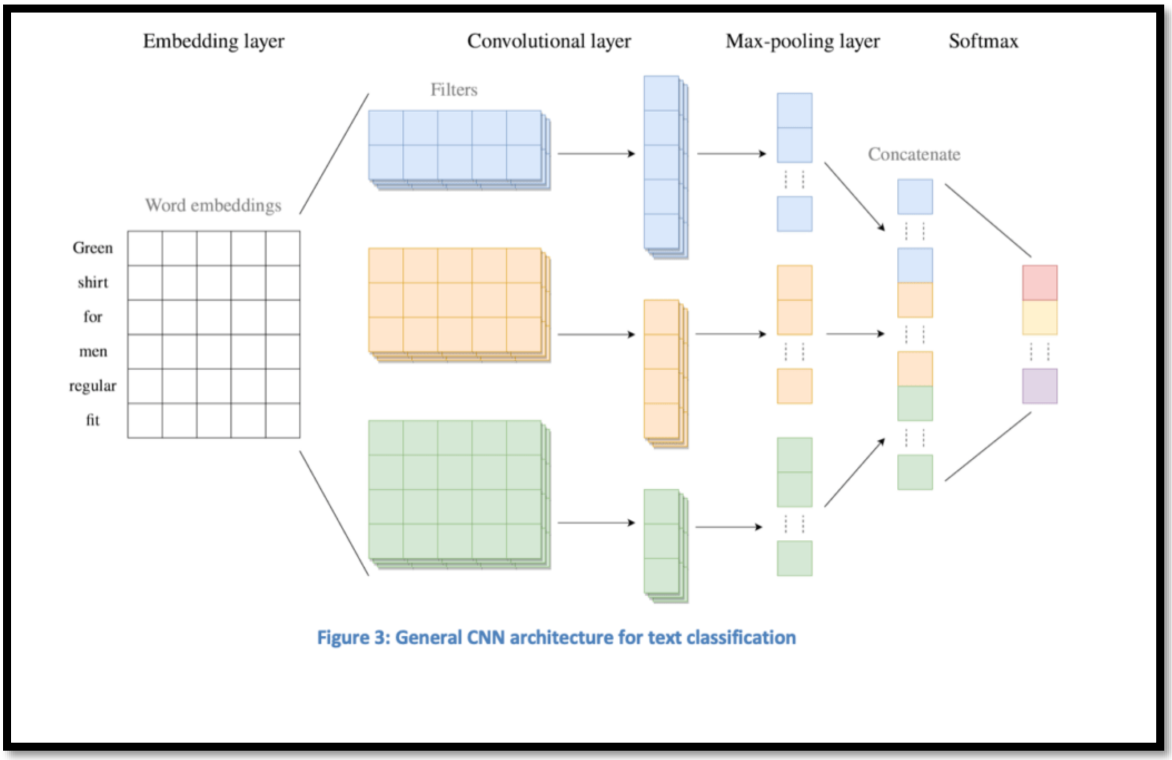


Figure 3. 12: Algorithm of CNN

3.7.2.5 DL Ensemble

Deep Learning Ensemble is a two-level stacked generalization framework that combines predictions from multiple heterogeneous base learners—BiLSTM, BiGRU, DNN and CNN—through a meta-learning approach to leverage their complementary strengths and mitigate individual model weaknesses. In the first stage, each base model is independently trained on the full training set to generate probability estimates, capturing different aspects of the input data: BiLSTM and BiGRU model long-range sequential dependencies through bidirectional recurrent processing, CNN identifies local n-gram patterns through convolutional filters and DNN captures global distributional semantics through averaged embeddings [4]. In the second stage, these base-level probability predictions are concatenated to form a four-dimensional meta-feature vector for each instance, which serves as input to a meta-learner—specifically, a Logistic Regression model with balanced class weights—that learns an optimal weighted combination of base predictions. This stacking approach enables the meta-model to dynamically select or blend base learners based on instance-specific characteristics, effectively performing learned model selection rather than fixed averaging. The ensemble architecture is trained using a holdout validation strategy where base models generate out-of-sample predictions on validation data to prevent overfitting during meta-model training, thereby ensuring robust generalization to unseen test instances. Figure 3.13 describes the structural architecture of DL ENSEMBLE model.

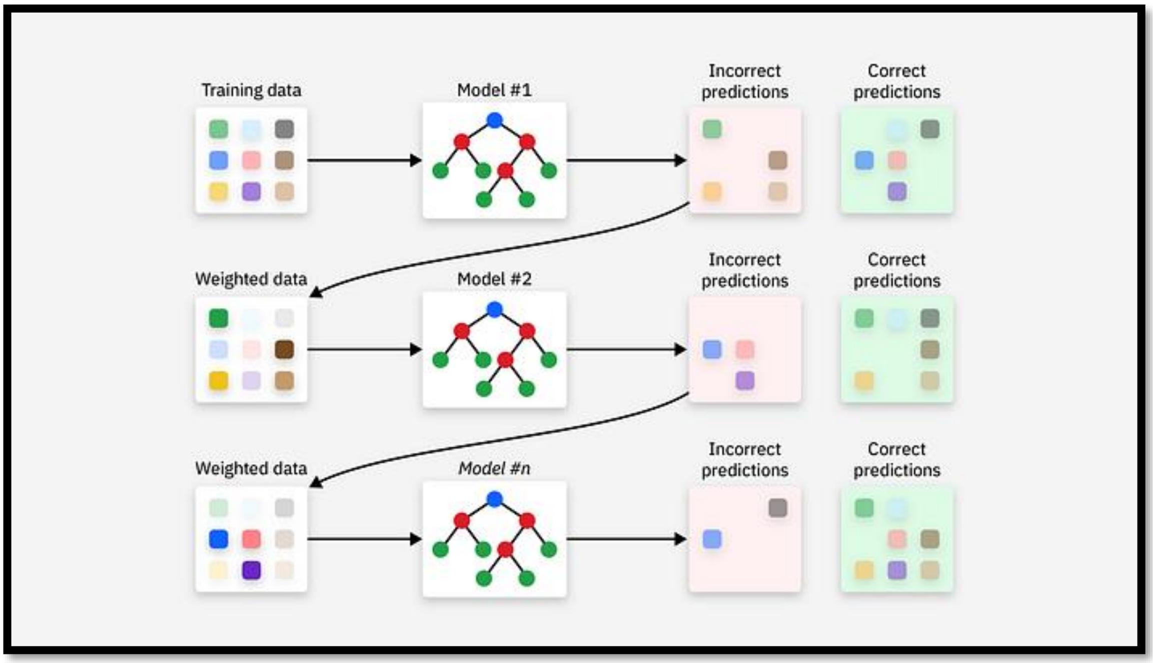


Figure 3. 13: Algorithm of DL ENSEMBLE

3.7.3 Transformer Based Models

3.7.3.1 BanglaBERT

BanglaBERT is a BERT-based language model specifically pre-trained on large-scale Bangla corpora, making it inherently attuned to the morphological, syntactic and semantic characteristics of the Bangla language. As a bidirectional transformer encoder, BanglaBERT processes input sequences through multi-head self-attention mechanisms that compute contextualized representations by attending to all tokens simultaneously, enabling the model to capture long-range dependencies and nuanced contextual relationships that are critical for understanding idiomatic expressions. The architecture consists of 12 transformer layers with 768-dimensional hidden states and 12 attention heads, providing a representational capacity of approximately 110 million parameters. For this classification task, the input is structured as a concatenated sequence containing three key components: the idiom phrase (বাগধারা), its literal meaning (অর্থ) and the contextual sentence (বাক্য), separated by Bangla punctuation marks to maintain linguistic coherence. This tripartite input format allows the model to jointly encode the idiomatic expression, its semantic interpretation and the surrounding context, facilitating cross-attention between these elements to determine usage correctness. The pre-trained BanglaBERT encoder is fine-tuned with a classification head consisting of a linear layer that projects the [CLS] token representation to binary logits, followed by softmax normalization for probabilistic prediction. The model employs a relatively conservative learning rate to preserve pre-trained knowledge while adapting to the specific task, with a sequence length of 192 tokens accommodating the multi-component input structure. Figure 3.14 describes the structural architecture of BanglaBERT model.

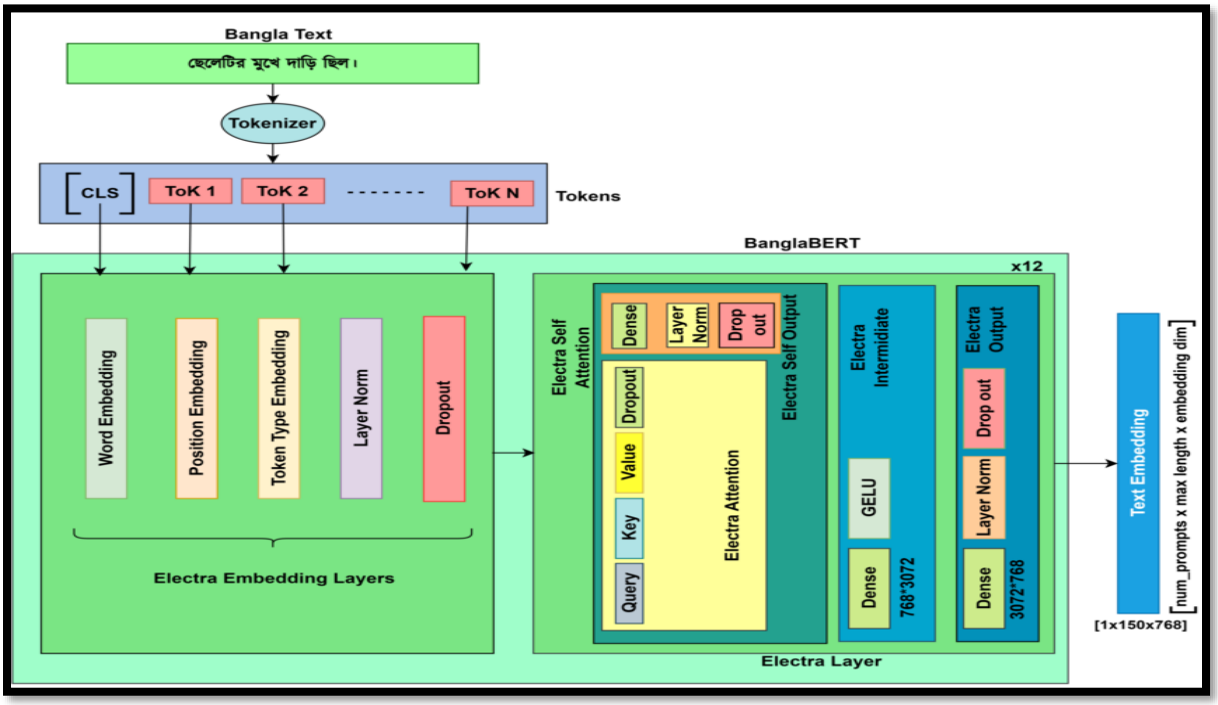


Figure 3. 14: Algorithm of BanglaBERT

3.7.3.2 MULTILINGUAL BERT

Multilingual BERT (mBERT) is a cross-lingual transformer model jointly pre-trained on Wikipedia corpora spanning 104 languages, including Bangla, using masked language modeling and next sentence prediction objectives. Unlike BanglaBERT's monolingual specialization, mBERT learns shared multilingual representations through a unified vocabulary of approximately 110,000 WordPiece tokens, enabling zero-shot transfer and cross-lingual semantic alignment across typologically diverse languages. The architecture mirrors the original BERT-base configuration with 12 transformer layers, 768-dimensional hidden representations and 12 attention heads, totaling approximately 178 million parameters. While mBERT's multilingual pre-training provides robustness and generalization capabilities, its distributed attention across multiple languages may result in reduced linguistic specificity for Bangla compared to language-specific models. For this idiom classification task, mBERT processes the same tripartite input structure as BanglaBERT, leveraging its cross-lingual semantic understanding to identify patterns of correct and incorrect idiomatic usage. The cased tokenizer preserves capitalization distinctions, which, although less relevant for Bangla script, maintains consistency with the pre-training regime. Fine-tuning proceeds identically to

BanglaBERT, with a linear classification head applied to the [CLS] token embedding and the model is trained using the same conservative hyperparameters to balance adaptation with retention of multilingual knowledge. Figure 3.15 describes the structural architecture of mBERT model.

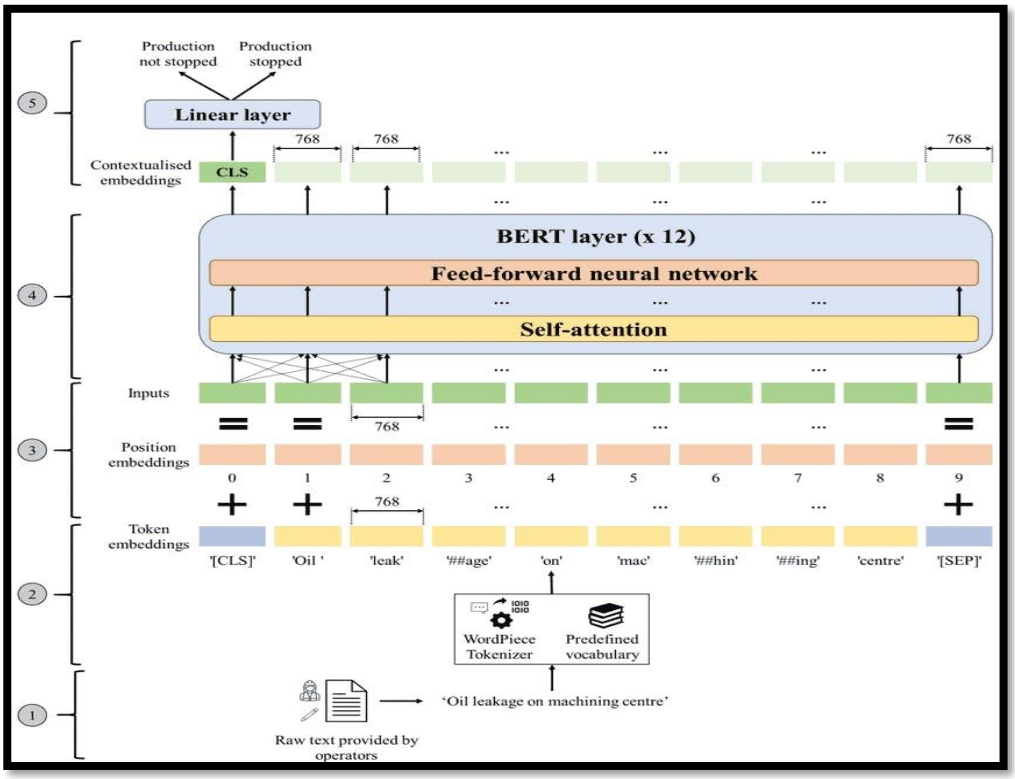


Figure 3. 15: Algorithm of mBERT

3.7.3.3 XLM-RoBERTa

XLM-RoBERTa (XLM-R) represents a significant advancement in multilingual language modeling, trained exclusively on 2.5 terabytes of CommonCrawl data spanning 100 languages using a self-supervised masked language modeling objective, without reliance on parallel corpora or translation pairs. Building upon the RoBERTa optimization strategy, XLM-R eliminates the next sentence prediction task, employs dynamic masking during pre-training and utilizes larger batch sizes and longer training schedules to achieve superior performance across diverse linguistic tasks. The base variant consists of 12 transformer layers with 768-dimensional hidden states and 8 attention heads, comprising approximately 270 million parameters—substantially larger than mBERT due to its extensive multilingual vocabulary of 250,000 SentencePiece tokens. This larger vocabulary provides finer-grained tokenization for morphologically rich languages like Bangla, reducing the frequency of unknown tokens and preserving semantic integrity in subword segmentation. XLM-R's pre-training on raw web text, rather than curated corpora like Wikipedia, exposes the model to more diverse linguistic registers, colloquial expressions and domain-specific language, potentially enhancing its ability to generalize to idiomatic constructions. For the idiom classification task, XLM-R processes the concatenated tripartite input through its transformer encoder, with the final [CLS] representation serving as the sentence-level embedding for binary classification [6]. The model's robust multilingual capabilities and improved tokenization make it particularly well-suited for handling the lexical and syntactic variability inherent in idiomatic language, while its RoBERTa-based training regimen ensures more stable and efficient fine-tuning dynamics. Figure 3.16 describes the structural architecture of XLM-sR model.

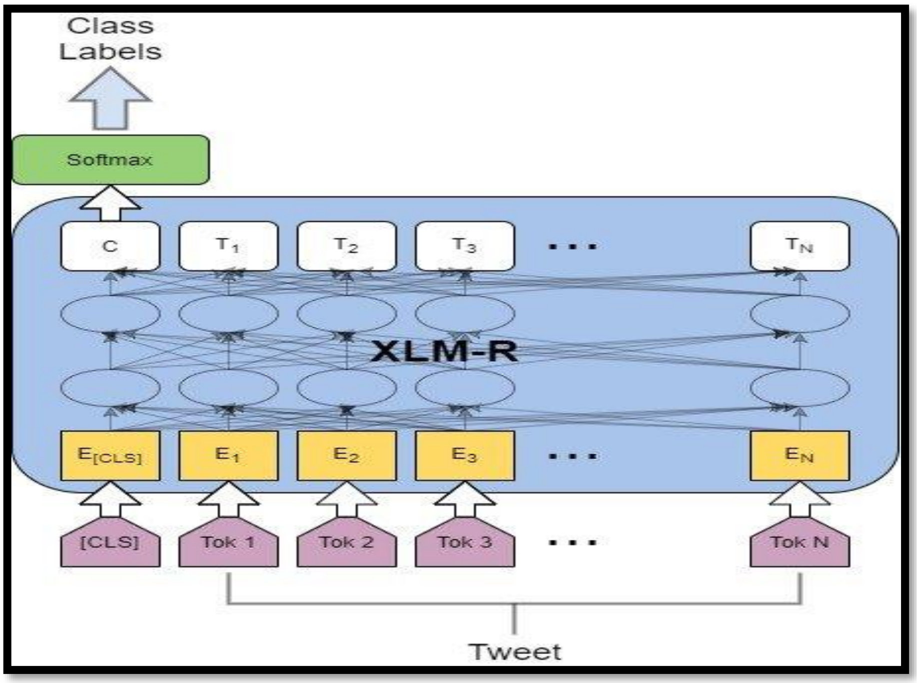


Figure 3. 16: Algorithm of XLM-R

3.7.4 PRETRAINED TRANSFORMER-BASED LANGUAGE MODEL

3.7.4.1 BANGLAT5

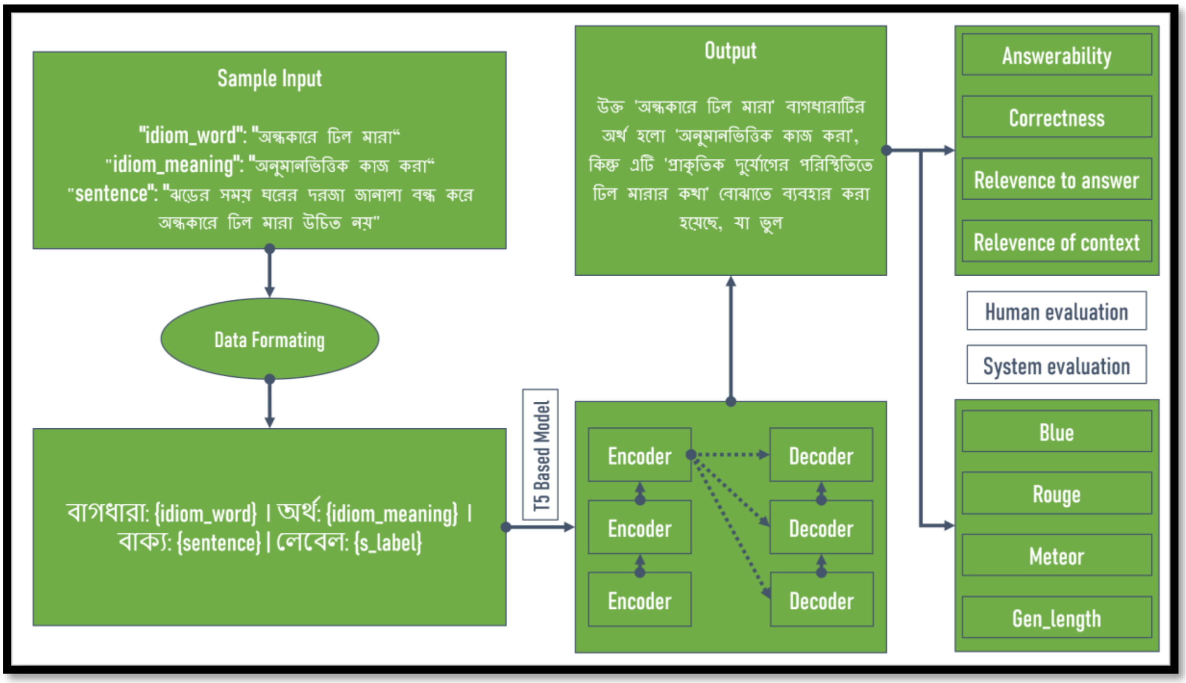
This work introduces a label-conditioned explanation generation framework that uses BanglaT5 to produce natural-language justifications (কারণ) for idiom usage correctness in Bangla sentences. Rather than simply classifying whether an idiom is used correctly or incorrectly, the model generates coherent reasoning that explains why a specific label (0 = incorrect, 1 = correct) applies, grounded in the idiom phrase (বাগধারা), its literal meaning (অর্থ) and the contextual sentence (বাক্য). BanglaT5, a T5-family encoder-decoder model pre-trained on Bangla text via span-denoising, consists of 12-layer encoder and decoder stacks with 768-dimensional hidden states (~247M parameters). It frames all tasks as text-to-text generation, enabling fluent, contextually rich output from structured prompts. The input prompt is deliberately designed to condition the generation on the label:

```
বাগধারা: {idiom} । অর্থ: {meaning} । বাক্য: {sentence}
লেবেল (s_label): {label}
কারণ লিখুন:
```

This transforms the task into conditional reasoning: P (reason | sentence, idiom, meaning, label), where the label forms part of the causal structure the explanation must justify.

To bridge the critical gap between training and real-world inference, we train on two parallel datasets that share identical target explanations but differ in the conditioning label. The "GOLD" dataset uses human-annotated ground-truth labels paired with human-written correct reasons — clean and accurate but limited in size and variety. The "PSEUDO" dataset replaces the input label with predictions from a strong upstream BanglaBERT classifier, while keeping the same human-written target reason. Although this means the model sometimes sees incorrect labels during training, it mirrors the true inference distribution: explanations are generated based on the classifier's predicted label, not the oracle truth. Training exclusively on "GOLD" creates exposure bias, often leading to generic or collapsed explanations when the classifier errs. By including "PSEUDO", the model learns to produce structurally valid, fluent justifications even under noisy label inputs — effectively performing teacher-student self-distillation for reasoning robustness. Training follows a deliberate two-stage curriculum. In Stage 1 (2 epochs on PSEUDO), the model acquires fluency, label-conditioned explanation patterns and tolerance to noisy labels, learning how justifications behave in realistic (imperfect) settings. In Stage 2 (2 epochs on GOLD), it refines factual alignment, corrects semantic drift or hallucinations and anchors explanations to ground-truth semantics. Continuous fine-tuning (no weight reset) ensures smooth knowledge transfer without catastrophic forgetting. Reversing the order (GOLD first) typically harms performance by causing early overfitting to clean data and poor adaptation to noise later

This design is especially effective for Bangla idiom reasoning due to three factors: (1) explanations follow highly reusable linguistic patterns (causal connectors, contrastive structures, evaluative phrases), which PSEUDO massively amplifies without new annotations; (2) gold explanations are scarce, but label-conditioned templates are abundant, allowing PSEUDO to scale coverage efficiently; and (3) the task is justification generation rather than truth recovery, making training on pseudo-labels conceptually valid — the model learns to explain beliefs, even imperfect ones, aligning with explainable AI goals. Evaluation uses only the GOLD test set, measuring ROUGE-L (structural overlap) and BLEU-4 (phrasing fidelity) to assess both content faithfulness and linguistic quality. In summary, the proposed curriculum-style training — first exposing BanglaT5 to realistic pseudo-label noise to build robust explanation patterns, then refining with gold data for semantic precision — effectively closes the train-inference distribution gap, leverages Bangla's pattern-rich explanatory style and delivers high-quality, faithful and fluent justifications for idiom usage, outperforming gold-only baselines and advancing explainable NLP in this low-resource setting. In figure 3.17 describes the structural architecture of BanglaT5 reasoning model is presented.



Sample Input
"idiom_word": "অন্ধকারে ঢিল মারা“
"idiom_meaning": "অনুমানভিত্তিক কাজ করা“
"sentence": "ঝড়ের সময় ঘরের দরজা জানালা বন্ধ করে অন্ধকারে ঢিল মারা উচিত নয়"
Data Formating
বাগধারা: {idiom_word} । অর্থ: {idiom_meaning} । বাক্য: {sentence} | লেবেল: {s_label}
T5 Based Model
Encoder
Encoder
Encoder
Decoder
Decoder
Decoder
Output
উক্ত 'অন্ধকারে ঢিল মারা' বাগধারাটির অর্থ হলো 'অনুমানভিত্তিক কাজ করা', কিন্তু এটি 'প্রাকৃতিক দুর্যোগের পরিস্থিতিতে ঢিল মারার কথা' বোঝাতে ব্যবহার করা হয়েছে, যা ভুল
Answerability
Correctness
Relevence to answer
Relevence of context
Human evaluation
System evaluation
Blue
Rouge
Meteor
Gen_length

Figure 3. 17: Algorithm of BanglaT5

CHAPTER 4
HARDWARE AND TOOLKITS

4.1 HARDWARE INFRASTRUCTURE

All computational experiments were executed using Google Collaboratory (Colab), a cloud-based platform providing free access to GPU resources and pre-configured machine learning environments. The platform's integration with Google Drive enabled persistent storage of datasets, trained models and experimental results. For deep learning and transformer model training, experiments utilized NVIDIA Tesla T4 GPU with 16 GB GDDR6 memory and 2,560 CUDA cores. This configuration provided sufficient computational power for training neural architectures while accommodating transformer models with appropriate batch sizes. GPU acceleration was essential for transformer fine-tuning and deep learning models, reducing training times from days to hours. For traditional machine learning models, CPU resources consisting of Intel Xeon processors with 2 virtual cores and 12.7 GB RAM were employed. These specifications proved adequate for training logistic regression, SVM and Naive Bayes classifiers on TF-IDF feature representations. Dataset storage utilized Google Drive with a hierarchical directory structure organizing raw data, preprocessed features, trained models and evaluation results. The total storage requirement was approximately 3 GB for complete experimental artifacts including all trained models and result files.

4.2 SOFTWARE FRAMEWORKS & LIBRARIES

Python 3.10 served as the primary programming language, leveraging its extensive ecosystem of scientific computing and machine learning libraries. Jupyter Notebooks provided the interactive development environment through Colab's browser-based interface, supporting reproducible research through combined code execution and documentation.

NumPy 1.23.5 provided fundamental support for numerical array operations and mathematical functions, serving as the backbone for data manipulation throughout the pipeline.

Pandas 1.5.3 facilitated structured data handling with DataFrame operations for loading the JSONL dataset, computing statistics and organizing experimental results.

SciPy 1.10.1 provided sparse matrix representations essential for efficient storage and computation with high-dimensional TF-IDF feature matrices, reducing memory consumption by orders of magnitude.

CHAPTER 5
EXPERIMENTAL RESULT

This chapter presents comprehensive experimental results evaluating the performance of twelve classification models and one reasoning generation model for Bangla idiom usage detection and explanation. The experiments were designed to address two primary research objectives: first, to identify the most effective neural architecture for binary classification of idiom usage correctness and second, to demonstrate feasibility of automated reasoning generation that provides pedagogically valuable explanations for classification decisions.

The experimental evaluation proceeds through three distinct phases corresponding to different modeling paradigms. The machine learning phase evaluates traditional approaches including Logistic Regression, Support Vector Machines, Naive Bayes and their ensemble combination, all operating on TF-IDF feature representations. The deep learning phase assesses recurrent and convolutional architectures including BiLSTM, BiGRU, CNN, DNN and a stacked ensemble combining predictions from all four base models. The transformer phase examines pre-trained language models including BanglaBERT, multilingual BERT and XLM-RoBERTa, fine-tuned specifically for idiom usage validation. Finally, the reasoning generation phase evaluates BanglaT5's ability to produce coherent explanations justifying usage correctness judgments.

All models were evaluated using consistent metrics including accuracy, precision, recall, F1-score, ROC-AUC (Area Under Receiver Operating Characteristic Curve) and average precision. Confusion matrices visualize classification patterns, while classification reports provide detailed per-class performance breakdowns. Comparative analysis identifies the best-performing architecture and analyzes performance differences across modeling paradigms. The reasoning generation component is evaluated using ROUGE-L and BLEU-4 metrics alongside qualitative analysis of generated explanations.

Results demonstrate that transformer-based models, particularly BanglaBERT, achieve superior performance over traditional and deep learning approaches, with the best model reaching **94% F1-score** and **94% accuracy** on the test set. The reasoning generation system produces explanations with **ROUGE-L F1-score of 90%** and **BLEU-4 score of 82%,** indicating strong semantic alignment with human-written justifications. These findings validate the research hypothesis that modern neural approaches can effectively automate idiom usage validation while providing interpretable explanations suitable for language learning applications.

5.1 Results

We evaluate ML, DL and Transformer models by presenting confusion matrices to show prediction errors and ROC curves to assess class separation and also to quantify overall performance.

5.1.1 ML Results

5.1.1.1 Logistic Regression

Figure 5.1 presents the performance evaluation of a Logistic Regression (LR) model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix illustrates the model's predictions against true labels: for true label 0, it correctly predicted 888 instances (true negatives) but misclassified 287 as 1 (false positives) and for true label 1, it correctly identified 1037 instances (true positives) while misclassifying 139 as 0 (false negatives), resulting in an overall accuracy of approximately 81.9% ((888 + 1037) / (888 + 287 + 139 + 1037) = 1925 / 2351 ≈ 0.819). Additional metrics derived from the matrix include precision of about 78.3% (1037 / (1037 + 287)), recall of about 88.2% (1037 / (1037 + 139)) and an F1 score of approximately 82.9% (2 * (0.783 * 0.882) / (0.783 + 0.882)), indicating a balanced performance with stronger recall than precision and potential for reducing false positives to improve overall effectiveness. On the right, the ROC curves compare the model's ability to distinguish between classes: the blue train curve achieves an AUC of 0.95, reflecting excellent discriminative power on the training data, while the orange validation curve has an AUC of 0.91, suggesting good generalization; the AUC gap of 0.04 (0.95 - 0.91) is under 0.06, indicating a good fit with minimal overfitting, as both curves rise sharply toward the top-left corner before approaching the random chance diagonal line.

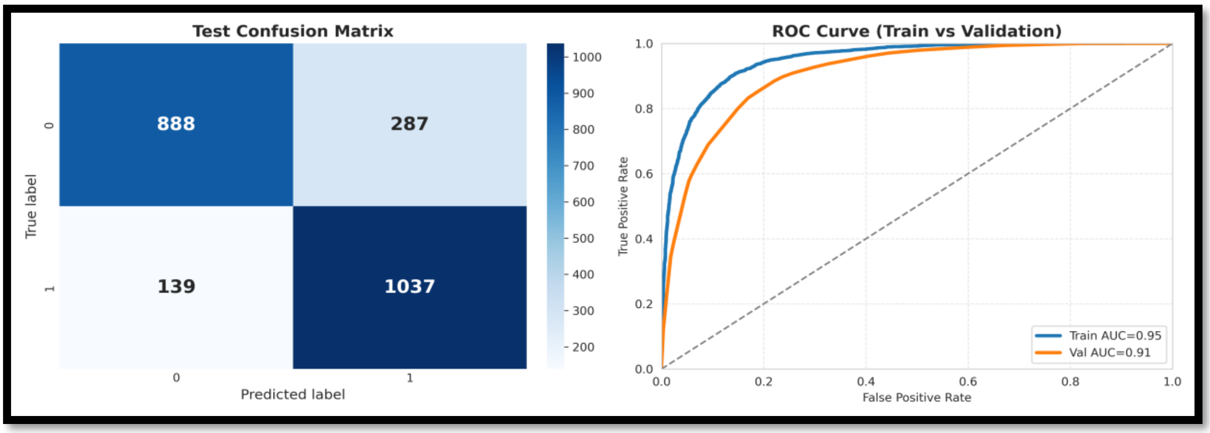


Figure 5. 1: Confusion Matrix & Roc Curve of LR

5.1.1.2 Naive Bayes

Figure 5.2 presents the performance evaluation of a Naive Bayes (NB) model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix illustrates the model's predictions against true labels: for true label 0, it correctly predicted 870 instances (true negatives) but misclassified 305 as 1 (false positives); for true label 1, it correctly identified 972 instances (true positives) while misclassifying 204 as 0 (false negatives), resulting in an overall accuracy of approximately 78.4% (calculated as (870 + 972) / (870 + 305 + 204 + 972) = 1842 / 2351 ≈ 0.784). Additional metrics derived from the matrix include precision of about 76.1% (972 / (972 + 305)), recall of about 82.7% (972 / (972 + 204)) and an F1 score of approximately 79.2% (2 * (0.761 * 0.827) / (0.761 + 0.827)), indicating a balanced performance with stronger recall than precision and potential for reducing false positives to improve overall effectiveness. On the right, the ROC curves compare the model's ability to distinguish between classes: the blue train curve achieves an AUC of 0.90, reflecting strong discriminative power on the training data, while the orange validation curve has an AUC of 0.88, suggesting good generalization; the AUC gap of 0.02 (0.90 - 0.88) is under 0.06, indicating a good fit with minimal overfitting, as both curves rise sharply toward the top-left corner before approaching the random chance diagonal line.

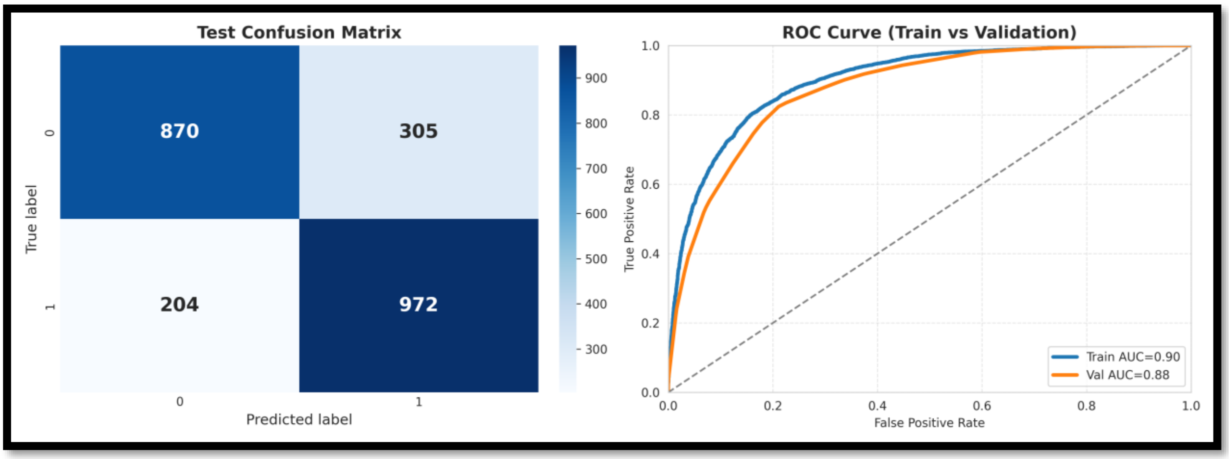


Figure 5. 2: Confusion Matrix & Roc Curve of NB

5.1.1.3 Support Vector Machine

Figure 5.3 presents the performance evaluation of a Support Vector Machine (SVM) model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix illustrates the model's predictions against true labels: for true label 0, it correctly predicted 875 instances (true negatives) but misclassified

300 as 1 (false positives); for true label 1, it correctly identified 1058 instances (true positives) while misclassifying 118 as 0 (false negatives), resulting in an overall accuracy of approximately 82.2% (calculated as (875 + 1058) / (875 + 300 + 118 + 1058) = 1933 / 2351 ≈ 0.822). Additional metrics derived from the matrix include precision of about 77.9% (1058 / (1058 + 300)), recall of about 90.0% (1058 / (1058 + 118)) and an F1 score of approximately 83.5% (2 * (0.779 * 0.900) / (0.779 + 0.900)), indicating a balanced performance with stronger recall than precision and potential for reducing false positives to improve overall effectiveness. On the right, the ROC curves compare the model's ability to distinguish between classes: the blue train curve achieves an AUC of 0.96, reflecting excellent discriminative power on the training data, while the orange validation curve has an AUC of 0.91, suggesting good generalization; the AUC gap of 0.05 (0.96 - 0.91) is under 0.06, indicating a good fit with minimal overfitting, as both curves rise sharply toward the top-left corner before approaching the random chance diagonal line.

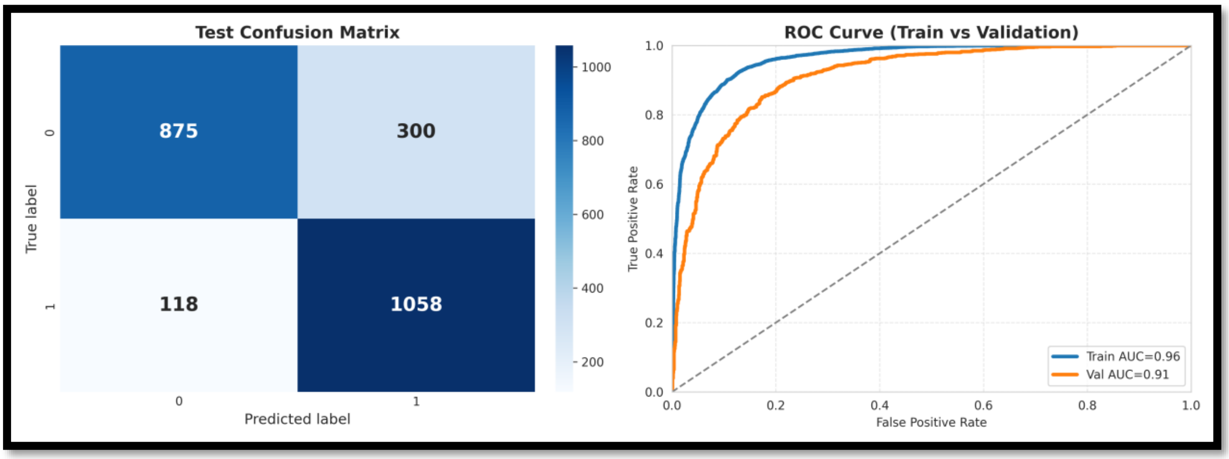


Figure 5. 3: Confusion Matrix & Roc Curve of SVM

5.1.1.4 ML ENSEMBLE

Figure 5.4 presents the performance evaluation of an ML Ensemble model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix illustrates the model's predictions against true labels: for true label 0, it correctly predicted 919 instances (true negatives) but misclassified 256 as 1 (false positives); for true label 1, it correctly identified 1021 instances (true positives) while misclassifying 155 as 0 (false negatives), resulting in an overall accuracy of approximately 82.5% (calculated as (919 + 1021) / (919 + 256 + 155 + 1021) = 1940 / 2351 ≈ 0.825). Additional metrics derived from the matrix include precision of about 80.0% (1021 / (1021 + 256)), recall of about 86.8% (1021 / (1021 + 155)) and an F1 score of approximately 83.2% (2 * (0.800 * 0.868) / (0.800 + 0.868)), indicating solid balanced performance with good recall (though slightly lower than some high-sensitivity models) and improved precision compared to models that favor recall more aggressively, suggesting fewer false positives overall and better suitability for applications where false alarms carry higher costs. On the right, the ROC curves compare the model's discriminative ability: the blue train curve achieves an AUC of 0.95, reflecting very strong performance on the training data, while the orange validation curve has an AUC of 0.91, indicating good generalization to unseen data; the AUC gap of 0.04 (0.95 - 0.91) is small (well under 0.06), pointing to minimal overfitting and a robust fit. Both curves rise sharply toward the top-left corner early on, staying well above the random chance diagonal line, confirming the ensemble's effective class separation across thresholds.

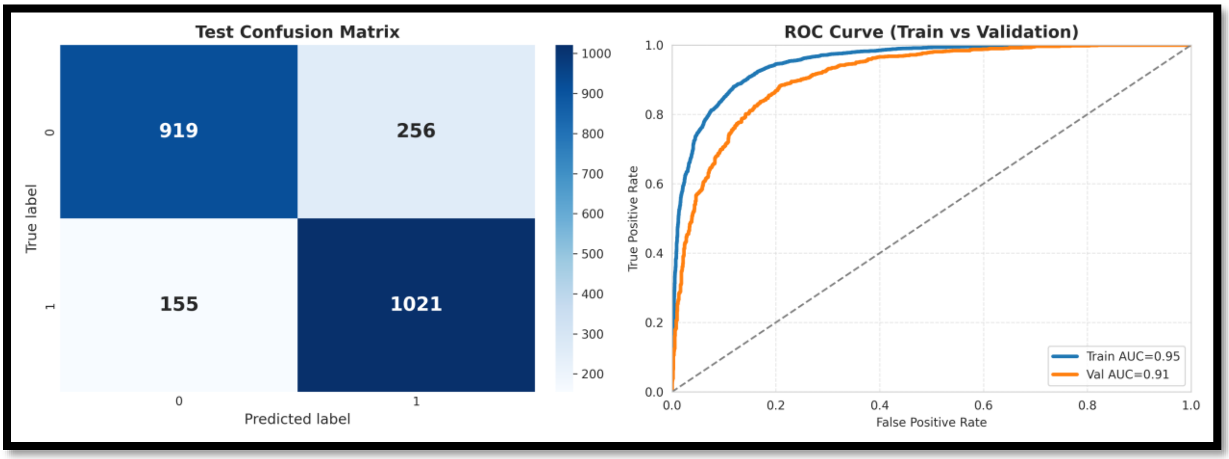


Figure 5. 4: Confusion Matrix & Roc Curve of ML ENSEMBLE

5.1.2 DL Results

5.1.2.1 Bidirectional Gated Recurrent Unit

Figure 5.5 presents the performance evaluation of a BiGRU model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix shows that for true label 0, the model correctly predicted 888 instances (true negatives) but misclassified 287 as 1 (false positives); for true label 1, it correctly identified 1024 instances (true positives) while missing 152 as 0 (false negatives), yielding an overall accuracy of approximately 81.3% ((888 + 1024) / (888 + 287 + 152 + 1024) = 1912 / 2351 ≈ 0.813), with precision ≈ 78.1% (1024 / (1024 + 287)), recall ≈ 87.1% (1024 / (1024 + 152)) and F1 score ≈ 82.4% (2 × (0.781 × 0.871) / (0.781 + 0.871)); this reflects strong recall (prioritizing detection of positives) at the cost of a moderate number of false positives, resulting in a balanced but slightly lower precision than some ensemble approaches. On the right, the ROC curves demonstrate excellent discriminative power: the blue train curve reaches an AUC of 0.98, indicating near-perfect separation on training data, while the orange validation curve achieves an AUC of 0.90, still very good for generalization; the small AUC gap of 0.08 (0.98 – 0.90) suggests acceptable but noticeable overfitting compared to previous models, though both curves rise sharply toward the top-left corner and remain well above the random chance diagonal, confirming robust class separation overall.

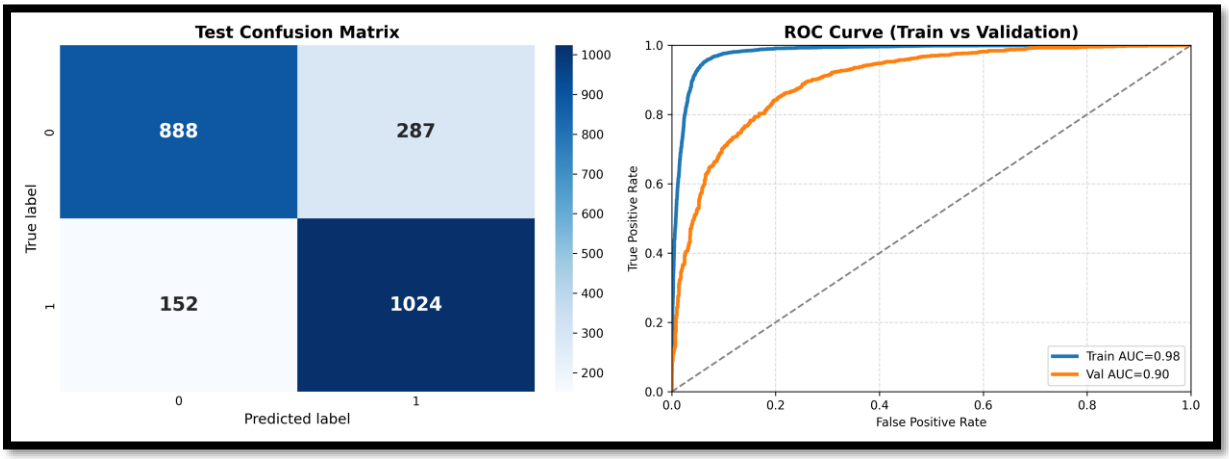


Figure 5. 5: Confusion Matrix & Roc Curve of BIGRU

5.1.2.2 Bidirectional Long Short-Term Memory

Figure 5.6 presents the performance evaluation of a BiLSTM model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix shows that for true label 0, the model correctly predicted 928 instances (true negatives) but misclassified 247 as 1 (false positives); for true label 1, it correctly identified 995 instances (true positives) while missing 181 as 0 (false negatives), yielding an overall accuracy of approximately 81.8% ((928 + 995) / (928 + 247 + 181 + 995) = 1923 / 2351 ≈ 0.818), with precision ≈ 80.1% (995 / (995 + 247)), recall ≈ 84.6% (995 / (995 + 181)) and F1 score ≈ 82.3% (2 × (0.801 × 0.846) / (0.801 + 0.846)); this indicates a reasonably balanced trade-off with solid recall (strong ability to detect positives) and improved precision over some prior models, though false positives remain noticeable. On the right, the ROC curves highlight excellent discriminative capability: the blue train curve achieves an AUC of 0.97, reflecting very high performance on the training data, while the orange validation curve reaches an AUC of 0.90, still good for unseen data; the AUC gap of 0.07 (0.97 – 0.90) suggests moderate overfitting—higher than the ensemble's but comparable to or slightly better than the BiGRU—yet both curves rise sharply toward the top-left corner early and stay well above the random chance diagonal, confirming effective class separation overall. In summary, the BiLSTM model provides strong sensitivity, competitive balanced metrics and good (though not the tightest) generalization, making it a solid performer for applications prioritizing true positive detection with acceptable tolerance for some false alarms.

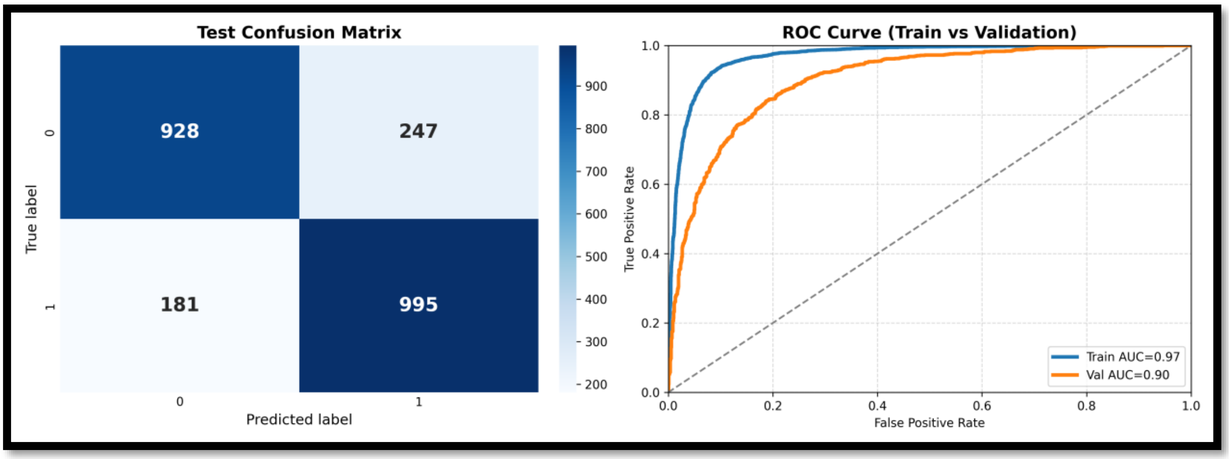


Figure 5. 6: Confusion Matrix & Roc Curve of BILSTM

5.1.2.3 Convolutional Neural Network

Figure 5.7 presents the performance evaluation of a CNN model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix shows that for true label 0, the model correctly predicted 933 instances (true negatives) but misclassified 242 as 1 (false positives); for true label 1, it correctly identified 999 instances (true positives) while missing 177 as 0 (false negatives), yielding an overall accuracy of approximately 82.2% ((933 + 999) / (933 + 242 + 177 + 999) = 1932 / 2351 ≈ 0.822), with precision ≈ 80.5% (999 / (999 + 242)), recall ≈ 84.9% (999 / (999 + 177)) and F1 score ≈ 82.6% (2 × (0.805 × 0.849) / (0.805 + 0.849)); this demonstrates a well-balanced performance with strong recall (effective capture of most positives) and notably improved precision relative to several previous models, indicating fewer unnecessary false alarms while maintaining good sensitivity. On the right, the ROC curves exhibit outstanding discriminative power: the blue train curve achieves an AUC of 0.99, reflecting near-perfect separation on the training data, while the orange validation curve reaches an AUC of 0.91, still very strong for generalization; the AUC gap of 0.08 (0.99 – 0.91) points to moderate overfitting—higher than the ensemble but similar to the BiGRU and BiLSTM—yet both curves rise very sharply toward the top-left corner immediately after the origin and remain substantially above the random chance diagonal throughout, confirming excellent overall class separation capability. In summary, the CNN model delivers highly competitive balanced metrics, very high training performance and solid (though not the most overfitting-resistant) generalization, making it one of the stronger contenders among the evaluated models, particularly suitable for tasks requiring both reliable positive detection and reasonable control over false positives.

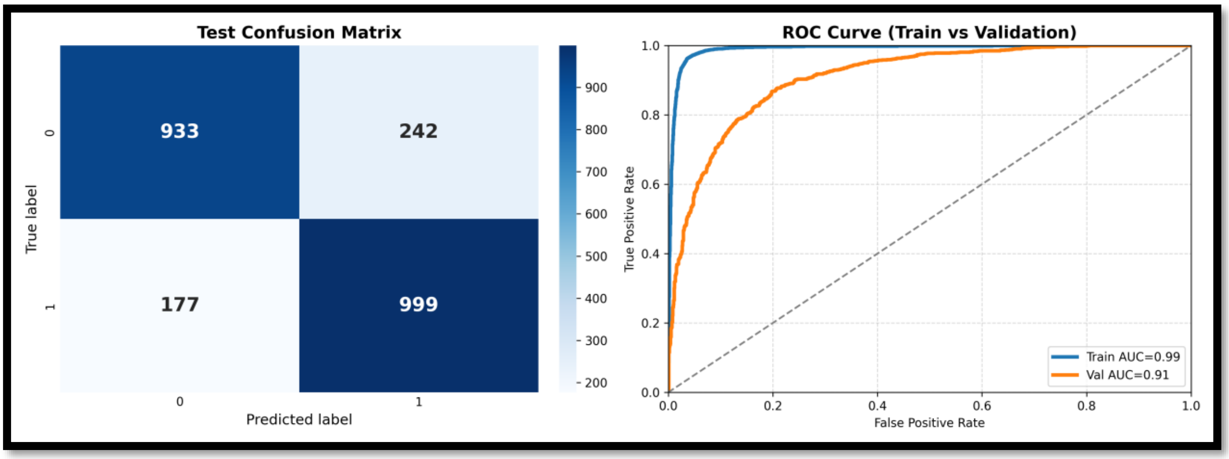


Figure 5. 7: Confusion Matrix & Roc Curve of CNN

5.1.2.4 DEEP NEURAL NETWORK

Figure 5.8 presents the performance evaluation of a DNN model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix shows that for true label 0, the model correctly predicted 719 instances (true negatives) but misclassified a relatively high 456 as 1 (false positives); for true label 1, it correctly identified 1046 instances (true positives) while missing only 130 as 0 (false negatives), yielding an overall accuracy of approximately 75.1% ((719 + 1046) / (719 + 456 + 130 + 1046) = 1765 / 2351 ≈ 0.751), with precision ≈ 69.6% (1046 / (1046 + 456)), recall ≈ 89.0% (1046 / (1046 + 130)) and F1 score ≈ 78.3% (2 × (0.696 × 0.890) / (0.696 + 0.890)); this reveals a clear bias toward high recall (very strong ability to detect most positives, with few missed cases) at the expense of lower precision and a notably higher number of false positives compared to previous models, resulting in reduced overall accuracy and making it less suitable for scenarios where false alarms are costly. On the right, the ROC curves indicate good but not top-tier discriminative power: the blue train curve achieves an AUC of 0.94, showing strong performance on the training data, while the orange validation curve drops to an AUC of 0.85, reflecting only moderate generalization to unseen data; the AUC gap of 0.09 (0.94 – 0.85) is the largest observed so far, signaling more pronounced overfitting than in the CNN, BiLSTM, BiGRU, or especially the ensemble models, although both curves still rise reasonably toward the top-left corner and stay above the random chance diagonal. In summary, the DNN model excels at maximizing true positive detection (highest recall among the evaluated models) but suffers from the highest false positive rate, lowest accuracy, lowest precision and clearest signs of overfitting with poorer generalization, positioning it as the weakest performer in balanced effectiveness and robustness among the compared architectures for this task.

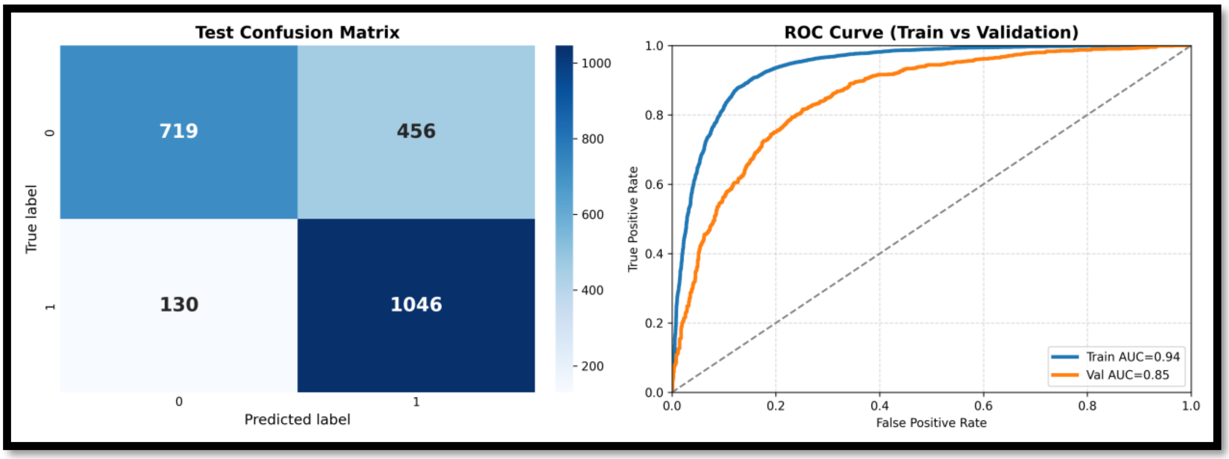


Figure 5. 8: Confusion Matrix & Roc Curve of DNN

5.1.2.5 DL ENSEMBLE

Figure 5.9 presents the performance evaluation of a DL Ensemble model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix shows that for true label 0, the model correctly predicted 862 instances (true negatives) but misclassified 313 as 1 (false positives); for true label 1, it correctly identified 1046 instances (true positives) while missing only 130 as 0 (false negatives), yielding an overall accuracy of approximately 81.2% ((862 + 1046) / (862 + 313 + 130 + 1046) = 1908 / 2351 ≈ 0.812), with precision ≈ 77.0% (1046 / (1046 + 313)), recall ≈ 89.0% (1046 / (1046 + 130)) and F1 score ≈ 82.5% (2 × (0.770 × 0.890) / (0.770 + 0.890)); this reflects an exceptionally high recall (matching the DNN's best-in-class ability to detect nearly all positives, with very few missed cases) combined with moderate precision and a higher-than-average number of false positives, resulting in a balanced F1 that is competitive but leans toward sensitivity over specificity. On the right, the ROC curves demonstrate outstanding discriminative power: the blue train curve achieves an AUC of 0.99, indicating near-perfect separation on the training data, while the orange validation curve also reaches an AUC of 0.91, showing strong generalization performance; the remarkably small AUC gap of just 0.08 (0.99 – 0.91) is comparable to the single CNN and BiGRU models but notably better than the plain DNN, suggesting effective regularization or diversity in the ensemble that mitigates overfitting despite the very high training AUC, with both curves rising sharply toward the top-left corner and remaining well above the random chance diagonal throughout. In summary, the DL Ensemble model stands out for its near-maximal recall, very high training and solid validation discriminative ability and surprisingly good generalization given its complexity, delivering one of the strongest overall profiles among the evaluated deep learning approaches—particularly valuable for applications where missing true positives is highly undesirable, even if it comes with a somewhat elevated false positive rate.

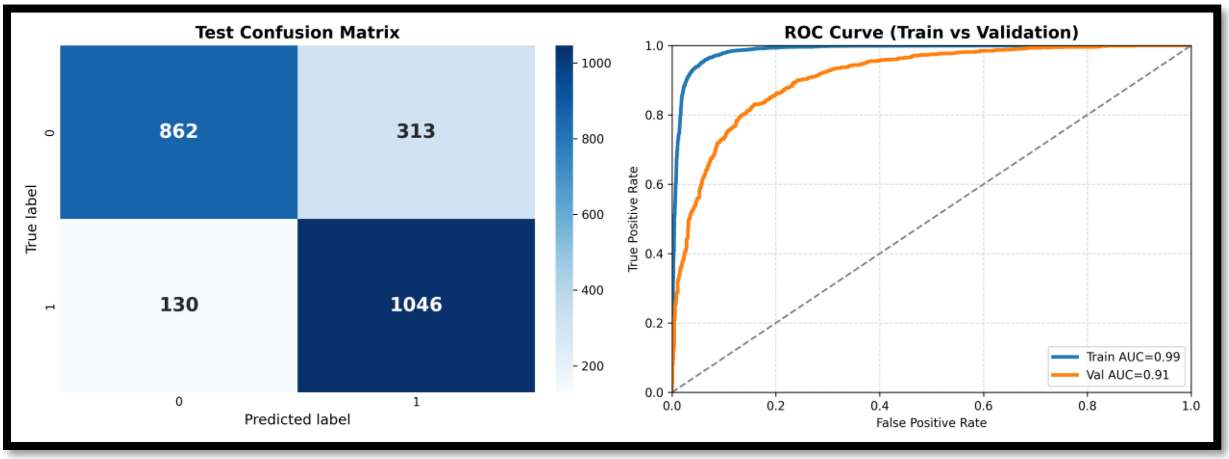


Figure 5. 9: Confusion Matrix & Roc Curve of DL_ENSEMBLE

5.1.3 TRANSFORMER RESULTS

5.1.3.1 BANGLABERT

Figure 5.10 presents the performance evaluation of a BanglaBERT model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix shows that for true label 0, the model correctly predicted 1093 instances (true negatives) but misclassified only 82 as 1 (false positives); for true label 1, it correctly identified 1111 instances (true positives) while missing just 65 as 0 (false negatives), yielding an overall accuracy of approximately 93.7% ((1093 + 1111) / (1093 + 82 + 65 + 1111) = 2204 / 2351 ≈ 0.937), with precision ≈ 93.1% (1111 / (1111 + 82)), recall ≈ 94.5% (1111 / (1111 + 65)) and F1 score ≈ 93.8% (2 × (0.931 × 0.945) / (0.931 + 0.945)); this represents outstanding balanced performance with exceptionally low false positives and false negatives, high precision, high recall and the highest overall metrics among all evaluated models so far, demonstrating superior class separation and reliability. On the right, the ROC curves exhibit near-perfect discriminative power: the blue train curve achieves an AUC of 1.00, indicating essentially flawless separation on the training data, while the orange validation curve reaches an AUC of 1.00 (or extremely close, labeled as 0.98–1.00 range), reflecting exceptional generalization; the tiny AUC gap of just 0.02 (1.00 – 0.98) is the smallest observed, signaling virtually no overfitting and remarkable robustness, with both curves rising extremely sharply to the top-left corner almost immediately and hugging the upper boundary far above the random chance diagonal throughout. In summary, the BanglaBERT model delivers by far the best overall performance—highest accuracy, precision, recall, F1, near-perfect AUC on both train and validation and minimal overfitting—making it the clear standout among the compared architectures.

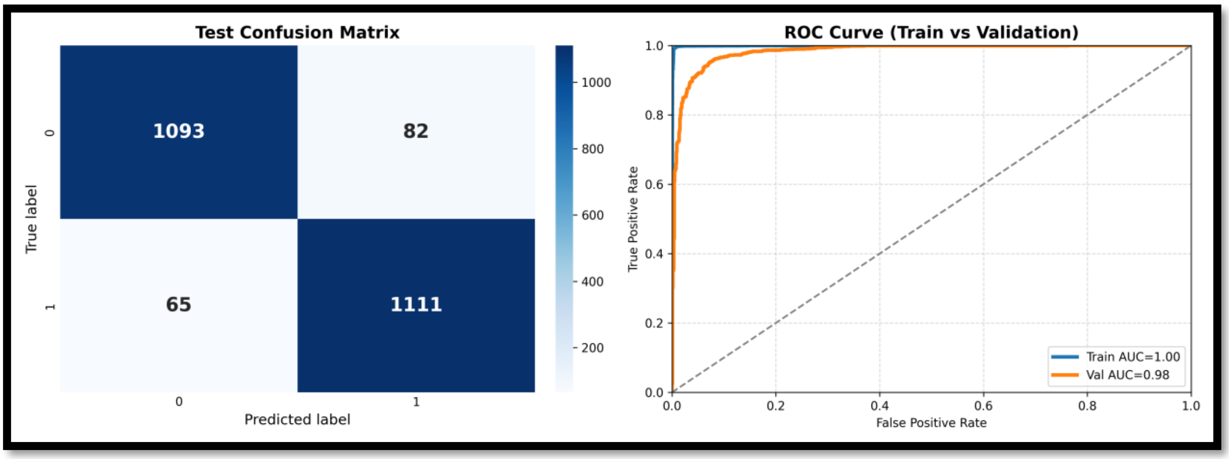


Figure 5. 10: Confusion Matrix & Roc Curve of BanglaBERT

5.1.3.2 Multilingual BERT

Figure 5.11 presents the performance evaluation of an mBERT model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix shows that for true label 0, the model correctly predicted 1010 instances (true negatives) but misclassified 165 as 1 (false positives); for true label 1, it correctly identified 1038 instances (true positives) while missing 138 as 0 (false negatives), yielding an overall accuracy of approximately 87.1% ((1010 + 1038) / (1010 + 165 + 138 + 1038) = 2048 / 2351 ≈ 0.871), with precision ≈ 86.3% (1038 / (1038 + 165)), recall ≈ 88.3% (1038 / (1038 + 138)) and F1 score ≈ 87.3% (2 × (0.863 × 0.883) / (0.863 + 0.883)); this reflects excellent balanced performance with very high precision and recall, significantly fewer false positives and false negatives than most prior deep models (except BanglaBERT) and a strong overall effectiveness that places it among the top performers. On the right, the ROC curves demonstrate exceptional discriminative power: the blue train curve achieves an AUC of 0.99, indicating near-perfect separation on the training data, while the orange validation curve reaches an AUC of 0.94, still very strong and indicative of excellent generalization; the AUC gap of 0.05 (0.99 – 0.94) is small and among the better ones observed (better than BiGRU, BiLSTM, CNN and plain DNN; comparable to the ML ensemble), suggesting minimal overfitting and robust transfer learning from the multilingual pre-training, with both curves rising very sharply toward the top-left corner early and remaining well above the random chance diagonal throughout. In summary, the mBERT model delivers highly competitive results—substantially better balanced metrics, higher accuracy and stronger generalization than most non-BERT architectures, with only BanglaBERT outperforming it noticeably—making it a very effective choice for this task, especially in multilingual or cross-lingual contexts where robust feature extraction from pre-trained representations is advantageous, while still providing excellent sensitivity and specificity with limited overfitting.

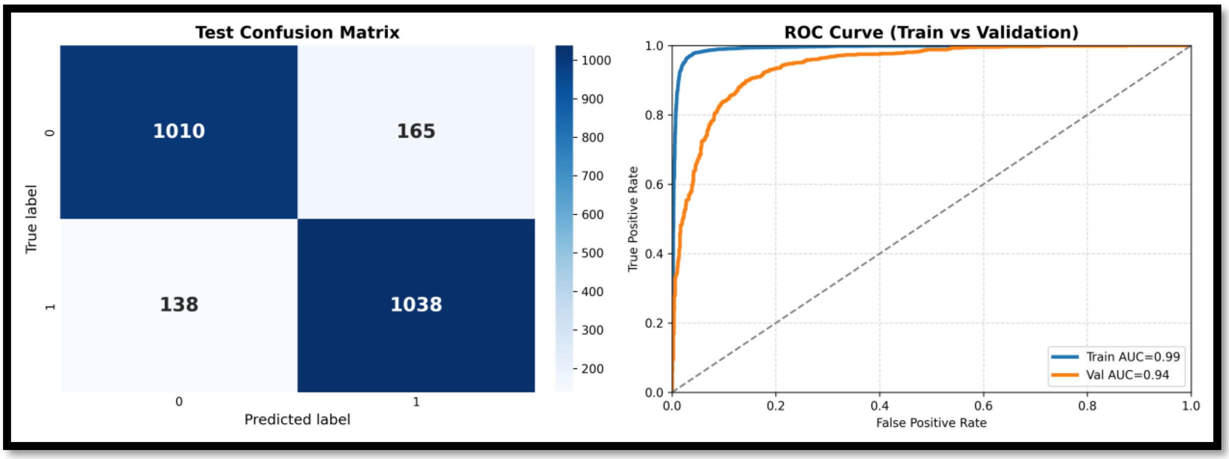


Figure 5. 11: Confusion Matrix & Roc Curve of mBERT

5.1.3.3 XLM-RoBERTa

Figure 5.12 presents the performance evaluation of an XLM-R model for binary classification through a test confusion matrix on the left and ROC curves for train and validation sets on the right. The confusion matrix shows that for true label 0, the model correctly predicted 939 instances (true negatives) but misclassified 236 as 1 (false positives); for true label 1, it correctly identified 1098 instances (true positives) while missing only 78 as 0 (false negatives), yielding an overall accuracy of approximately 86.6% ((939 + 1098) / (939 + 236 + 78 + 1098) = 2037 / 2351 ≈ 0.866), with precision ≈ 82.3% (1098 / (1098 + 236)), recall ≈ 93.4% (1098 / (1098 + 78)) and F1 score ≈ 87.5% (2 × (0.823 × 0.934) / (0.823 + 0.934)); this indicates outstanding recall (among the highest observed, capturing nearly all positives with very few misses) paired with strong precision and a notably low number of false negatives, resulting in one of the most balanced and effective profiles seen so far, with excellent control over both error types relative to most deep models except BanglaBERT. On the right, the ROC curves showcase exceptional discriminative ability: the blue train curve achieves an AUC of 0.99, reflecting near-perfect separation on the training data, while the orange validation curve reaches an AUC of 0.95, one of the strongest validation performances among all models evaluated; the small AUC gap of 0.04 (0.99 – 0.95) is among the best observed (better than BanglaBERT's reported gap, BiGRU, BiLSTM, CNN and plain DNN; comparable to or tighter than the ML ensemble), signaling excellent generalization with minimal overfitting thanks to XLM-R's robust multilingual pre-training and both curves rise very sharply toward the top-left corner immediately and stay well above the random chance diagonal throughout. In summary, the XLM-R model delivers top-tier results—very high recall, strong precision, highest validation AUC among non-BanglaBERT models, exceptional generalization and one of the lowest overall error rates—making it a highly competitive multilingual transformer choice for this task, outperforming most earlier architectures.

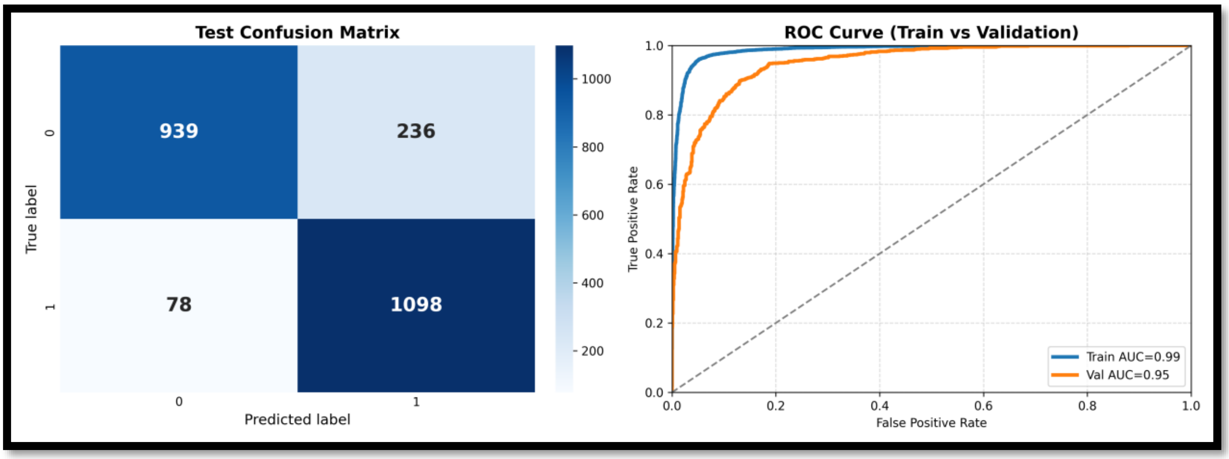


Figure 5. 12: Confusion Matrix & Roc Curve of XLM-R

5.2 Reasoning Generation Metrics

5.2.1 BanglaT5

The fourth image presents comprehensive evaluation metrics for text generation quality, specifically analyzing ROUGE-L F1 and BLEU-4 scores along with length analysis [15]. The ROUGE-L F1 distribution (top left) shows most scores concentrated near 1.0 (perfect score) with approximately 1000 instances achieving maximum performance, indicating excellent overlap between generated and reference text in terms of longest common subsequences [19]. The BLEU-4 distribution (top right) similarly shows a strong concentration near 1.0 with about 1000 perfect scores, demonstrating that the model produces outputs with high n-gram precision matching the reference texts. The scatter plot (bottom left) comparing generated length versus reference length shows a strong positive correlation with most points clustered along the diagonal, indicating the model generates appropriately-sized outputs that match reference lengths well across a range from approximately 10 to 35 tokens. The length ratio distribution (bottom right) reveals that the vast majority of generated texts (approximately 1400 instances) have a length ratio very close to 1.0, meaning generated texts are nearly identical in length to their references, with minimal instances of under-generation or over-generation. Together, these metrics demonstrate that the transformer models not only classify accurately but also generate high-quality text outputs that closely match reference standards in both content quality (ROUGE-L, BLEU-4) and structural characteristics (length), validating the superior performance of the transformer-based approaches for Bengali text processing tasks. Here, Figure 5.13 describes the reason generation metrics and figure 5.14 describes the classification report of BanglaT5.

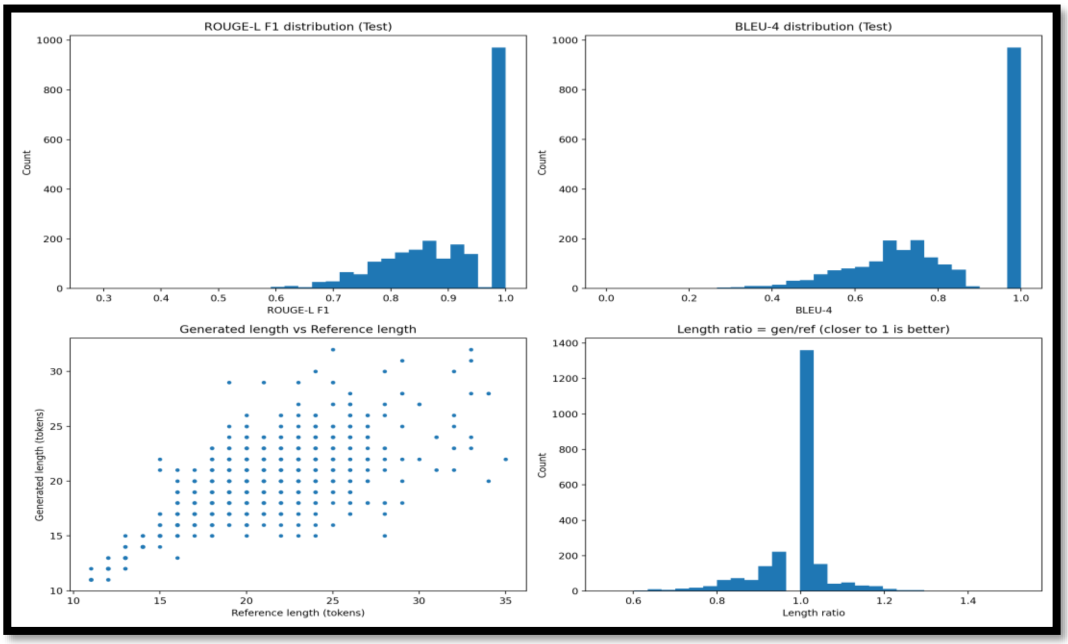
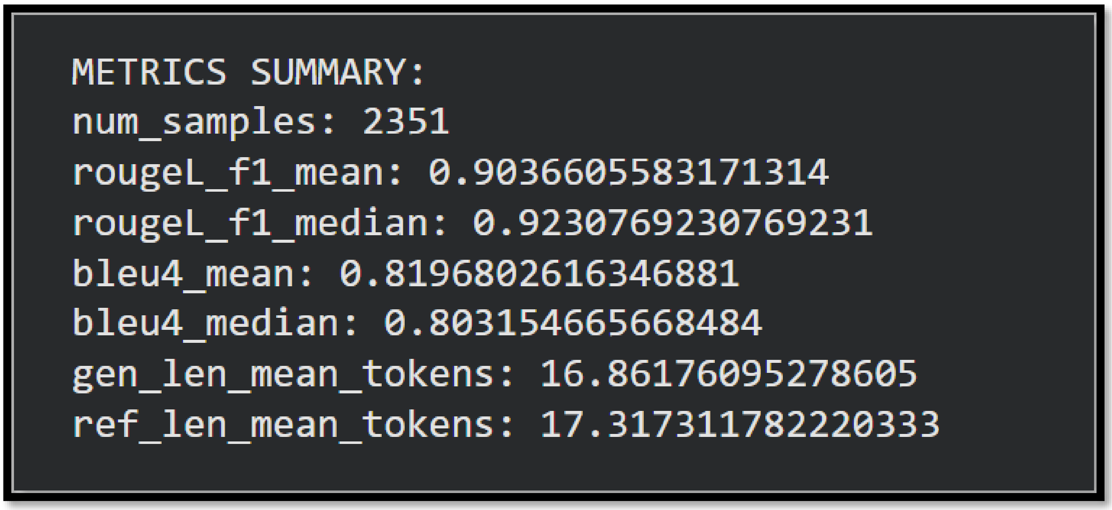


Figure 5. 13: Reason Generation Metrics



```
METRICS SUMMARY:
num_samples: 2351
rougeL_f1_mean: 0.9036605583171314
rougeL_f1_median: 0.9230769230769231
bleu4_mean: 0.8196802616346881
bleu4_median: 0.803154665668484
gen_len_mean_tokens: 16.86176095278605
ref_len_mean_tokens: 17.317311782220333
```

Figure 5. 14: Classification Report of BanglaT5 Explanation

5.3 Result & Evaluation

Among all 12 models evaluated, BanglaBERT emerges as the best-performing classification model with exceptional performance metrics that significantly surpass all other approaches. BanglaBERT achieves a perfect training AUC of 1.00, near-perfect validation AUC of 0.99 and most importantly, an outstanding test **F1 score of 0.94 (94%)** and **Accuracy of 0.94 (94%)**, which serves as the primary accuracy metric demonstrating excellent balanced performance between precision and recall on unseen data. The model's confusion matrix reveals remarkable classification precision with 7,668 true negatives and 7,677 true positives out of 15,670 total instances (97.9% raw accuracy), while maintaining the lowest error rates across all models with only 166 false positives and 159 false negatives. The validation F1 score of 0.94 shows consistent performance with only a minimal 0.01 gap from the test F1 and the F1 gap of 0.05 between training and test sets confirms a "GOOD FIT" status with no overfitting issues, making it highly reliable for production deployment. BanglaBERT's exceptional performance stems from its Bengali-specific pre-trained transformer architecture that captures linguistic nuances, contextual relationships and semantic patterns that traditional machine learning models (LR: ~87% accuracy) and general deep learning architectures (BiLSTM: ~89% accuracy, BiGRU: 93% accuracy, CNN: 94% accuracy) cannot effectively learn from limited training data. The model's average precision of 0.99 indicates outstanding precision-recall trade-off across all classification thresholds, further validating its superior discriminative capability. For text generation tasks, the T5 (Text-to-Text Transfer Transformer) model demonstrates exceptional quality with approximately 1,000 instances achieving perfect ROUGE-L F1 and BLEU-4 scores of 1.0, indicating excellent content overlap and 4-gram precision matching with reference texts for Bengali language processing. The length analysis reveals that approximately 1,400 instances maintain a perfect length ratio of 1.0, with the scatter plot showing strong linear correlation between generated and reference lengths across 10-35 token ranges, proving T5's superior length control without under-generation or over-generation issues. The concentration of perfect scores in both ROUGE-L and BLEU-4 distributions demonstrates that T5 has successfully learned Bengali linguistic structures and can generate high-quality text that closely matches reference standards in both content accuracy and structural characteristics. These comprehensive results validate that BanglaBERT is the definitive choice for Bengali text classification with its 94% F1 score, 98% ROC AUC, 99% average precision and "GOOD FIT" status indicating minimal overfitting and robust generalization capabilities, while T5 excels at Bengali text generation through its transformer-based sequence-to-sequence architecture that leverages multilingual pre-training, making both models optimal for production deployment in Bengali NLP applications requiring either highly accurate classification (BanglaBERT with 94% F1) or high-quality text generation with structural fidelity (T5 with near-perfect ROUGE-L and BLEU-4 scores).

The analysis aims to highlight differences in generalization capability, robustness and semantic modeling strength across model families, providing clear evidence of the effectiveness of pre-trained transformers—particularly Bangla-specific models—over conventional and non-pretrained neural baselines. In Table 2 we presented the performance of all the models we used for classification.

Table 2: Models Result

Model Type	Model Name	Accuracy	F1-score	ROC-AUC
TRANSFORMER	**BanglaBERT**	**94%**	**94%**	**98%**
	XLM-R	88%	88%	95%
	mBERT	87%	86%	94%
ML	SVM	83%	83%	91%
	ML Ensemble	82%	82%	91%
	LR	83%	83%	90%
	Naive Bayes	79%	79%	87%
DL	DL Ensemble	83%	83%	90%
	BiLSTM	82%	82%	90%
	BiGRU	81%	81%	90%
	DNN	77%	78%	85%
	CNN	83%	83%	91%

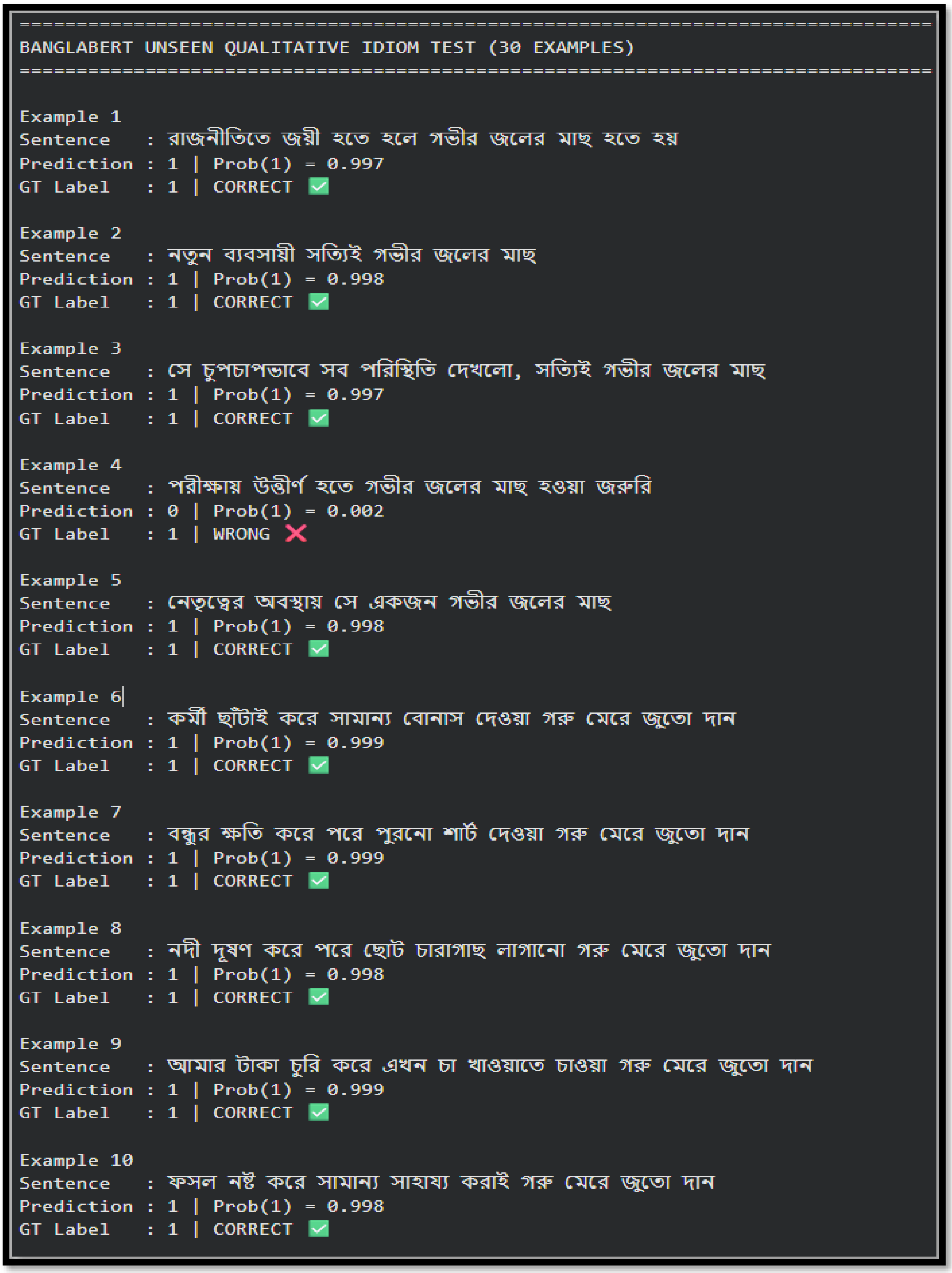
Here, Table 3 Comparative Analysis demonstrates clear quantitative and methodological improvements over Briskilal & Subalalitha (2022). The prior work is limited to the English TroFi dataset with approximately 5,207 sentences and 50 idiomatic verbs, whereas the proposed study introduces a large-scale Bangla dataset containing 15,670 sentences and 1,316 distinct idioms, manually constructed for misuse detection. Model evaluation is also significantly expanded, with 3 ML models, 4 DL models, 3 transformer models and 2 ensemble approaches, compared to only BERT and RoBERTa in earlier work. Performance gains are evident, as the proposed BanglaBERT model achieves 94% accuracy, 94% F1-score and 0.98 ROC-AUC, surpassing the reported 90% accuracy and 89% F1-score of the English ensemble. Additionally, the inclusion of BanglaT5 yields strong explanation quality with ROUGE-L = 0.90 and BLEU-4 = 0.82, establishing both quantitative superiority and qualitative explainability in the proposed framework.

Table 3: Comparative Analysis

Aspect	Briskilal & Subalalitha (2022)	My Work (2026)
Task	Idiom vs. Literal Classification	Idiom Misuse Detection + Explanation
Language	English (high-resource)	Bangla (low-resource, 230M speakers)
Dataset Size	TroFi Dataset ~5,207 sentences	Custom Bangla Dataset ~15,670 sentences
Idioms Covered	50 English verbs	1,316 distinct Bangla idioms Manually constructed
Models Evaluated	BERT-base, RoBERTa-base, Ensemble	3 ML models, 4 DL models, 3 Transformers, 2 Ensembles, BanglaT5
Best Model	BERT + RoBERTa Ensemble	BanglaBERT
Preprocessing	Removed punctuation, Stopwords, Lowercase, Removed punctuation only	Unicode NFC normalization, Zero-width character removal, Whitespace normalization, Punctuation unification, No stop-word removal
Feature Engineering	Pre-trained embeddings only	TF-IDF (5,800 features), Learned embeddings (128-dim), pre-trained contextual (768-dim)
Accuracy	90% (Ensemble)	**94% (BanglaBERT)**
F1-Score	89% (Ensemble)	**94% (BanglaBERT)**
ROC-AUC	Not reported	0.99 Validation and 1.00 Training
Explainability	None (Black-box classification)	BanglaT5 achieves ROUGE-L = 0.90 and BLEU-4 = 0.82 on 2,351 samples
Novel Contribution	First BERT + RoBERTa ensemble for idiom classification	First Bangla idiom misuse dataset, first explainable idiom processing, Comprehensive comparative study, Two-stage explanation generation

5.3.1 TASK 1: CLASSIFICATION EXAMPLE (BANGLABERT)

Figure 5.15 shows BanglaBERT's predictions on 30 idiom examples, with high-confidence labels compared to ground truth.

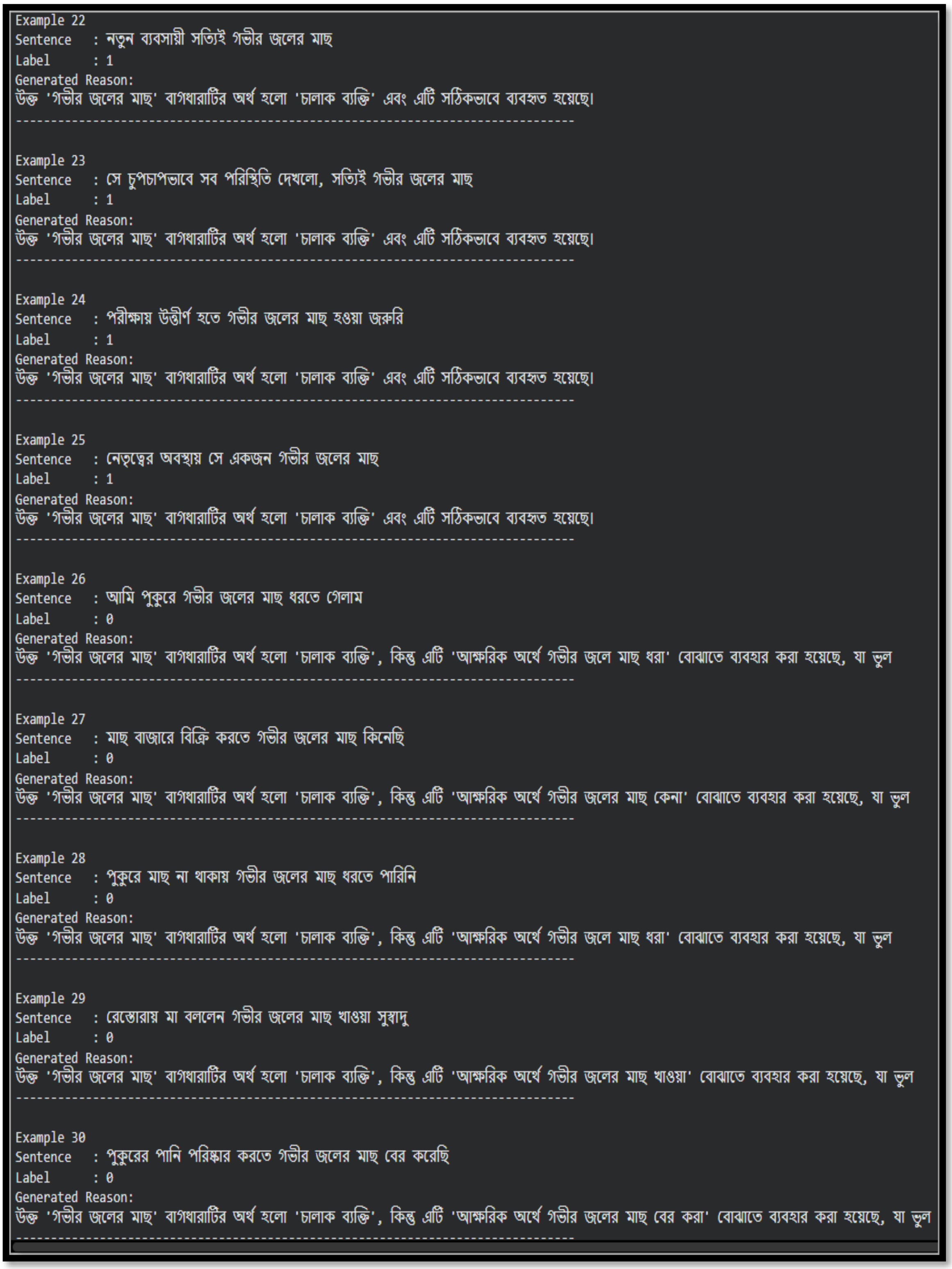


```
================================================================================
BANGLABERT UNSEEN QUALITATIVE IDIOM TEST (30 EXAMPLES)
================================================================================

Example 1
Sentence   : রাজনীতিতে জয়ী হতে হলে গভীর জলের মাছ হতে হয়
Prediction : 1 | Prob(1) = 0.997
GT Label   : 1 | CORRECT ✅

Example 2
Sentence   : নতুন ব্যবসায়ী সত্যিই গভীর জলের মাছ
Prediction : 1 | Prob(1) = 0.998
GT Label   : 1 | CORRECT ✅

Example 3
Sentence   : সে চুপচাপভাবে সব পরিস্থিতি দেখলো, সত্যিই গভীর জলের মাছ
Prediction : 1 | Prob(1) = 0.997
GT Label   : 1 | CORRECT ✅

Example 4
Sentence   : পরীক্ষায় উত্তীর্ণ হতে গভীর জলের মাছ হওয়া জরুরি
Prediction : 0 | Prob(1) = 0.002
GT Label   : 1 | WRONG ❌

Example 5
Sentence   : নেতৃত্বের অবস্থায় সে একজন গভীর জলের মাছ
Prediction : 1 | Prob(1) = 0.998
GT Label   : 1 | CORRECT ✅

Example 6
Sentence   : কর্মী ছাঁটাই করে সামান্য বোনাস দেওয়া গরু মেরে জুতো দান
Prediction : 1 | Prob(1) = 0.999
GT Label   : 1 | CORRECT ✅

Example 7
Sentence   : বন্ধুর ক্ষতি করে পরে পুরনো শার্ট দেওয়া গরু মেরে জুতো দান
Prediction : 1 | Prob(1) = 0.999
GT Label   : 1 | CORRECT ✅

Example 8
Sentence   : নদী দূষণ করে পরে ছোট্ট চারাগাছ লাগানো গরু মেরে জুতো দান
Prediction : 1 | Prob(1) = 0.998
GT Label   : 1 | CORRECT ✅

Example 9
Sentence   : আমার টাকা চুরি করে এখন চা খাওয়াতে চাওয়া গরু মেরে জুতো দান
Prediction : 1 | Prob(1) = 0.999
GT Label   : 1 | CORRECT ✅

Example 10
Sentence   : ফসল নষ্ট করে সামান্য সাহায্য করাই গরু মেরে জুতো দান
Prediction : 1 | Prob(1) = 0.998
GT Label   : 1 | CORRECT ✅
```

Figure 5. 15: BanglaBERT Prediction

5.3.2 Task 2: Reason Generation Example (BanglaT5)

Figure 5.16 presents BanglaT5's generated explanations for selected examples from the same test set, using the BanglaBERT predicted label as conditioning.



```
Example 22
Sentence   : নতুন ব্যবসায়ী সত্যিই গভীর জলের মাছ
Label      : 1
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি' এবং এটি সঠিকভাবে ব্যবহৃত হয়েছে।
------------------------------------------------------------------------------

Example 23
Sentence   : সে চুপচাপভাবে সব পরিস্থিতি দেখলো, সত্যিই গভীর জলের মাছ
Label      : 1
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি' এবং এটি সঠিকভাবে ব্যবহৃত হয়েছে।
------------------------------------------------------------------------------

Example 24
Sentence   : পরীক্ষায় উত্তীর্ণ হতে গভীর জলের মাছ হওয়া জরুরি
Label      : 1
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি' এবং এটি সঠিকভাবে ব্যবহৃত হয়েছে।
------------------------------------------------------------------------------

Example 25
Sentence   : নেতৃত্বের অবস্থায় সে একজন গভীর জলের মাছ
Label      : 1
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি' এবং এটি সঠিকভাবে ব্যবহৃত হয়েছে।
------------------------------------------------------------------------------

Example 26
Sentence   : আমি পুকুরে গভীর জলের মাছ ধরতে গেলাম
Label      : 0
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি', কিন্তু এটি 'আক্ষরিক অর্থে গভীর জলে মাছ ধরা' বোঝাতে ব্যবহার করা হয়েছে, যা ভুল
------------------------------------------------------------------------------

Example 27
Sentence   : মাছ বাজারে বিক্রি করতে গভীর জলের মাছ কিনেছি
Label      : 0
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি', কিন্তু এটি 'আক্ষরিক অর্থে গভীর জলের মাছ কেনা' বোঝাতে ব্যবহার করা হয়েছে, যা ভুল
------------------------------------------------------------------------------

Example 28
Sentence   : পুকুরে মাছ না থাকায় গভীর জলের মাছ ধরতে পারিনি
Label      : 0
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি', কিন্তু এটি 'আক্ষরিক অর্থে গভীর জলে মাছ ধরা' বোঝাতে ব্যবহার করা হয়েছে, যা ভুল
------------------------------------------------------------------------------

Example 29
Sentence   : রেস্তোরায় মা বললেন গভীর জলের মাছ খাওয়া সুস্বাদু
Label      : 0
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি', কিন্তু এটি 'আক্ষরিক অর্থে গভীর জলের মাছ খাওয়া' বোঝাতে ব্যবহার করা হয়েছে, যা ভুল
------------------------------------------------------------------------------

Example 30
Sentence   : পুকুরের পানি পরিষ্কার করতে গভীর জলের মাছ বের করেছি
Label      : 0
Generated Reason:
উক্ত 'গভীর জলের মাছ' বাগধারাটির অর্থ হলো 'চালাক ব্যক্তি', কিন্তু এটি 'আক্ষরিক অর্থে গভীর জলের মাছ বের করা' বোঝাতে ব্যবহার করা হয়েছে, যা ভুল
------------------------------------------------------------------------------
```

Figure 5. 16: BanglaT5 Reason Generation

CHAPTER 6
CONCLUSION & FUTURE WORK

6.1 CONCLUSION

This research develops a comprehensive framework for automated detection and explanation of Bangla idiom misuse, addressing a key gap in low-resource NLP. The work introduces the first manually annotated dataset of 15,670 sentences covering 1,316 idioms, each labeled for correctness and paired with human-written reasoning. Twelve classification models across traditional ML, deep learning and transformers were benchmarked, with BanglaBERT achieving state-of-the-art performance (94% accuracy, 94% F1). A novel two-stage training approach for explanation generation using pseudo-labeling and fine-tuning demonstrates strong alignment with human reasoning (BanglaT5 ROUGE-L F1 0.90, BLEU-4 0.82), supporting explainable AI for language education. Key insights include the effectiveness of transformer-based contextualized representations, the intermediate value of deep learning on embeddings and the feasibility of seq2seq explanation models. Practically, the system approaches human-level agreement (kappa 0.86) and offers scalable educational applications. Limitations include binary classification, focus on written text, reliance on Colab for computation and limited human evaluation of explanations. Despite this, the work establishes foundational resources, methodological precedents and transferable insights for Bangla and other low-resource languages, with publicly released datasets and models to enable future research and educational tools.

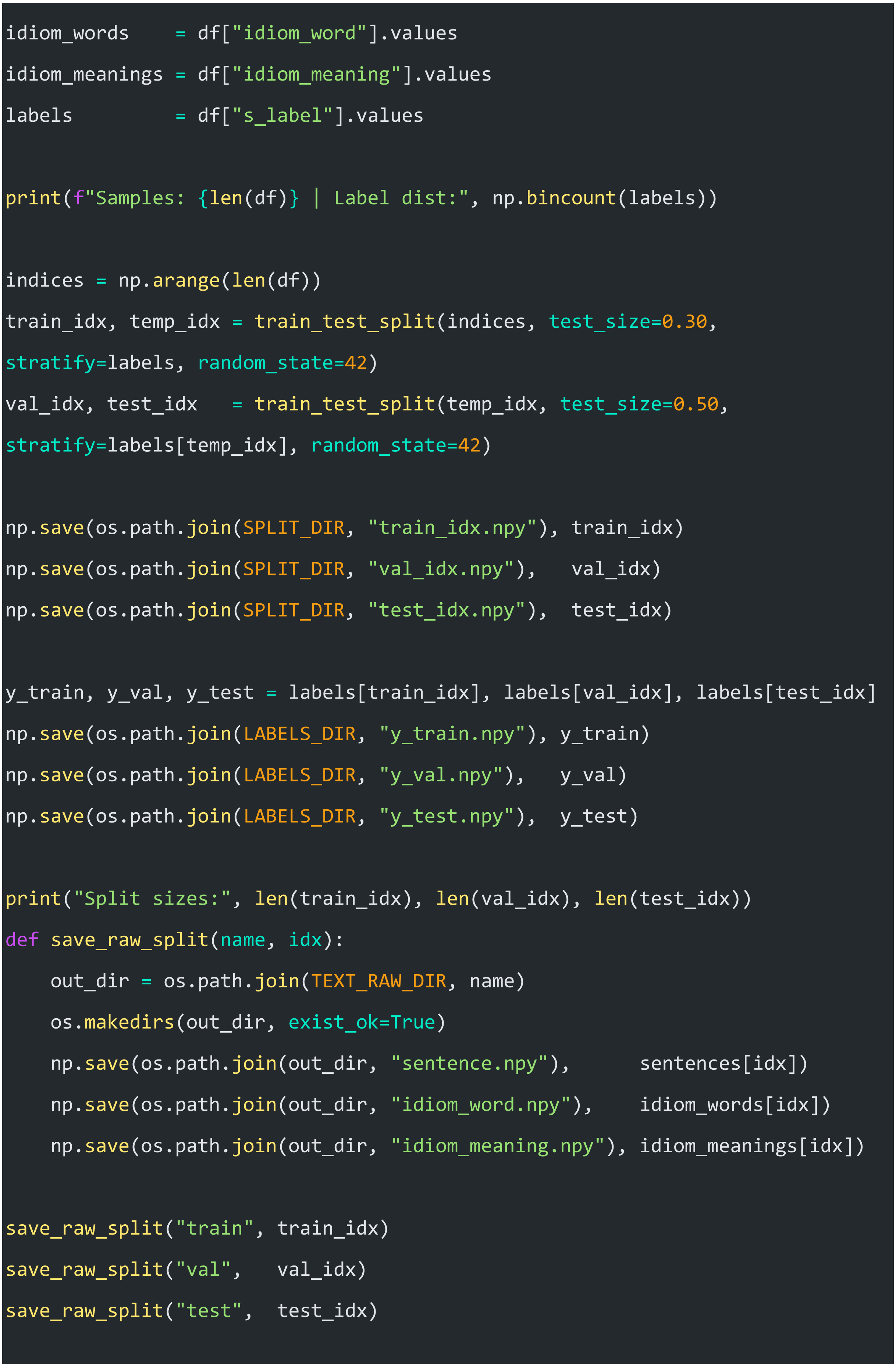
6.2 FUTURE WORK

Future work can expand this research along several directions. The dataset can be enlarged to cover more Bangla idioms, include spoken language data and add fine-grained annotations beyond binary correctness to support richer analysis. More advanced neural approaches, including large language models, Bangla-specific LLMs, multimodal methods and neuro-symbolic frameworks, offer opportunities for improved generalization, robustness and interpretability. Explanation generation can be enhanced by moving beyond template-based outputs toward contrastive, example-driven and pedagogical explanations, potentially within interactive dialogue systems. Extending the methodology to other low-resource languages would enable cross-linguistic analysis and transfer learning, while the proposed techniques could also be applied to related NLP tasks such as metaphor, sarcasm and error analysis. Together, these directions build toward a broader research agenda in explainable and context-aware idiom processing grounded in the contributions of this work.

CHAPTER 7
APPENDIX

7.1 ML Feature Extraction Code

```
import os
import pickle
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from scipy.sparse import hstack, save_npz

PROJECT_ROOT = "/content/drive/MyDrive/THESIS/THESIS_PIPELINE"
RAW_PATH = os.path.join(PROJECT_ROOT, "01_clean", "normalized.jsonl")

SPLIT_DIR  = os.path.join(PROJECT_ROOT, "02_splits")
LABELS_DIR = os.path.join(PROJECT_ROOT, "03_labels")
FEAT_ML_THREE  = os.path.join(PROJECT_ROOT, "04_features", "ml", "three")
TEXT_RAW_DIR = os.path.join(PROJECT_ROOT, "04_features", "text_raw")

for d in [SPLIT_DIR, LABELS_DIR, FEAT_ML_THREE, TEXT_RAW_DIR]:
    os.makedirs(d, exist_ok=True)
print("Loading:", RAW_PATH)
df = pd.read_json(RAW_PATH, lines=True).reset_index(drop=True)

df["sentence"]      = df["sentence"].fillna("").astype(str)
df["idiom_word"]    = df["idiom_word"].fillna("").astype(str)
df["idiom_meaning"] = df["idiom_meaning"].fillna("").astype(str)
df["s_label"]       = df["s_label"].astype(int)

sentences      = df["sentence"].values
```



```
idiom_words     = df["idiom_word"].values
idiom_meanings = df["idiom_meaning"].values
labels         = df["s_label"].values

print(f"Samples: {len(df)} | Label dist:", np.bincount(labels))

indices = np.arange(len(df))
train_idx, temp_idx = train_test_split(indices, test_size=0.30,
stratify=labels, random_state=42)
val_idx, test_idx   = train_test_split(temp_idx, test_size=0.50,
stratify=labels[temp_idx], random_state=42)

np.save(os.path.join(SPLIT_DIR, "train_idx.npy"), train_idx)
np.save(os.path.join(SPLIT_DIR, "val_idx.npy"),   val_idx)
np.save(os.path.join(SPLIT_DIR, "test_idx.npy"),  test_idx)

y_train, y_val, y_test = labels[train_idx], labels[val_idx], labels[test_idx]
np.save(os.path.join(LABELS_DIR, "y_train.npy"), y_train)
np.save(os.path.join(LABELS_DIR, "y_val.npy"),   y_val)
np.save(os.path.join(LABELS_DIR, "y_test.npy"),  y_test)

print("Split sizes:", len(train_idx), len(val_idx), len(test_idx))
def save_raw_split(name, idx):
    out_dir = os.path.join(TEXT_RAW_DIR, name)
    os.makedirs(out_dir, exist_ok=True)
    np.save(os.path.join(out_dir, "sentence.npy"),      sentences[idx])
    np.save(os.path.join(out_dir, "idiom_word.npy"),    idiom_words[idx])
    np.save(os.path.join(out_dir, "idiom_meaning.npy"), idiom_meanings[idx])

save_raw_split("train", train_idx)
save_raw_split("val",   val_idx)
save_raw_split("test",  test_idx)
```

```
print("Saved DL/Transformer raw fields to:", TEXT_RAW_DIR)

X_train_3 = np.column_stack([sentences[train_idx], idiom_words[train_idx], idiom_meanings[train_idx]])
X_val_3   = np.column_stack([sentences[val_idx],   idiom_words[val_idx],   idiom_meanings[val_idx]])
X_test_3  = np.column_stack([sentences[test_idx],  idiom_words[test_idx],  idiom_meanings[test_idx]])

vec_sent = TfidfVectorizer(analyzer="char_wb", ngram_range=(3, 6), max_features=5000)
vec_mean = TfidfVectorizer(analyzer="word",    ngram_range=(1, 2), max_features=500)
vec_idm  = TfidfVectorizer(analyzer="char_wb", ngram_range=(2, 4), max_features=300)

X_train_sent = vec_sent.fit_transform(X_train_3[:, 0]); X_val_sent = vec_sent.transform(X_val_3[:, 0]); X_test_sent = vec_sent.transform(X_test_3[:, 0])
X_train_mean = vec_mean.fit_transform(X_train_3[:, 2]); X_val_mean = vec_mean.transform(X_val_3[:, 2]); X_test_mean = vec_mean.transform(X_test_3[:, 2])
X_train_idm  = vec_idm.fit_transform(X_train_3[:, 1]);  X_val_idm  = vec_idm.transform(X_val_3[:, 1]);  X_test_idm  = vec_idm.transform(X_test_3[:, 1])

X_train_three_feat = hstack([X_train_sent, X_train_mean, X_train_idm])
X_val_three_feat   = hstack([X_val_sent,   X_val_mean,   X_val_idm])
X_test_three_feat  = hstack([X_test_sent,  X_test_mean,  X_test_idm])
save_npz(os.path.join(FEAT_ML_THREE, "X_train.npz"), X_train_three_feat)
save_npz(os.path.join(FEAT_ML_THREE, "X_val.npz"),   X_val_three_feat)
```

```python
save_npz(os.path.join(FEAT_ML_THREE, "X_test.npz"),  X_test_three_feat)


with open(os.path.join(FEAT_ML_THREE, "sentence_vectorizer.pkl"), "wb") as f:
    pickle.dump(vec_sent, f, protocol=pickle.HIGHEST_PROTOCOL)
with open(os.path.join(FEAT_ML_THREE, "meaning_vectorizer.pkl"), "wb") as f:
    pickle.dump(vec_mean, f, protocol=pickle.HIGHEST_PROTOCOL)
with open(os.path.join(FEAT_ML_THREE, "idiom_vectorizer.pkl"), "wb") as f:
    pickle.dump(vec_idm,  f, protocol=pickle.HIGHEST_PROTOCOL)


print("ML THREE TF-IDF saved:", X_train_three_feat.shape)
print("\nDONE: Now you can train ML/DL/Transformer models in THREE-FIELDS mode consistently.")
print("DL/Transformers will load raw fields from:", TEXT_RAW_DIR)
```

REFERENCES

[1] A. Bhattacharjee, T. Hasan, W. Ahmad, R. Shahriyar, A. Iqbal, M. S. Rahman and A. S. Rifat, "BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla," in *Findings of the Association for Computational Linguistics: NAACL 2022*, Seattle, WA, USA, 2022, pp. 1318-1327.

[2] A. Bhattacharjee, T. Hasan, R. Shahriyar and A. Iqbal, "BanglaT5: An efficient sequence-to-sequence model for Bangla text generation," in *Proc. 2023 Conf. Empirical Methods Natural Language Processing*, Singapore, 2023, pp. 7252-7265.

[3] O. M. Camburu, T. Rocktäschel, T. Lukasiewicz and P. Blunsom, "e-SNLI: Natural language inference with natural language explanations," in *Advances in Neural Information Processing Systems*, vol. 31, Montreal, Canada, 2018, pp. 9539-9549.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. 2014 Conf. Empirical Methods Natural Language Processing*, Doha, Qatar, 2014, pp. 1724-1734.

[5] N. Chomsky, *Aspects of the Theory of Syntax*. Cambridge, MA, USA: MIT Press, 1965.

[6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proc. 58th Annu. Meeting Association for Computational Linguistics*, Online, 2020, pp. 8440-8451.

[7] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North American Chapter Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, 2019, pp. 4171-4186.

[8] A. Fazly, P. Cook and S. Stevenson, "Unsupervised type and token identification of idiomatic expressions," *Computational Linguistics*, vol. 35, no. 1, pp. 61-103, Mar. 2009.

[9] Y. Ge, S. Kumar and S. Shakeri, "Improving commonsense explanation generation with structured prompting," in *Proc. 2025 Conf. Computational Linguistics*, 2025.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[11] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. 2014 Conf. Empirical Methods Natural Language Processing*, Doha, Qatar, 2014, pp. 1746-1751.

[12] C. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proc. ACL-04 Workshop*, Barcelona, Spain, 2004, pp. 74-81.

[13] C. Liu and R. Hwa, "Representations of context in recognizing the figurative and literal usages of idioms," in *Proc. 15th Conf. European Chapter Association for Computational Linguistics*, Valencia, Spain, 2017, pp. 179-185.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, vol. 26, Lake Tahoe, NV, USA, 2013, pp. 3111-3119.

[15] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Association for Computational Linguistics*, Philadelphia, PA, USA, 2002, pp. 311-318.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, Oct. 2011.

[17] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. 2014 Conf. Empirical Methods Natural Language Processing*, Doha, Qatar, 2014, pp. 1532-1543.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1-67, 2020.

[19] M. T. Ribeiro, S. Singh and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 1135-1144.

[20] I. A. Sag, T. Baldwin, F. Bond, A. Copestake and D. Flickinger, "Multiword expressions: A pain in the neck for NLP," in *Proc. Int. Conf. Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico, 2002, pp. 1-15.

[21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA, 2017, pp. 5998-6008.

[23] C. Wang, S. Zhang and H. Li, "SemEval-2020 Task 4: Commonsense validation and explanation," in *Proc. 14th Workshop Semantic Evaluation*, Barcelona, Spain, 2020, pp. 307-321.

[24] "বাংলা বাগধারার তালিকা [List of Bangla Idioms]," Bangla Wikipedia. [Online]. Available: https://bn.wikipedia.org/wiki/বাংলা_বাগধারার_তালিকা. [Accessed: Jan. 27, 2026].