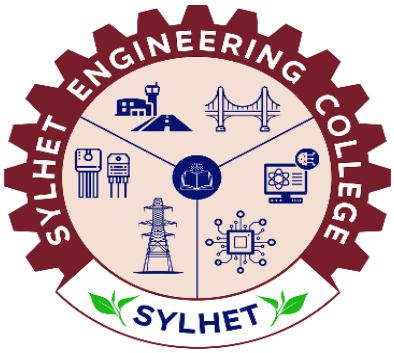
SYLHET ENGINEERING COLLEGE

(Shahjalal University of Science and Technology)

Department of Computer Science & Engineering



A Hybrid Deep Learning and Stylometric Feature-Based Framework for Authorship Attribution in Bangla Literature

Submitted by

Abdullah Md. Omar
Reg. No.: 2019331535
Session: 2019-20

Maliha Kaisar
Reg. No.: 2019331555
Session: 2019-20

Department of Computer Science & Engineering

Supervisor

Md. Abu Naser Mojumder
Associate Professor & Head
Department of Computer Science & Engineering
Sylhet Engineering College

22nd July 2025

Recommendation Letter from Thesis Supervisor

The thesis entitled "**A Hybrid Deep Learning and Stylometric Feature-Based Framework for Authorship Attribution in Bangla Literature**", which has been submitted by the students:

1. **Abdullah Md. Omar**
2. **Maliha Kaisar**

is a record of original research work that was carried out under my supervision, and I hereby approve that this report is being submitted in partial fulfillment of the requirements for the award of their Bachelor's Degree.

Signature of the Supervisor
Md. Abu Naser Mojumder
Associate Professor & Head
Department of Computer Science & Engineering
Sylhet Engineering College
22nd July, 2025

Certificate of Acceptance

The thesis is titled **"A Hybrid Deep Learning and Stylometric Feature-Based Framework for Authorship Attribution in Bangla Literature"** submitted by **Abdullah Md. Omar** and **Maliha Kaisar**; Student ID. **2019331535** and **2019331555**; Session **2019-20**, to the Department of Computer Science and Engineering , Sylhet Engineering College, has been accepted as satisfactory in partial fulfilment of the requirement for the Degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Internal

Nayan Kumar Nath
Lecturer
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet

Internal

Md Lysuzzaman
Lecturer
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet

Internal

Md. Rasel Ahmed
Assistant Professor
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet

Internal

Md. Nagrul Islam
Assistant Professor
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet

Chairman

Md. Abu Naser Mojumder
Head
Department of Computer Science and Engineering
Sylhet Engineering College, Sylhet

Member (External)

Dr. Mohammad Shahidur Rahman
Professor
Department of Computer Science and Engineering
Shahjalal University of Science and Technology

Acknowledgement

We sincerely express our gratitude to the Department of Computer Science and Engineering at Sylhet Engineering College for providing the necessary support and resources to initiate and advance our research.

We are especially thankful to our supervisor, Md. Abu Naser Mojumder, Associate Professor, Head of the Department of CSE, Sylhet Engineering College, for his invaluable guidance, continuous encouragement, and dedicated supervision throughout our research journey. His insightful feedback and expertise have been crucial in shaping the progress of our work thus far.

Additionally, we acknowledge the contributions of previous researchers and scholars, whose work has provided a strong foundation and valuable insights for our study. Their research has been an essential source of inspiration and reference.

As we continue refining our work through further data processing, model training, and analysis, we remain grateful for the support and encouragement from our families, friends, and peers, who have been with us throughout this journey.

Abstract

This thesis formulates a hybrid scheme that combines deep learning and stylometric features for authorship attribution of Bangla compositions. Authorship attribution, a key issue of Natural Language Processing (NLP), identifies authors of anonymous texts based on their unique linguistic and stylistic features. Though much improvement has been achieved in resource-abundant languages such as English, Bangla is yet to be explored thoroughly despite being the 7th most widely spoken language in the world, mainly because of a lack of digital linguistic resources and annotated corpora, coupled with its high inflectional complexity and diglossic nature (Shadhu Bhasha and Cholito Bhasha).To overcome these issues, we introduce the Authorship Attribution Bengali Dataset 20 (AABD20), a new, handpicked corpus of 20 influential Bangla writers covering classical and contemporary periods, aimed at facilitating strong computational research in Bangla authorship attribution. The dataset was developed by a two-stage approach, including manual extraction of data from PDF documents and automatic data extraction using an in-house web crawler, followed by text segmentation into fixed-size samples of around 750 words. Our suggested methodology combines both traditional machine learning and state-of-the-art deep learning approaches with great care to catch fine stylistic details. The approach includes multi-dimensional feature extraction, various word embedding schemes, various model training schemes, and a rigorous evaluation process. We experimented with various word embedding schemes,tri-dimensional feature extraction, FastText with Hierarchical Softmax, and a 6-layer Bi-LSTM model. The BanglaBERT model was fine-tuned on the AABD20 corpus to leverage its deep contextual understanding of Bengali for authorship attribution. Experimental results indicate that our system employing BanglaBERT fine-tuning on the AABD20 dataset yields a 99.00% best accuracy rate. This significantly outperforms previously reported methods and highlights the superior ability of transformer-based models to capture the rich contextual and syntactic nuances inherent in the Bangla language. Among the static embedding-based models, the Word2Vec Skip-Gram CNN achieved an impressive accuracy of 98.54%. This research is a rich addition to Bangla NLP, providing a basis for future innovations in low-resource language processing.

Keywords: Natural Language Processing (NLP), Authorship Attribution (AA), Stylometric analysis, Deep learning, Machine learning, Word embeddings, BanglaBERT, Text classification.

Contents

Table of Contents	Page
Title Page	1
Recommendation Letter from Thesis Supervisor	2
Certificate of Acceptance of the Thesis	3
Acknowledgement	4
Abstract	5
Contents	6
Chapter 1: Introduction	
1.1 Background	8
1.2 Authorship Attribution in Contemporary Digital Society	9
1.3 Evolution of Author Attribution Techniques	10
1.4 Problem Statement	10
1.5 Existing Research Gaps	11
1.6 Motivation for Research	12
1.7 Research Objectives	13
Chapter 2: Literature Review	
2.1 On Authorship Attribution	15
2.2 On Bangla	16
Chapter 3: Methodology	
3.1 Overview	19
3.2 Data Collection and Corpus Creation	19
3.2.1 Corpus Formation: AABD20	21
3.3 Proposed Methodology	25
3.3.1 Word Embeddings Approach with NN	27
3.3.2 Stylometric Analysis (Tri-Dimensional Features)	36
3.3.3 FastText with Hierarchical Softmax	43
3.3.4 Bi-LSTM with 6 Layers	45
3.3.5 Fine-Tuning BanglaBERT on AABD20	46
Chapter 4: Experimental Setup	
4.1 Hardware and Software Environment	50
4.2 Dataset Partitioning	50
4.3 Preprocessing and Feature Extraction	51
4.4 Model Training	51
4.5 Evaluation Metrics	51
Chapter 5: Results and Discussion	
5.1 Summary of Model Performance	52
5.1.1 Superiority of Pre-trained Language Models	54
5.1.2 Effectiveness of Word Embeddings	54
5.1.3 Architectural Strengths	55
5.1.4 Error Analysis	56
5.1.5 Comparative Analysis	57
5.1 Bangla Author Identification Interface (Powered by BanglaBERT & AABD20)	58
5.2.1 System Overview	59

Chapter 6: Conclusion and Future Work	
6.1 Conclusion	60
6.2 Limitation	61
6.3 Future Work	62
Reference	63

List of Figures

Title	Page	Title	Page
Fig.1. Selected Author List (AABD20)	22	Fig.18. Average Sentence Length	40
Fig.2. Sample Dataset	22	Fig.19. Diminisher Ratio	40
Fig.3. Distribution of samples per author	23	Fig.20. Function Word Ratio	40
Fig.4. Proposed hybrid methodology for Authorship Attribution	26	Fig.21. Conjunction ratio	40
Fig.5. Example of Word-Level Tokenization	27	Fig.22. Negative Word Ratio	40
Fig.6. CBOW architecture of a single word	29	Fig.23. Average Exclamatory Mark	40
Fig.7. CBOW architecture of multi-word	29	Fig.24. Interrogative Mark Count	41
Fig.8. Skip-gram architecture	30	Fig.25. Intensifier Ratio Distribution	41
Fig.9. Architecture of Multi-Layer Perceptron	32	Fig.26. Sentence Count Distribution	41
Fig.10. LSTM Model Architecture	33	Fig.27. Full Stop Count	41
Fig.11. Convolutional Neural Network architecture	35	Fig.28. Modal Ratio by Author	41
Fig.12. Total Word Count	39	Fig.29. Heatmap of TF-IDF	41
Fig.13. Hapax Legomena Ratio	39	Fig.30. Hierarchical Softmax Architecture	45
Fig.14. Stop Word ratio	39	Fig.31. Bi-LSTM Model architecture.	46
Fig.15. Unique Word Count	39	Fig.32. Training and Evaluation Loss Curve	56
Fig.16. Average Word Length	39	Fig.33. Confusion Matrix for BanglaBERT	57
Fig.17 . Type-Token Ratio	39	Fig.34. User Interface for Bangla Author Classification	59

List of Tables

Title	Page	Table	Page
TABLE. I. AABD20 Training Set statistics	24	TABLE. V. BanglaBERT Model Evaluation	49
TABLE. II. Model Performance (%) Across Different Embedding Types	36	TABLE.VI. Model Performance Summary	52
TABLE. III. Model Evaluation (Feature-Based)	43	TABLE. VII. Comparison of the proposed system with existing works in Bangla authorship attribution	58
TABLE. IV. Fine-tuning hyperparameter	48		

Chapter 1

Introduction

1.1 Background

Language is a necessary tool of human communication, enabling individuals to express their concepts, emotions, and opinions to each other in either the spoken or written word. It is a continually evolving system of vocabulary, syntax, and cultural context, evolving over space and time. While varying geographically, country by country, and culture by culture, language and, more specifically, within a language, personal style can similarly vary significantly. These variations are determined by experience, education, and ability in language, and therefore, each person's use of language is individual both in speech and writing. This diversity has resulted in such areas as authorship attribution, the function of which is to determine who authored a given passage of writing from these linguistic and stylistic features.

Authorship Attribution (AA) is a specialized classification task in Natural Language Processing (NLP) that entails identifying the author of an anonymous or disputed document from a set of predetermined candidate authors. AA is all about the principle of identifying the more subtle stylistic and linguistic tendencies present in one's writing at its core. These tendencies, since they are subconscious, are measurable through stylometric, lexical, and syntactic features extracted from their digitized writings.

Each author tends to develop a unique and recognizable writing style, whether consciously or unconsciously, reflected in their choice of vocabulary, syntax, sentence structure, punctuation patterns, and stylistic nuances. Authorship attribution leverages these stylometric, lexical, and linguistic features to distinguish between writing patterns across different individuals.

Despite its broad applicability, authorship attribution remains a challenging task. Several factors complicate this problem: i) the inherent complexity of natural language, ii) similarity of topics across different authors' texts, iii) implicit writing styles that are not easily distinguishable, and iv) structural uniformity in specific genres such as fiction or news articles. These challenges make it difficult to extract useful, author-specific features, especially when building an end-to-end system capable of robust authorship prediction.

1.2 Authorship Attribution in Contemporary Digital Society

Recent developments in Authorship Attribution (AA) have led to a dramatic price surge in value: Life has changed dramatically as social communication has undergone a massive upheaval in cyberspace, allowing for anonymity and pseudonymity. The internet has changed the way humans communicate with each other, share content, and make assertions, sometimes without purposefully disclosing identity. Nowadays, most text-based communications are performed on the internet via social media, online forums, blogs, news comment pages, and updated instant messaging programs. Most of these communications will be anonymous or done with pen names.

The increasing prevalence of anonymous and pseudonymous writing has created opportunities and challenges in many fields, meaning authorship attribution should be viewed not as simply a linguistic or computational task, but rather a useful tool to both inform and mitigate challenges related to security, integrity, accountability, and scholarship. The application areas of authorship attribution are diverse and increasingly vital:

- Forensic investigations: AA is used in forensic linguistics to identify suspects through anonymous writings like threatening letters, ransom notes, or online posts, aiding law enforcement in tracing illicit communications.
- Plagiarism detection: In academic and publishing contexts, AA helps confirm the originality of submissions by detecting instances of copied work, supporting academic and intellectual integrity.
- Cybersecurity and Law Enforcement: AA is crucial in identifying sources of cyber threats such as hate speech, phishing, and disinformation, enabling legal action against malicious actors.
- Literature and History: In digital humanities, AA resolves authorship debates in historical manuscripts and anonymous texts, enriching studies of literary tradition and authorial style.
- Education and Academic Integrity: AA detects ghostwriting and contract cheating in student work, ensuring accountability in academic submissions.
- Content Verification and Accountability in Media: In journalism and media monitoring,

AA verifies the legitimacy or origin of published articles or opinion pieces in circumstances where the content may contain misinformation or manipulation.

Overall, with the vastness of digital communications constantly changing, there is a growing need for robust authorship attribution systems that work across domains, languages, and genres. This drives the demand for scalable, accurate, and interpretable models capable of handling diverse writing styles and linguistic variations.

1.3 Evolution of Author Attribution Techniques

Earlier research in authorship attribution relied heavily on statistical and stylometric methods, which typically used features such as word length, sentence structure, and frequency of function words. While these methods offered initial insights, they often struggled with short texts, lacked semantic understanding, and were largely language-dependent. Among them, one popular method included the use of character n-grams, which proved to be effective in capturing stylistic patterns (Stamatatos, 2009) [1].

Acknowledging the limitations of hand-engineered features, researchers started looking into machine learning-based techniques that enabled the automated extraction of discriminative patterns. Research has been done, for example, Zhang et al. (2014), to define statistical frameworks to model the bag-of-words representations better, resulting in improved classification performance across various text lengths and text domains [2]. Character-level n-gram models also gained popularity for their robustness to spelling variations and ability to capture subtle stylistic patterns, especially in morphologically rich languages [3].

As deep learning has gained traction, AA research is evolving Jafariakinabad et al. (2019) have shown that Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), particularly Long Short-Term Memory (LSTM) networks, can outperform traditional models by learning advanced complex linguistic representations from the data. Such types of architectures, especially when combined with pre-trained word or character embeddings, can be effective at learning local and sequential dependencies in text and generalizing across genres and authors [4].

1.4 Problem Statement

Despite notable progress in authorship attribution (AA) for English and other resource-rich languages, the Bangla (Bengali) language remains significantly underexplored in

computational linguistics, particularly in the domain of author identification. As the 7th most spoken language in the world, Bangla is an Indo-Aryan language widely used in Bangladesh, several Indian states including West Bengal, Assam, and Tripura, and among the global Bangla-speaking diaspora. Nevertheless, Bangla continues to be classified as a low-resource language in the field of Natural Language Processing (NLP), largely due to the scarcity of digital linguistic resources and annotated corpora. All these contribute to the complex written form of Bangla and its extended vocabulary. Bangla lacks significant work in this area due to its high inflection with more than 160 different inflected forms for verbs, 36 other forms for nouns, and 24 different forms for pronouns" (Bhattacharya et al., 2005) [5].

This high degree of inflection complicates the extraction of consistent stylistic and lexical features across authors. The language exhibits a form of diglossia, with a clear distinction between Shadhu Bhasha (the classical, formal register) and Cholito Bhasha (the modern, colloquial form). These two forms differ in terms of vocabulary, syntax, and stylistic norms, posing additional challenges for corpus standardization and model generalization. Lexical choice, syntactic structure, and literary conventions vary widely across periods, literary genres, and individual authors, making it difficult to isolate author-specific stylistic patterns.

One of the primary challenges in advancing authorship attribution for Bangla lies in its status as a low-resource language within the field of Natural Language Processing (NLP). The language suffers from a shortage of high-quality annotated datasets, limited availability of linguistic tools, and a lack of pre-trained models specifically optimized for Bangla. This scarcity of resources significantly constrains the performance and generalizability of both traditional machine learning and modern deep learning approaches in AA tasks.

Moreover, most existing studies on Bangla authorship attribution are limited in scale typically involving a small number of authors and depend heavily on manual feature engineering, such as average word or sentence length, token frequency, and basic stylometric indicators (Das et al., 2015) [6]. These handcrafted methods are not only labor-intensive but also often fail to capture deeper semantic and contextual information.

Consequently, the development of robust and scalable AA systems for Bangla necessitates not only innovative model architectures and feature extraction techniques but also focused efforts to curate, annotate, and expand large-scale linguistic corpora tailored to the complexities of the language.

1.5 Existing Research Gaps

Despite the growing interest in authorship attribution (AA), research in the context of the Bangla language remains limited and fragmented. Only a handful of studies have addressed the problem, and most existing approaches suffer from several key limitations that restrict their applicability and scalability in real-world scenarios.

Firstly, the majority of previous works rely on manually engineered, corpus-dependent features, such as average sentence length, word frequency, and character n-grams (Hossain et al., 2017) [7]. While these features offer initial insights, they often lack robustness and fail to capture the deeper semantic or contextual elements of an author's writing style. These handcrafted techniques are also sensitive to variations in dataset composition and do not generalize well across diverse writing samples.

Secondly, dataset size remains a significant constraint. Most studies focus on small-scale corpora containing fewer than ten authors, often with limited textual samples per author (Islam et al., 2018) [8]. The lack of large, annotated datasets hampers deep learning models, which need ample labeled data for effective training. Existing models also struggle to scale with more authors or handle short, noisy, and varied texts, and are rarely tested in realistic, language-diverse scenarios.

Another notable gap is the absence of hybrid frameworks that integrate both linguistically informed features and deep learning architectures. While deep learning has shown promise in capturing latent stylistic patterns, combining it with stylometric or syntactic insights could significantly enhance performance, especially in low-resource scenarios like Bangla. However, such hybrid methodologies remain largely unexplored in existing literature.

In summary, the current research landscape in Bangla authorship attribution is hindered by (i) reliance on handcrafted features, (ii) limited dataset availability, (iii) poor scalability across authors and domains, and (iv) a lack of hybrid approaches. Addressing these gaps is critical for developing robust, scalable, and generalizable AA systems for Bangla.

1.6 Motivation for Research

Bangla, one of the most widely spoken languages globally, holds a profound literary and cultural legacy, encompassing classical poetry, modern prose, essays, and philosophical writings. Despite this rich tradition, the application of computational techniques, particularly authorship attribution (AA) in Bangla remains significantly underexplored. While languages like English benefit from

decades of robust NLP research and readily available resources, Bangla continues to be treated as a low-resource language, with limited datasets, tools, and adaptable methodologies.

Bangla's morphological richness, stylistic diversity, and diglossic nature (Shadhu and Cholito bhasha) pose unique challenges for authorship attribution. Traditional techniques using handcrafted features and statistical models often struggle to capture these complexities, especially across diverse genres and authorial styles.

Moreover, the lack of large-scale annotated datasets and scalable models has hindered the development of accurate and generalizable AA systems for Bangla. Most existing studies are narrow in scope, involving few authors and limited text samples, and rely heavily on Scopus-specific features, making them unsuitable for broader or real-world applications.

This research is motivated by the pressing need to address several critical challenges in Bangla authorship attribution. It aims to bridge the existing gap in computational approaches by developing a robust and scalable framework capable of effectively handling both classical (Shadhu bhasha) and modern (Cholito bhasha) Bangla texts. A key focus of this study is to minimize dependence on manual feature engineering by leveraging automated feature extraction techniques that can better capture the rich morphological and syntactic complexity of the Bangla language. Additionally, this work seeks to contribute to the broader field of natural language processing by advancing research in low-resource language contexts, with a particular emphasis on improving author identification systems for underrepresented languages like Bangla.

1.7 Research Objectives

The primary goal of this research is to advance the field of authorship attribution (AA) in Bangla by addressing existing limitations in dataset availability, linguistic complexity, and model adaptability. The specific objectives of the study are as follows:

- To curate and expand a high-quality Bangla authorship attribution dataset, incorporating literary works from a diverse range of 20 prominent Bangla authors.
- To develop and evaluate multiple deep learning architectures—including CNN, Bi-LSTM, and LSTM—tailored to capture stylometric, lexical, and syntactic features in Bangla texts.
- To design an ensemble framework that combines deep learning models with traditional machine learning classifiers (e.g., SVM, MLP) for enhanced performance and robustness.
- To incorporate both classical (Shadhu bhasha) and modern (Cholito bhasha) writing styles

in the experimental setup to ensure linguistic and stylistic coverage across different periods and genres.

- To contribute new resources and methodologies for Bangla NLP, promoting further research in low-resource languages by sharing experimental results, insights, and best practices.

Chapter 2

Literature Review

2.1 On Authorship Attribution

Authorship attribution has been a vital area of research for decades, particularly in computational linguistics and stylometry. The underlying justification of this research is that each author possesses a unique, habitual writing style, typically referred to as a homological idiolect, which manifests subconsciously in their written texts. In an effort to identify such stylistic fingerprints, researchers have employed numerous varying feature extraction techniques to identify linguistic patterns at lexical, syntactic, and semantic levels.

Yunita Sari et al. (2018) investigated the effect of dataset properties on the performance of different types of features used in authorship attribution. From their results, style-based features like function words and punctuation are good at topically homogeneous datasets. In contrast, content-based features like lexical or topic-specific words are better suited for topically heterogeneous corpora. They further noted that character n-grams, while commonly used in stylometry, need not necessarily be optimal for any kind of dataset. They then demonstrated that the topic modeling techniques are capable of predicting what class of feature set, either stylistic or content-based, would be most likely to yield better performance on a specific dataset and thus propose a more data-oriented approach to selecting features in authorship attribution issues. [9]

El Bakly, Darwish, and Hefny (2020) suggested a novel structured authorship attribution approach where thematic knowledge is described in terms of ontologies. Their method relies on thematic feature extraction from texts and comparing these features with those of typical authors using measures of similarity. The ontology-based model enables the association of anonymous texts with potential authors based on content and thematic similarity. [10]

Stylometric analysis aims to identify an author's unique writing style using statistical features like grammar and word frequency. Grammatical Markov models help isolate stylistic patterns without relying on content [11].

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert explored topic modeling in authorship attribution of online texts, a comparison between Latent Dirichlet Allocation (LDA) and the Author-Topic (AT) model in generating author representations. As a betterment in attribution accuracy, they also introduced a new model called the Disjoint Author-Document Topic (DADT) model with unique

separation of author and document topic distributions to reduce overlap and increase author discriminability. Their results showed how well the authorial characteristics could be represented using topic-based models, especially in brief, content-focused texts [12]. Kreutz and Daelemans (2018) explored classifier combinations, specifically using Linear SVMs on lexical (word n-grams) and syntactic (PoS n-grams) features with a confidence vote approach, to identify language varieties in subtitles [13].

Sundararajan and Woodard (2018) demonstrated that masking proper nouns significantly improves cross-domain authorship attribution, as these words often introduce strong topic bias. While syntactic features alone were not sufficient for accurate attribution, combining them with selective lexical cues—specifically topic-neutral nouns, verbs, adjectives, and adverbs—yielded stronger performance. Despite these enhancements, character n-gram models continued to outperform other methods, reaffirming their robustness in capturing authorial style [14]. Proceeding with the advancement of deep learning, a large body of work is available on authorship attribution and stylometry using various deep learning models. For example, a multi-headed recurrent neural network character language model was used that outperformed the other methods in PAN 2015.[15]

Some works used syntactic recurrent neural networks, which learn document representations from parts-of-speech tags and then use attention mechanisms to detect the authorial writing style [4]. Convolutional neural networks have also been employed for this task. Impressive performance was achieved by using character-level and multi-channel CNN for large-scale authorship attribution (Ruder et al., 2016). [3]

Shrestha et al. (2017) introduced a novel CNN model leveraging character n-grams for authorship attribution of short texts like tweets, demonstrating competitive performance against existing methods. The research also provides a strategy for model interpretability by analyzing the importance of input text fragments in classification decisions [16]. A CNN model that hides rare words with POS-tags can identify authors based on writing style, even across similar scientific papers, with surprising accuracy and privacy implications [17]. Others investigate syntactic information in the authorship task by building separate language models for each author using part-of-speech tags besides word and character-level information (Fourkioti et al., 2019) [18].

2.2 On Bangla

Despite the significant progress in English and other Western languages for authorship attribution, not much has been done in the Bangla language. A notable body of literature exists on the use of hand-drawn features to extract authorial styles from texts. Tanmoy Chakraborty conducted a

comparative study on authorship attribution in Bengali literature using both statistical and machine learning methods. The study evaluated 450 texts from Rabindranath Tagore, Sarat Chandra Chattopadhyay, and other contemporaries, applying 14 handcrafted stylistic features like word length, punctuation, and dialogue usage. Among the tested models, Support Vector Machine (SVM) achieved the highest accuracy (83.3%), outperforming statistical similarity measures like cosine similarity and chi-square. The study demonstrated that machine learning models offer superior performance over traditional stylometric approaches in Bengali authorship attribution. [19]

Phani et al. (2016) proposed an authorship attribution system for Bengali blog texts using a newly curated corpus of 3,000 passages from three authors. They evaluated lexical features such as character bigrams/trigrams, word n-grams, and Bengali stop words with TF, TF-IDF, and binary representations. The study highlighted the effectiveness of neural models and feature ranking via Information Gain for Bengali authorship attribution [20]. Pal et al. (2017) introduced a machine learning solution to Bangla literary stylometry by investigating Bengali blog text authored by 20 political and educational bloggers. They considered 14 distinctive linguistic features: negative word frequency, sentence length, and word length distribution, suffix usage, and punctuation patterns, as representations of subconscious stylistic traits. Classification performance was measured using Support Vector Machines and Naive Bayes classifiers. Their approach achieved up to 90.74% accuracy for smaller groups of authors, which indicates great potential for author recognition and comparative literary analysis in Bangla. [21]

Hossain, Akter, and Islam (2020) provide a stylometric approach of Bangla authorship attribution using classical machine learning-based classifiers and neural network-based models. The paper compares a collection of features, including lexical and syntactic, to distinguish between the styles of various writers. Experiments were conducted on a handpicked Bangla dataset, and Support Vector Machines, Decision Trees, and Feedforward Neural Networks were employed. Among the models experimented with, neural networks represented competitive accuracy, suggesting their appropriateness in detecting complex linguistic patterns. The results indicate that stylistic features could be combined with neural models to further enhance authorship prediction for Bangla texts, especially in low-resource settings [22].

In recent years, stylometric approaches have gained significant attention for author attribution in Bengali literature. Earlier, Das et al. (2015) spearheaded a pioneering work of stylometry in Bangla literature that included the stylistic features of word and sentence length, type-token ratio, frequency of parts-of-speech, and author-specific distinctive words. Statistical methods like standard deviation

and Jaccard similarity were employed in their research to distinguish between the styles of four prominent Bangladeshi writers. Although they did not perform their work on advanced machine learning classifiers, it laid a good foundation for feature selection for Bangla stylometric work [23].

Das and Mitra (2011) examined authorship identification for Bengali literary texts, specifically for three great writers: Rabindranath Tagore, Bankim Chandra Chattopadhyay, and Sukanta Bhattacharyay. They demonstrated that individual lexical features, unigrams, bigrams, and vocabulary richness, proved to be highly effective in distinguishing the writers. Their system achieved over 90% classification accuracy with respect to unigram features and nearly 100% with bigrams for a sufficiently large training corpus. However, they saw degraded performance with limited training data. The study suggests the potential of even simple stylometric features for robust author identification in Bengali writings [24].

Anisuzzaman and Salam (2018) proposed a hybrid approach to Bengali authorship attribution by combining N-gram models and the Naive Bayes algorithm. Based on three authors—Humayun Ahmed, Rabindranath Tagore, and Shamsur Rahman—the authors employed a home-grown corpus that had over 100,000 words. Both unigram and bigram frequency patterns and Laplace smoothing were used in their approach to estimate probabilities for classes. While the baseline Naive Bayes model did 86%, the fusion model had a performance jump to 95%, demonstrating the strength of combining adjacent word patterns and probabilistic classifiers in such low-resource languages as Bengali. Yet one of the primary limitations of the study may be the fairly modest dataset and limited author population that form the basis for the experiment, potentially affecting the model's power to generalize to a larger literary body or population of writers. [25]

Phani et al. (2017) introduced a supervised learning framework for Bengali authorship attribution using a curated corpus of 3,000 prose samples from Rabindranath Tagore, Sarat Chandra Chattopadhyay, and Bankim Chandra Chattopadhyay. The study employed lexical features—particularly character bigrams—and topic modeling via MALLET, achieving near-perfect accuracy (up to 100%) and outperforming prior methods. However, alternative techniques using flexible pattern models and GloVe embeddings yielded significantly lower performance. This disparity underscores a core limitation: Bengali's low-resource nature limits the availability of rich syntactic and semantic tools, impeding deeper stylometric analysis and hindering the model’s ability to generalize across diverse authorial styles [26].

Chapter 3

Methodology

3.1 Overview

This chapter outlines the comprehensive methodology employed in this thesis to address the problem of Authorship Attribution (AA) for Bengali literary texts. The research adopts a supervised learning paradigm, integrating both conventional machine learning and advanced deep learning techniques. The overarching goal is to develop a robust and accurate system capable of identifying the author of a given Bengali prose sample. This methodology details the entire research pipeline, from data collection and corpus creation to sophisticated text preprocessing, multi-dimensional feature extraction, various word embedding approaches, diverse model training strategies, and a rigorous evaluation framework. Each step is meticulously designed to leverage the unique characteristics of the Bengali language while addressing the challenges associated with its low-resource status in the Natural Language Processing (NLP) domain. The iterative nature of model development and hyperparameter optimization is also emphasized, ensuring the scientific rigor and reproducibility of the findings.

3.2 Data Collection and Corpus Creation

The foundation of any supervised learning task lies in the availability of a high-quality, balanced, and representative dataset. This is particularly true in the domain of authorship attribution, where the model's ability to discern subtle stylistic patterns is critically dependent on the richness and diversity of the training corpus. In the context of Bangla language processing, this challenge is compounded by the scarcity of standard, publicly available benchmark datasets that are both large-scale and author-diverse.

To bridge this gap, this study introduces the “Authorship Attribution Bengali Dataset 20 (AABD20)” a novel, carefully curated corpus designed to advance computational research in Bangla authorship attribution. Unlike prior works that typically relied on limited corpora with 3–6 authors, AABD20 includes 20 distinguished authors from diverse literary periods, genres, and styles. This broad authorial representation enhances the dataset’s complexity and makes it a robust benchmark for evaluating attribution models.

AABD20 was built using a two-phase strategy to ensure broad and deep authorial representation. Phase one focused on selecting prominent prose writers, novelists, essayists, critics, and scholars, excluding poetry to maintain stylistic consistency.

Phase two involved systematic digitization, annotation, and quality control, ensuring diverse, high-quality samples per author. Formatting, metadata, and linguistic integrity were preserved, while genre and temporal variations were included to reflect real-world attribution scenarios.

This approach aimed to simulate realistic AA tasks across themes and time periods and provide a reliable, reusable resource for Bangla NLP. The dataset followed strict linguistic and ethical standards, preserving the authenticity of all texts.

In summary, AABD20 fills a critical void by offering a linguistically rich, author-balanced, and methodologically rigorous dataset suitable for both traditional and deep learning models. It not only supports the development of more accurate authorship identification systems but also contributes to deeper insights into literary style and linguistic nuance in Bangla. AABD20 stands as both a benchmark corpus and a foundational resource for future computational literary studies.

A. Manual Data Collection from PDF Files

A considerable portion of Bengali literature, particularly classical and modern works, is often preserved in digitized PDF formats, commonly accessible through online literary platforms. For this research, the manual data collection phase focused on acquiring prose works in PDF form from two reputable and widely used digital repositories: Boitoi and bdeBooks.

These platforms host a diverse collection of Bengali literary texts, making them ideal for sourcing high-quality prose from selected authors. The collection process involved the following steps:

- **Source Identification:** Boitoi and bdeBooks were chosen due to their extensive and categorized collections of Bengali novels, essays, short stories, series, and articles. Both platforms are known for hosting works by renowned authors and ensuring decent quality in digitization, making them valuable for constructing a reliable corpus.
- **Selection Criteria:** All variation of works, novels, essays, articles, short stories, series, and treatises were selected to maintain consistency in stylistic representation. Poetry and drama were intentionally excluded to avoid genre-specific stylistic features that could introduce noise into the authorship attribution task.
- **Acquisition and Initial Curation:** PDFs were manually downloaded from these platforms after careful verification of authorship and genre. Each file underwent an initial inspection to assess

readability, structural clarity, and the presence of text versus image-based content. Files that were poorly digitized, corrupted, or dominated by scanned images without text recognition were excluded.

- **Text Extraction and Cleaning:** For PDFs with machine-readable text**,** tools such as PyPDF2 and PDFMiner were used for efficient content extraction. For image-based PDFs, Tesseract OCR was employed to convert scanned pages into machine-readable text.

Throughout this process, several challenges emerged, including inconsistent fonts, non-standard character encodings, and the inclusion of non-textual elements such as page numbers, headers, and watermarks. These issues were addressed using custom Python scripts designed to clean and standardize the extracted text while preserving the original sentence structure and authorial style.

B. Automated Data Collection via Custom Web Crawler from Bangla e-Library

To complement the manually collected prose and enhance the diversity of the dataset, an automated data collection approach was adopted targeting the Bangla e-Library, a prominent online repository of Bengali digital texts. This step was essential for expanding the corpus both in volume and authorial variety while maintaining quality and relevance.

- **Tool Selection and Setup**: Data extraction was conducted using the Web Scraper browser extension, configured with custom selectors to extract content from author pages on the Bangla e-Library website.
- **Navigation and Targeting Strategy**: The scraper followed a structured crawl pattern to collect texts, novels, essays, series, short stories, and articles by selected authors. XPath/CSS selectors were used to accurately isolate main text containers, excluding non-relevant site elements.
- **Ethical and Responsible Scraping:** All data collection complied with ethical standards, respecting robots.txt rules and using moderated scraping frequency. Only publicly available content was accessed, with no retrieval of copyrighted or restricted data.

3.2.1 Corpus Formation: Authorship Attribution Bengali Dataset 20 (AABD20)

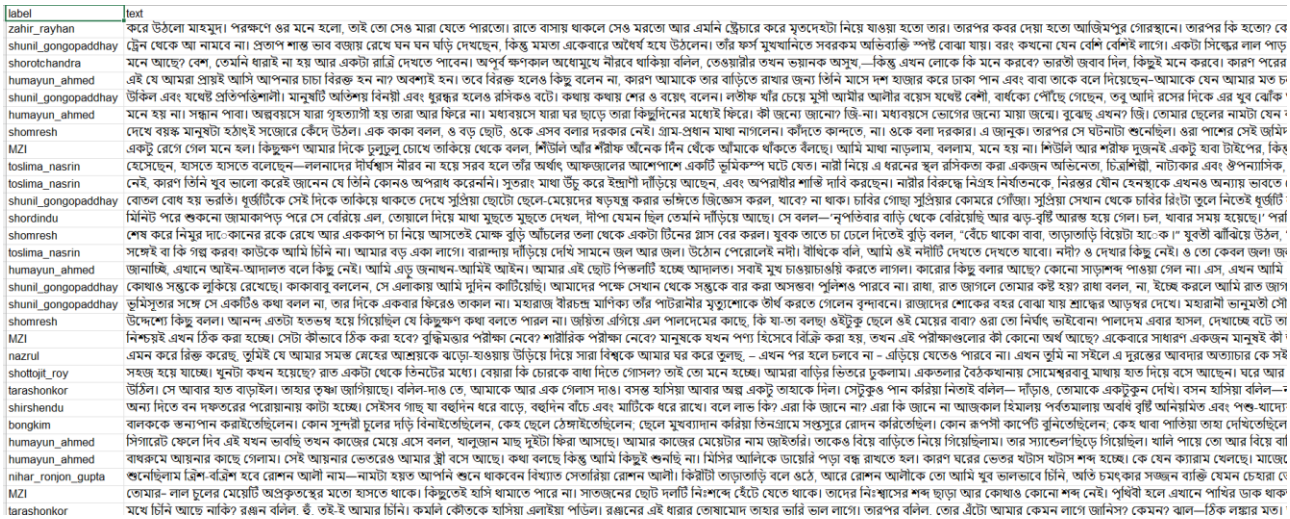
Following the completion of data collection, the next critical phase was the transformation of raw text into a structured and analyzable corpus, formally referred to as the Authorship Attribution Bengali Dataset 20 (AABD20). This process involved defining the corpus architecture, curating clean and consistent text samples to support meaningful authorship attribution tasks.

Author selection is central to the design and effectiveness of any authorship attribution dataset. For AABD20, twenty prominent Bengali literary figures were carefully curated to ensure broad stylistic

variation, temporal diversity, and literary significance. The authors span both classical and modern eras, reflecting various literary movements, narrative techniques, and writing styles. The dataset also captures the diglossia of Bengali, incorporating both Sadhu Bhasha (formal/classical) and Cholito Bhasha (modern/colloquial), as different authors preferred different registers based on era, audience, and context. This adds a valuable layer of linguistic complexity for deeper authorship analysis. The selected author list is shown in Fig.1, and Fig.2 shows the sample dataset.

Zahir Rayhan	Rabindranath Tagore
Kazi Nazrul Islam	Satyajit Ray
Manik Bandopadhyay	Taslima Nasrin
Nihar Ranjan Gupta	Sarat Chandra Chattopadhyay
Bibhutibhushan Bandyopadhyay	Samaresh Majumdar
Bankim Chandra Chatterjee	Sunil Gangopadhyay
Tarasankar Bandyopadhyay	Humayun Ahmed
Sharadindu Bandyopadhyay	Promoth Chowdhury
Shirshendu Mukhopadhyay	Buddhadeb Guha
Muhammed Zafar Iqbal	Akhteruzzaman Elias

Fig.1. Selected Author List for Authorship Attribution Bengali Dataset (AABD20)



label	text
zahir_rayhan	করে উঠলো মাহমুদ। পরক্ষণে ওর মনে হলো, তাই তো সেও মারা যেতে পারতো। রাতে বাসায় থাকলে সেও মরতো আর এমনি ষ্ট্রেচারে করে মৃতদেহটা নিয়ে যাওয়া হতো তার। তারপর কবর দেয়া হতো আজিমপুর গোরস্থানে। তারপর কি হতো? কে
shunil_gongopaddhay	ট্রেন থেকে আ নামবে না। প্রতাপ শান্ত ভাব বজায় রেখে ঘন ঘন ঘড়ি দেখছেন, কিন্তু মমতা একেবারে অধৈর্য হয়ে উঠলেন। তাঁর ফর্স মুখখানিতে সবরকম অভিব্যক্তি স্পষ্ট বোঝা যায়। বরং কখনো যেন বেশি বেশিই লাগে। একটা সিল্কের লাল পাড়
shorotchandra	মনে আছে? বেশ, তেমনি ধারাই না হয় আর একটা রাত্রি দেখতে পাবেন। অপূর্ব ক্ষণকাল অধোমুখে নীরবে থাকিয়া বলিল, তেওয়ারীর তখন ভয়ানক অসুখ,—কিন্তু এখন লোকে কি মনে করবে? ভারতী জবাব দিল, কিছুই মনে করবে। কারণ পরের
humayun_ahmed	এই যে আমরা প্রায়ই আসি আপনার চাচা বিরক্ত হন না? অবশ্যই হন। তবে বিরক্ত হলেও কিছু বলেন না, কারণ আমাকে তার বাড়িতে রাখার জন্য তিনি মাসে দশ হাজার করে ঢাকা পান এবং বাবা তাকে বলে দিয়েছেন-আমাকে যেন আমার মত চ
shunil_gongopaddhay	উকিল এবং যথেষ্ট প্রতিপত্তিশালী। মানুষটি আতিশয় বিনয়ী এবং ধুরন্ধর হলেও রসিকও বটে। কথায় কথায় শের ও বয়েৎ বলেন। লতীফ খাঁর চেয়ে মুন্সী আমীর আলীর বয়েস যথেষ্ট বেশী, বার্ধক্যে পৌঁছে গেছেন, তবু আদি রসের দিকে এর খুব ঝোঁক
humayun_ahmed	মনে হয় না। সন্ধান পাবা। অল্পবয়সে যারা গৃহত্যাগী হয় তারা আর ফিরে না। মধ্যবয়সে যারা ঘর ছাড়ে তারা কিছুদিনের মধ্যেই ফিরে। কী জন্যে জানো? জ্বি-না। মধ্যবয়সে ভোগের জন্যে মায়া জন্মে। বুঝেছ এখন? জ্বি। তোমার ছেলের নামটা যেন
shomresh	দেখে বয়স্ক মানুষটা হঠাৎই সজোরে কেঁদে উঠল। এক কাকা বলল, ও বড় ছোট, ওকে এসব বলার দরকার নেই। গ্রাম-প্রধান মাথা নাগলেন। কাঁদতে কাঁদতে, না। ওকে বলা দরকার। এ জানুক। তারপর সে ঘটনাটা শুনেছিল। ওরা পাশের সেই জমি
MZI	একটু রেগে গেল মনে হল। কিছুক্ষণ আমার দিকে টুলটুল চোখে তাকিয়ে থেকে বলল, শিউলি আর শরীফ অনেক দিন থেকে আমাকে থাকতে বলছে। আমি মাথা নাড়লাম, বললাম, মনে হয় না। শিউলি আর শরীফ দুজনই একটু হাবা টাইপের, কিন্তু
toslima_nasrin	হেসেছেন, হাসতে হাসতে বলেছেন—ললনাদের দীর্ঘশ্বাস নীরব না হয়ে সরব হলে তাঁর অর্থাৎ আফজালের আশেপাশে একটি ভূমিকম্প ঘটে যেত। নারী নিয়ে এ ধরনের স্থূল রসিকতা করা একজন অভিনেতা, চিত্রশিল্পী, নাট্যকার এবং ঔপন্যাসিক,
toslima_nasrin	নেই, কারণ তিনি খুব ভালো করেই জানেন যে তিনি কোনও অপরাধ করেননি। সুতরাং মাথা উঁচু করে ইন্দ্রাণী দাঁড়িয়ে আছেন, এবং অপরাধীর শাস্তি দাবি করছেন। নারীর বিরুদ্ধে নিগ্রহ নির্যাতনকে, নিরন্তর যৌন হেনস্থাকে এখনও অন্যায় ভাবতে
shunil_gongopaddhay	বোতল বোধ হয় ভরতি। ধূর্জটিকে সেই দিকে তাকিয়ে থাকতে দেখে সুপ্রিয়া ছোটো ছেলে-মেয়েদের ষড়যন্ত্র করার ভঙ্গিতে জিজ্ঞেস করল, খাবে? না থাক। চাবির গোছা সুপ্রিয়ার কোমরে গোঁজা। সুপ্রিয়া সেখান থেকে চাবির রিংটা তুলে নিতেই ধূর্জটি
shordindu	মিনিট পরে শুকনো জামাকাপড় পরে সে বেরিয়ে এল, তোয়ালে দিয়ে মাথা মুছতে মুছতে দেখল, দীপা যেমন ছিল তেমনি দাঁড়িয়ে আছে। সে বলল—'নৃপতিবার বাড়ি থেকে বেরিয়েছি আর ঝড়-বৃষ্টি আরম্ভ হয়ে গেল। চল, খাবার সময় হয়েছে।' পরা
shomresh	শেষ করে নিমুর দাকানের রকে রেখে আর এককাপ চা নিয়ে আসতেই মোক্ষ বুড়ি আঁচলের তলা থেকে একটা টিনের গ্লাস বের করল। যুবক তাতে চা ঢেলে দিতেই বুড়ি বলল, "বেঁচে থাকো বাবা, তাড়াতাড়ি বিয়েটা হাকে।" যুবতী ঝাঁঝিয়ে উঠল,
toslima_nasrin	সঙ্গেই বা কি গল্প করবা কাউকে আমি চিনি না। আমার বড় একা লাগে। বারান্দায় দাঁড়িয়ে দেখি সামনে জল আর জল। উঠোন পেরোলেই নদী। বাঁধিকে বলি, আমি ওই নদীটি দেখতে দেখতে যাবো। নদী? ও দেখার কিছু নেই। ও তো কেবল জলা জ
humayun_ahmed	জানাচ্ছি, এখানে আইন-আদালত বলে কিছু নেই। আমি এড়ু জনাথন-আমিই আইন। আমার এই ছোট পিঞ্জলটি হচ্ছে আদালত। সবাই মুখ চাওয়াচাওয়ি করতে লাগল। কারোর কিছু বলার আছে? কোনো সাড়াশব্দ পাওয়া গেল না। এস, এখন আমি
shunil_gongopaddhay	কোথাও সন্তুকে লুকিয়ে রেখেছে। কাকাবাবু বললেন, সে এলাকায় আমি দুদিন কাটিয়েছি। আমাদের পক্ষে সেখান থেকে সন্তুকে বার করা অসম্ভব। পুলিশও পারবে না। রাধা, রাত জাগলে তোমার কষ্ট হয়? রাধা বলল, না, ইচ্ছে করলে আমি রাত জাগ
shunil_gongopaddhay	ভূমিসূতার সঙ্গে সে একটিও কথা বলল না, তার দিকে একবার ফিরেও তাকাল না। মহারাজ বীরচন্দ্র মাণিক্য তাঁর পাটরানীর মৃত্যুশোকে তীর্থ করতে গেলেন বৃন্দাবনে। রাজাদের শোকের বহর বোঝা যায় শ্রাদ্ধের আড়ম্বর দেখে। মহারানী ভানুমতী সৌ
shomresh	উদ্দেশ্যে কিছু বলল। আনন্দ এতটা হতভম্ব হয়ে গিয়েছিল যে কিছুক্ষণ কথা বলতে পারল না। জয়িতা এগিয়ে এল পালদেমের কাছে, কি যা-তা বলছ ওইটুকু ছেলে ওই মেয়ের বাবা? ওরা তো নির্ঘাৎ ভাইবোনা পালদেম এবার হাসল, দেখাচ্ছে বটে তা
MZI	নিশ্চয়ই এখন ঠিক করা হচ্ছে। সেটা কীভাবে ঠিক করা হবে? বুদ্ধিমত্তার পরীক্ষা নেবে? শারীরিক পরীক্ষা নেবে? মানুষকে যখন পণ্য হিসেবে বিক্রি করা হয়, তখন এই পরীক্ষাগুলোর কী কোনো অর্থ আছে? একেবারে সাধারণ একজন মানুষই কী
nazrul	এমন করে রিক্ত করেছ, তুমিই যে আমার সমস্ত স্নেহের আশ্রয়কে ঝড়ো-হাওয়ায় উড়িয়ে দিয়ে সারা বিশ্বকে আমার ঘর করে তুলছ, – এখন পর হলে চলবে না – এড়িয়ে যেতেও পারবে না। এখন তুমি না সইলে এ দুরন্তের আবদার অত্যাচার কে সই
shottojit_roy	সহজ হয়ে যাচ্ছে। খুনটা কখন হয়েছে? রাত একটা থেকে তিনটের মধ্যে। বেয়ারা কি চোরকে বাধা দিতে গোসল? তাই তো মনে হচ্ছে। আমরা বাড়ির ভিতরে ঢুকলাম। একতলার বৈঠকখানায় সোমেশ্বরবাবু মাথায় হাত দিয়ে বসে আছেন। ঘরে আর
tarashonkor	উঠিল। সে আবার হাত বাড়াইল। তাহার তৃষ্ণা জাগিয়াছে। বলিল-দাও তে, আমাকে আর এক গেলাস দাও। বসন্ত হাসিয়া আবার অল্প একটু তাহাকে দিল। সেটুকুও পান করিয়া নিতাই বলিল— দাঁড়াও, তোমাকে একটুকুন দেখি। বসন হাসিয়া বলিল—ন
shirshendu	অন্য দিতে বন দফতরের পরোয়ানায় কাটা হচ্ছে। সেইসব গাছ যা বহুদিন ধরে বাড়ে, বহুদিন বাঁচে এবং মাটিকে ধরে রাখে। বলে লাভ কি? এরা কি জানে না? আজকাল হিমালয় পর্বতমালায় অবাধ বৃষ্টি অনিয়মিত এবং পশু-খাদ্য
bongkim	বালককে স্তন্যপান করাইতেছিলেন। কোন সুন্দরী চুলের দড়ি বিনাইতেছিলেন, কেহ ছেলে ঠেঙ্গাইতেছিলেন, ছেলে মুখব্যাদান করিয়া তিনগ্রামে সপ্তসুরে রোদন করিতেছিল। কোন রূপসী কার্পেট বুনিতেছিলেন; কেহ ধাবা পাতিয়া তাহা দেখিতেছিলে
humayun_ahmed	সিগারেট ফেলে দিব এই যখন ভাবছি তখন কাজের মেয়ে এসে বলল, খালুজান মাছ দুইটা ফিরা আসছে। আমার কাজের মেয়েটার নাম জাইতারি। তাকেও বিয়ে বাড়িতে নিয়ে গিয়েছিলাম। তার স্যান্ডেল ছিঁড়ে গিয়েছিল। খালি পায়ে তো আর বিয়ে বা
humayun_ahmed	বাথরুমে আয়নার কাছে গেলাম। সেই আয়নার ভেতরেও আমার স্ত্রী বসে আছে। কথা বলছে কিন্তু আমি কিছুই শুনছি না। মিসির আলিকে ডায়েরি পড়া বন্ধ রাখতে হল। কারণ ঘরের ভেতর খটাস খটাস শব্দ হচ্ছে। কে যেন ক্যারাম খেলছে। মাজে
nihar_ronjon_gupta	শুনেছিলাম ত্রিশ-বত্রিশ হবে রোশন আলী নাম—নামটা হয়ত আপনি শুনে থাকবেন বিখ্যাত সেতারিয়া রোশন আলী। কিরীটী তাড়াতাড়ি বলে ওঠে, আরে রোশন আলীকে তো আমি খুব ভালভাবে চিনি, আতি চমৎকার সজ্জন ব্যক্তি যেমন চেহারা
MZI	তোমার– লাল চুলের মেয়েটি অপ্রকৃতস্থের মতো হাসতে থাকে। কিছুতেই হাসি থামাতে পারে না। সাতজনের ছোট দলটি নিঃশব্দে হেঁটে যেতে থাকে। তাদের নিঃশ্বাসের শব্দ ছাড়া আর কোথাও কোনো শব্দ নেই। পৃথিবী হলে এখানে পাখির ডাক থাক
tarashonkor	মখে চিনি আছে নাকি? রঞ্জন বলিল. হুঁ. তই-ই আমার চিনি। কমলি কৌতকে হাসিয়া এলাইয়া পডিল। রঞ্জনের এই ধারার তোষামোদ তাহার ভারি ভাল লাগে। তারপর বলিল. তোর এঁটো আমার কেমন লাগে জানিস? কেমন? ঝাল—ঠিক লঙ্কার মত।

Fig.2. Sample dataset

This diverse set of authors allows for a more challenging and realistic evaluation of authorship attribution models, requiring the systems to capture subtle stylistic and lexical distinctions across a wide range of types. Furthermore, the inclusion of both male and female authors, and those from both Indian and Bangladeshi literary traditions, helps ensure linguistic richness and cultural breadth.

A. Text Segmentation and Sample Generation

To facilitate consistent analysis and enable machine learning-based classification, the collected texts were segmented into fixed-length units. Each author's compiled prose corpus was divided into disjoint text samples of approximately 750 words each. This fixed-length sampling ensured uniformity across the dataset and was chosen to balance the trade-off between stylistic sufficiency and computational efficiency. Creating disjoint samples also mitigates the risk of data leakage between training, validation, and test sets, thereby preserving the validity of model evaluations. Unlike benchmark datasets with enforced class balance, AABD20 was intentionally built with an imbalanced class distribution to reflect real-world authorship scenarios. In practice, authors vary in output; some have extensive works, and others only a few pieces. This natural imbalance presents a more realistic challenge for model generalization and robustness. As a result, sample counts per author vary based on the availability and length of their digitized texts.

The final corpus was organized as a two-column structure, where:

- The first column represents the author's name (i.e., the label),
- The second column contains a text sample of 750 words.

This minimalist structure was adopted to streamline the data pipeline for training and evaluation, while still retaining the essential information required for supervised learning in the authorship attribution task. The Fig.3 demonstrates the sample distribution of text per author in our dataset.

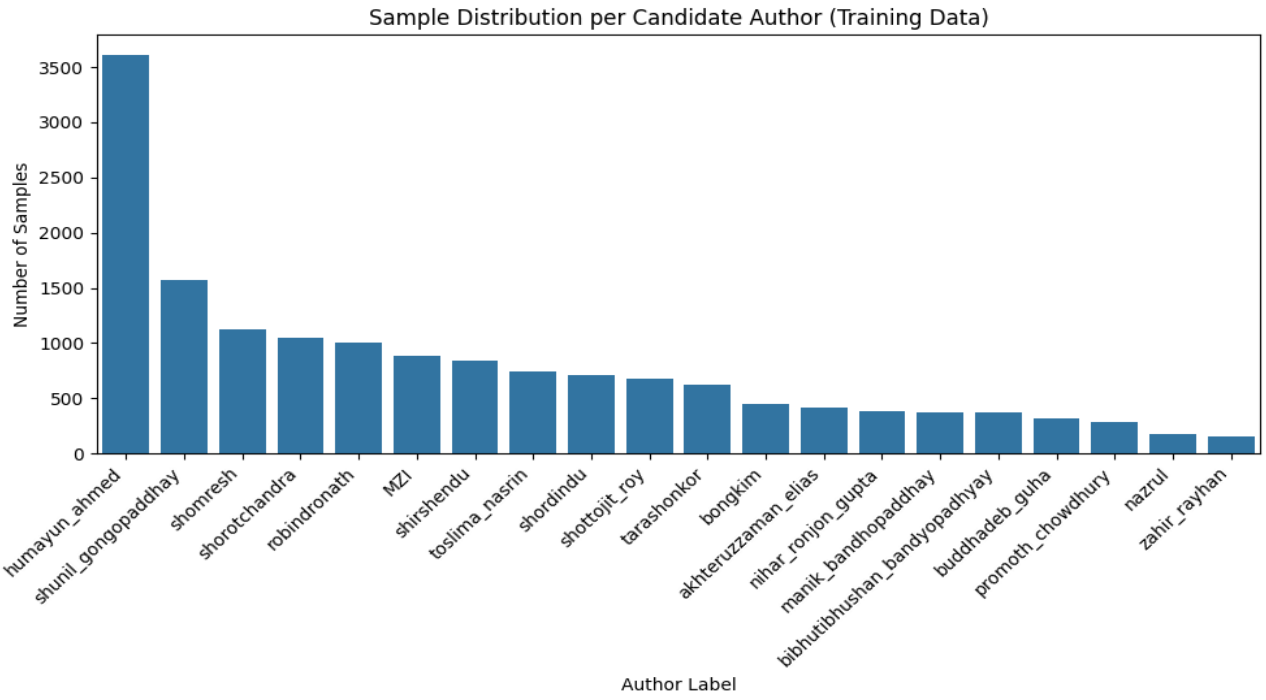


Fig.3. Distribution of samples per author

B. Data Storage and Management

The AABD20 corpus was stored in a structured CSV format for efficient processing, experimentation, and reproducibility. Each row includes the author's name (class label) and a 750-word prose sample, compatible with libraries like Pandas, scikit-learn, PyTorch, and TensorFlow for easy loading, filtering, and stratified splitting.

TABLE I. AABD20 Training Set statistics

Index	Sample Count	Total Word Count	Unique Word Count
MZI	880	659,388	40,252
Akhteruzzaman Elias	416	311,168	40,778
Bibhutibhushan Bandyopadhyay	369	274,554	43,336
Bongkim	450	336,745	46,587
Buddhadeb Guha	315	230,599	33,765
Humayun Ahmed	3,612	2,704,705	124,802
Manik Bandhopaddhay	376	281,915	33,691
Nazrul	179	134,005	25,010
Nihar Ronjon Gupta	381	285,374	33,331
Promoth Chowdhury	288	214,205	41,763
Robindronath	1,007	753,072	69,633
Shirshendu	838	628,451	52,838
Shomresh	1,126	844,353	53,462
Shordindu	711	532,818	67,344
Shorotchandra	1,051	788,077	58,394
Shottojit Roy	680	509,528	51,596
Shunil Gongopaddhay	1,570	1,177,151	80,812
Tarashonkor	620	464,673	63,746
Toslima Nasrin	745	557,691	57,129
Zahir Rayhan	148	110,958	16,604

To ensure reproducibility and future updates, the corpus was organized with standard file management and documented preprocessing steps. While Git LFS wasn't used due to size and format

constraints, internal backups and data snapshots were maintained. Table I, present the training set statistics

In line with open science principles, the AABD20 dataset is planned for public release to support reproducibility, collaboration, and advancement in Bengali authorship attribution research. The dataset will be hosted on an open-access digital repository with comprehensive documentation, including metadata, preprocessing scripts, annotation guidelines, and example use cases for easy integration into experimental workflows.

A clear license and citation protocol will accompany the release to ensure proper attribution and ethical use. AABD20 is intended to serve as both a benchmarking tool and a foundation for future research, educational projects, and model development in Bangla NLP.

Through this initiative, AABD20 aims to serve not only as a benchmarking tool but also as a catalyst for new research directions, educational projects, and the development of novel models that advance the state of the art in Bangla NLP.

3.3 Proposed Methodology

The proposed methodology, as illustrated in Fig.4 for author identification, is built upon a robust hybrid framework that meticulously integrates both classical and cutting-edge Natural Language Processing (NLP) techniques. Recognizing the inherent complexity of identifying authors from their written work, our approach transcends simple keyword analysis, delving into the subtle stylistic nuances that define an individual's writing signature. From the foundational stages of data acquisition and meticulous preprocessing to the sophisticated application of deep learning architectures and feature engineering, each component is designed to enhance the model's ability to discern authorship. This hybrid paradigm leverages the interpretability and efficiency of conventional machine learning models alongside the powerful pattern recognition capabilities of neural networks, including a specialized focus on low-resource language processing through fine-tuned pre-trained models. The ultimate goal is to provide a highly accurate and adaptable system capable of identifying possible authors with a high degree of confidence across diverse textual data.

Text preprocessing serves as the foundational step in any natural language processing (NLP) pipeline. For the task of Bengali author attribution, this step is especially crucial, as it ensures that raw textual inputs are cleaned, standardized, and prepared for downstream tasks like embedding generation and model training. Given the morphologically rich and orthographically complex nature of Bengali, careful preprocessing is essential to retain the distinctive linguistic signals that may differentiate authors' writing styles.

The collected Bengali literary texts often contain non-standard characters, punctuation marks, and irregular spacing due to their diverse sources. To address this, a custom cleaning function was applied that focuses specifically on preserving the Bengali script while removing unwanted noise. The following cleaning operations were implemented:

- **Character Filtering:** Removed all characters except Bengali (\u0980–\u09FF), English letters, digits, and whitespace using regex to eliminate emojis and special symbols.
- **Whitespace Normalization:** Multiple consecutive whitespaces collapsed into a single space. This helped eliminate inconsistencies caused by irregular formatting or line breaks in the raw data.

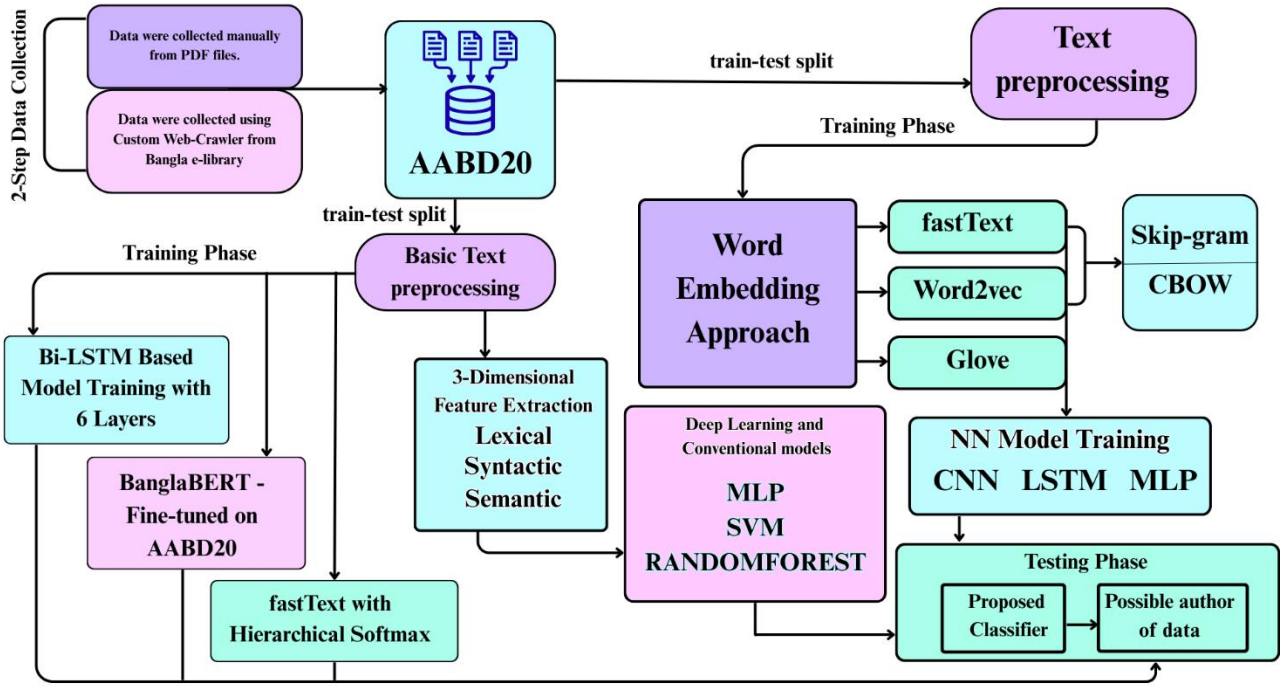


Fig.4. Proposed Hybrid Methodology for Authorship Attribution

This preprocessing pipeline retained both Bengali and English text components, as occasional code-switching between Bengali and English can appear in literary content and might contribute to authorial style.

After the cleaning step, all text data was tokenized using the word_tokenize() function from the Natural Language Toolkit (NLTK). This tokenization approach breaks down the cleaned text into individual word-level units (tokens), which is a prerequisite for generating word embeddings and extracting linguistic features. Tokenization was applied consistently across all models and embedding strategies. Fig.5 shows a word-level tokenization example.

Fig.5. Example Of Word-Level Tokenization

3.3.1 Word Embeddings Approach with NN Model Training

Word embedding is a technique that represents words as dense vectors in a continuous space, capturing semantic and syntactic relationships. First introduced by Bengio et al. (2003) in their work "A Neural Probabilistic Language Model" [27], it marked a shift from traditional isolated word representations to distributed ones learned through neural networks. In these embeddings, meaning is encoded across multiple dimensions, enabling computational models to process words, phrases, or sentences more effectively, even though individual dimensions may not be directly interpretable.

A key property of word embeddings is their geometric structure: semantically similar words are located close to each other in the vector space. For example, "king" and "queen" or "walk" and "run" are positioned nearby due to shared semantic traits. This spatial proximity reflects meaningful relationships between words.

Moreover, embeddings support "vector arithmetic" or semantic operations. A well-known example vector("King") - vector("Man") + vector("Woman") ≈ vector("Queen") demonstrates how embeddings encode complex analogical relationships, not just similarity. This relational capacity enables advanced AI systems to reason about language in a way that goes beyond surface-level associations.

In this study, word embeddings are used to convert Bengali text into machine-readable form, allowing models to capture linguistic patterns effectively. Unlike sparse representations like one-hot encoding or term-frequency matrices, embeddings retain syntactic and semantic similarities, enhancing model generalization even with lexical variation.

The research explores several embedding methods: prediction-based models like Word2Vec and FastText, and count-based models like GloVe. Each leverages the distributional hypothesis that words appearing in similar contexts tend to have similar meanings to learn meaningful vector representations.

Prediction-Based Embedding: Prediction-based embeddings are trained by solving a supervised learning task in which the model learns to predict a word based on its context or vice versa. The core assumption is that understanding the context around a word provides valuable information about its semantic properties. These models use shallow neural networks that are optimized to minimize a specific loss function, such as negative log-likelihood or cross-entropy, to fine-tune the word vectors. They capture not only co-occurrence statistics but also directional and position-aware contextual semantics, which is particularly advantageous in natural language understanding tasks like authorship attribution.

The key techniques under this category include CBOW and Skip-gram architecture, implemented through Word2Vec and FastText. The GloVe framework, mentioned above, uses a third type of technology.

1. **Continuous Bag of Words:** The Continuous Bag-of-Words (CBOW) model, introduced by Mikolov et al. (2013), predicts the target word given a fixed-size window of surrounding context words [28]. The input context words are projected into their embedding vectors and averaged, and this composite vector is used to predict the target word. The CBOW model tries to predict the probability of a word, according to its context. Representation of the context is like a bag of the contained words, in a fixed-sized window, around the target word. The example below will illustrate this concept further. If a corpus C represents the text" Hey, this is a sample corpus using only one context word, given the context window is set to be 1, the corpus can be configured in the following way to be a training set. So, the target for the data point will appear a lot like this. The base concept of multi-word architecture remains the same, but architecture is a bit more complex. The diagram of the single-word and multi-word CBOW model architecture is presented below, respectively in Fig.6 and Fig.7. Hence, CBOW basically is like predicting a word if the context is given.
2. **Skip-Gram***:* The Skip-gram model, a widely used prediction-based word embedding technique, was first introduced by Tomas Mikolov et al. in their influential work titled "Efficient Estimation of Word Representations in Vector Space" [28]. This model was part of the broader Word2Vec framework, which aimed to learn high-quality vector representations of words from large datasets by predicting the context of a given word. The Skip-gram architecture specifically seeks to maximize the probability of surrounding context words given a target word, capturing both semantic and syntactic patterns effectively. The Skip-Gram architecture is the inverse of CBOW.

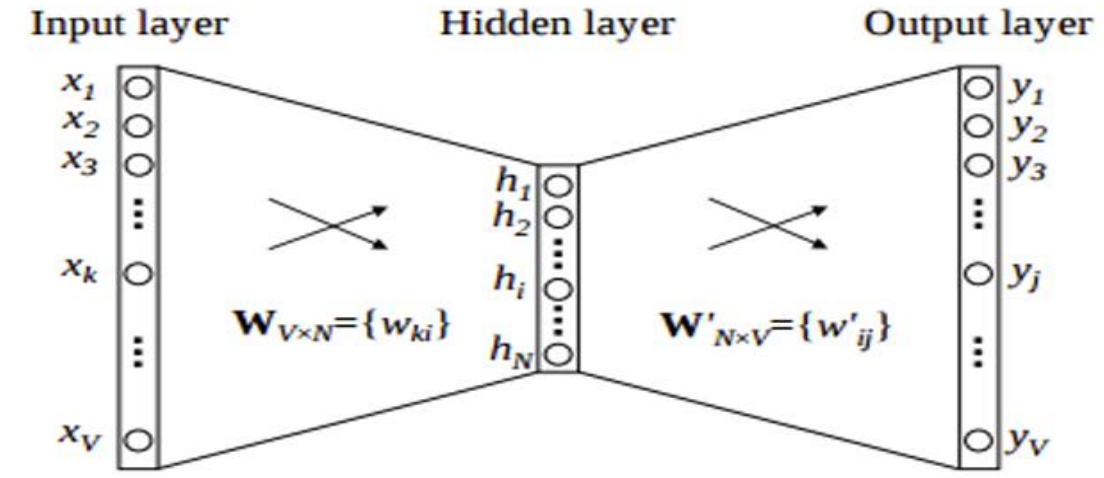


Fig.6. CBOW architecture of a single word

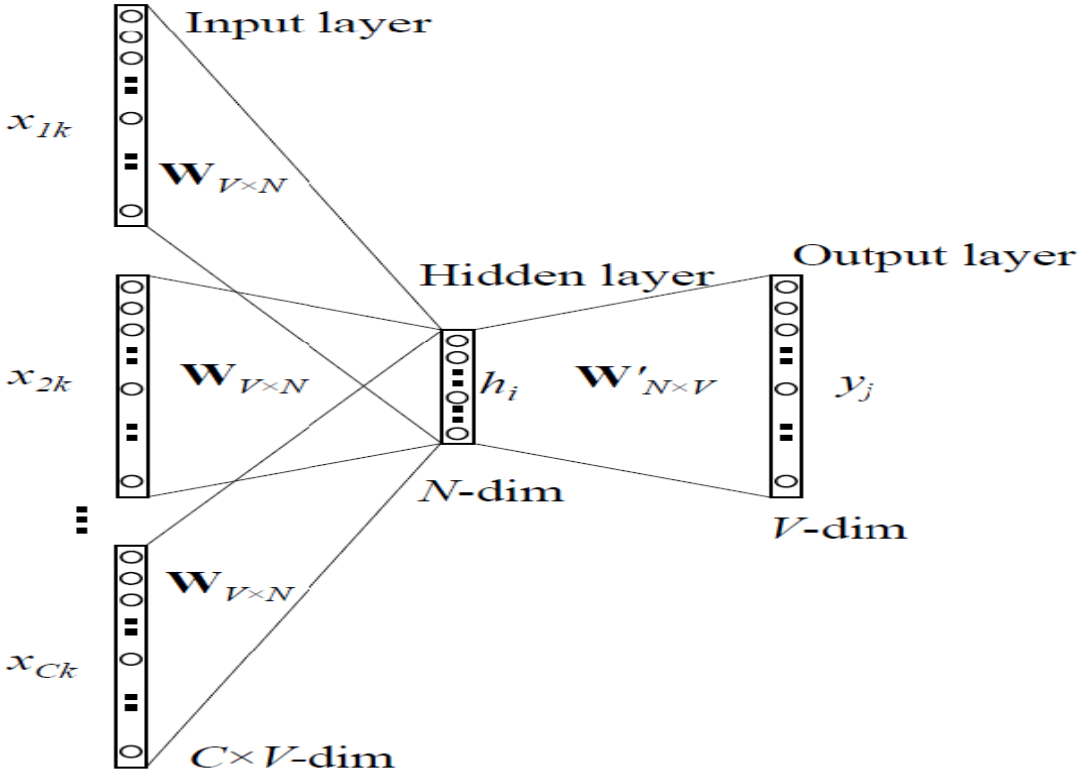


Fig.7. CBOW architecture of multi-words

Unlike CBOW, which predicts a target word from its context, Skip-Gram predicts the context given a word, assigning less weight to distant words through random sampling. The window size parameter defines a maximum size, but the actual context window varies randomly between 1 and this

maximum. For a window size of 1, both models generate two one-hot target vectors with corresponding outputs, producing two error vectors that are summed element-wise into a final error vector. After training, the weights between the input and hidden layers serve as the word embeddings. The loss function is essentially the same as in CBOW. The Fig.8 illustrates the Skip-gram architecture.

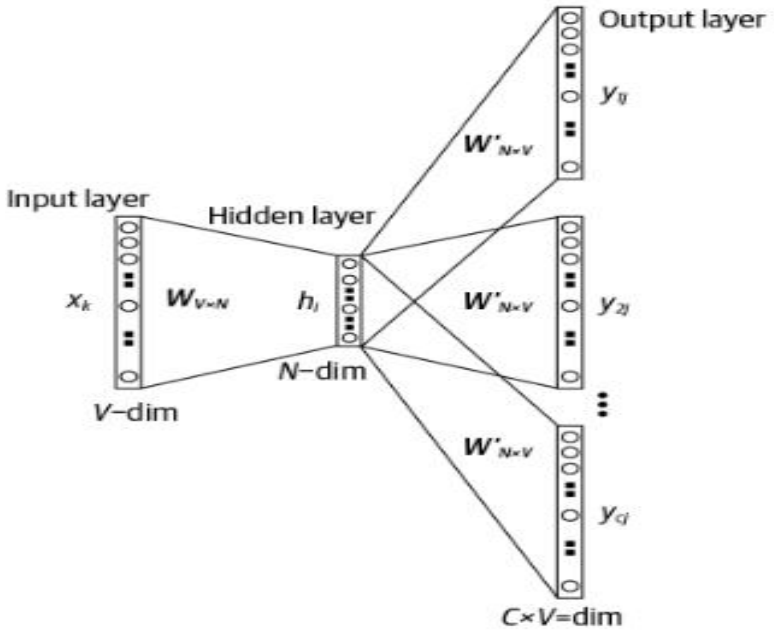


Fig.8. Skip-gram architecture

In this study, both CBOW and Skip-Gram implementations of Word2Vec were trained on the Bengali dataset to evaluate the impact of each architecture on the downstream authorship attribution task.

3. **GloVe***:* The GloVe model, short for Global Vectors for Word Representation, was introduced by Pennington et al. (2014**)** as a count-based word embedding technique that captures both local context and global statistical information of a corpus [29]. Unlike the prediction-based Skip-gram and CBOW models, GloVe constructs a large word-co-occurrence matrix and then factorizes it to obtain word vectors. This approach allows GloVe to encode meaningful linear substructures, such as word analogies, more effectively and robustly. It then factorizes this matrix to produce word embeddings that reflect the ratios of co-occurrence probabilities between words, capturing both local and global statistical information.

For our analysis purposes, we used the following embeddings for training on our dataset, each consisting of 100 dimensions:

- CBOW model by Word2Vec
- Skip-Gram model by Word2Vec

- CBOW model by fastText
- Skip-Gram model by fastText
- Word Embeddings by GloVe

In generating the word-level embeddings based on the 5 models mentioned above, we used the following parameters for all 5 of the models, for fair experimentation.

- Vector Dimensions of length 100.
- Context window of 5.
- 5 learning iterations

Neural Network Model Architectures and Training

Deep learning architectures have shown strong performance in NLP tasks due to their ability to automatically learn high-level features from raw text (Young et al., 2018) [30]. In this study, we apply three widely used neural models, CNN, MLP, and LSTM, for Bengali authorship attribution, utilizing pre-trained static word embeddings: GloVe, Word2Vec (CBOW and Skip-Gram), and FastText (CBOW and Skip-Gram). These embeddings remain fixed during training to preserve consistent linguistic patterns. To capture both local and sequential features from diverse literary styles, models were trained on processed Bengali texts tokenized into fixed-length sequences of 2000 tokens. A softmax output layer enabled multi-class classification, with each class representing an author. Optimization used categorical cross-entropy loss and the Adam optimizer, addressing class imbalance via weighting or balanced splits.

Embeddings were evaluated using three neural architectures: Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN), with each model trained separately on five embedding types mentioned above.

Before feeding the embeddings into the neural networks, the tokenized text data was processed:

Mean Vector for MLP: Each document's tokens were averaged into a single fixed-size vector of word embeddings, yielding a document-level representation.

Padded Sequences for LSTM and CNN: Tokens were mapped to numerical indices and padded to a uniform length of 750. Both models used a non-trainable embedding layer initialized with pre-trained embeddings to preserve learned features.

1. Multi-layer Perceptron (MLP)

The Multi-layer Perceptron model serves as a baseline deep learning classifier, focusing on capturing non-linear relationships among features extracted from text. Here, the entire embedded sequence is flattened into a single feature vector, discarding word order information but retaining all feature

values.

The Multi-layer Perceptron (MLP) model is utilized to capture complex non-linear patterns from fixed-length vector representations of literary texts. Specifically, each document is transformed into a mean word embedding vector, where the embedding dimension is 100. This approach flattens the sequential structure but retains the overall semantic content of the text, making it suitable for feedforward neural networks. Fig.9 demonstrates the architecture of a Multi-layer Perceptron with two hidden layers.

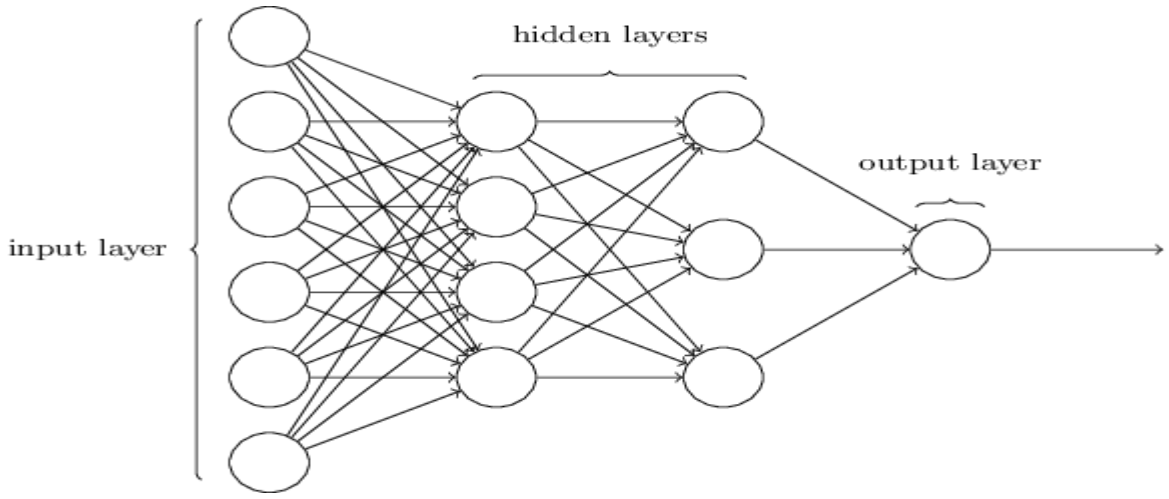


Fig.9. Architecture of Multi-Layer Perceptron

The MLP model consists of fully connected layers that learn hierarchical features via non-linear transformations. Optional dropout helps prevent overfitting, but is minimal here. Although it does not capture sequential dependencies, MLP effectively detects statistical patterns and high-level abstractions from embeddings, highlighting the discriminative power of word-level feature distributions.

Model Architecture

Input Layer: Accepts a 100-dimensional input vector, representing the meaning of all word embeddings in a document.

Hidden Layer 1: Consists of 128 neurons with ReLU (Rectified Linear Unit) activation, enabling the model to learn complex feature interactions.

Hidden Layer 2: Contains 64 neurons, also activated using ReLU, allowing further abstraction of the learned features.

Output Layer: A dense layer with neurons equal to the number of author classes, followed by a Softmax activation function for multi-class classification.

Training Configuration

Optimizer: Adam optimizer was employed for its adaptive learning capabilities and computational efficiency.

Loss Function: Categorical Cross-Entropy, suitable for multi-class classification tasks.

Evaluation Metric: Model performance was tracked using Accuracy.

Epochs: The model was trained for 10 epochs, sufficient for convergence on the dataset.

Batch Size: A mini-batch size of 16 was used to balance training efficiency and generalization.

This MLP setup acts as a non-sequential baseline model, highlighting how author-specific patterns can still be captured effectively without modeling temporal dependencies.

2. Long Short-Term Memory (LSTM) Network

The Long Short-Term Memory (LSTM) network is a specialized form of Recurrent Neural Network (RNN) capable of learning long-term dependencies in sequential data. Originally introduced by Hochreiter and Schmidhuber [31], LSTMs address the limitations of standard RNNs by incorporating memory cells and gating mechanisms, which allow the network to retain relevant information over extended input sequences. This makes LSTM particularly suitable for tasks such as author attribution, where the linguistic and stylistic cues that differentiate authors may span across multiple sentences or even paragraphs.

In the context of Bengali literary prose, such long-range dependencies are critical. Bengali texts often exhibit complex syntactic structures, rich narrative forms, and author-specific stylistic choices that unfold gradually across sentences. Unlike traditional models such as Multi-Layer Perceptron's that treat text as bag-of-words representations or Convolutional Neural Networks (CNNs) that focus on local n-gram features, LSTMs preserve the sequential nature of language. They enable the model to understand not only which words are used but also how and where they occur in the text, which is essential for capturing deeper semantic and stylistic patterns.

Model Architecture

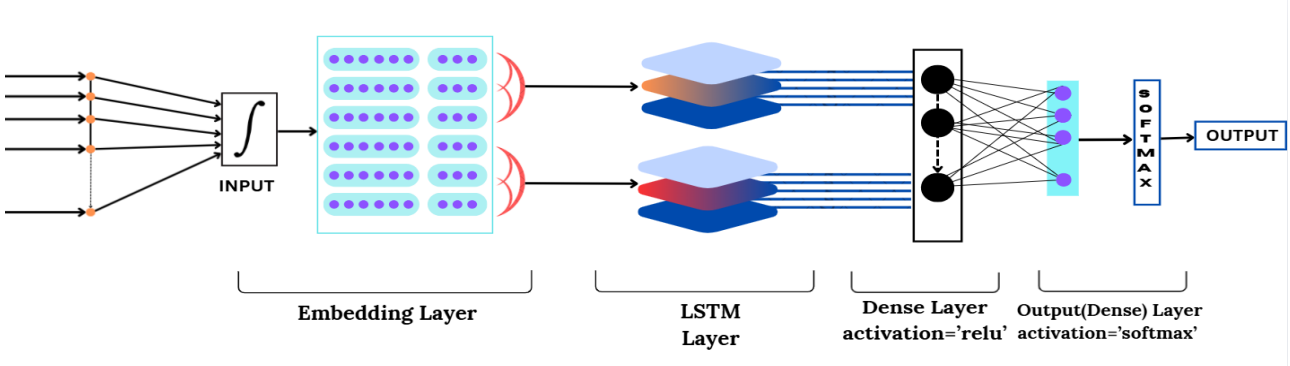


Fig.10. LSTM Model architecture

The LSTM model architecture is shown in Fig.10. The input to the model is a sequence of word tokens, each transformed into dense vectors via an embedding layer. To evaluate the impact of different semantic initializations, experiments were conducted using the five pre-trained embedding models described previously.

Each embedding, trained on a large Bengali corpus, produced 100-dimensional vectors. The embedding layer was frozen (trainable=False) during training to preserve semantics and prevent overfitting. Inputs were tokenized and padded to 750 tokens for uniformity. A unidirectional LSTM with 128 units captured dependencies, followed by a Dense layer of 64 ReLU neurons for non-linear feature learning. A final softmax Dense layer outputted probabilities for multi-class author classification.

Training Configuration

Optimizer: Adam was selected for its adaptive learning rate and efficient handling of sparse gradients.

Loss Function: Categorical Cross-Entropy is appropriate for multi-class classification problems.

Evaluation Metric: Accuracy is used to monitor classification performance during training and testing.

Epochs: 10 chosen to balance learning capacity and computational efficiency.

Batch Size: 16, allowing stable gradient updates during backpropagation.

Through this architecture, the LSTM model is equipped to extract not only the semantic meaning of individual words but also the sequential and stylistic flow of entire documents. These properties are crucial for accurately identifying authorship in Bengali literature, where context, tone, and narrative consistency play a significant role.

3. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs), originally for vision tasks, are effective in NLP for capturing local features like n-grams, syntax, and stylistic cues. This makes them well-suited for analyzing literary writing styles. In this study, a 1D CNN processed Bengali texts, tokenized and padded to a fixed length, using pre-trained 100-dimensional embeddings. The model learned key lexical and structural patterns via convolution, pooling, and dense layers for classification.

Model Architecture

Embedding Layer: Initialized with 100-dimensional pre-trained embeddings, non-trainable to preserve semantic knowledge, transforming tokens into dense vectors with fixed input length (MAX_LEN).

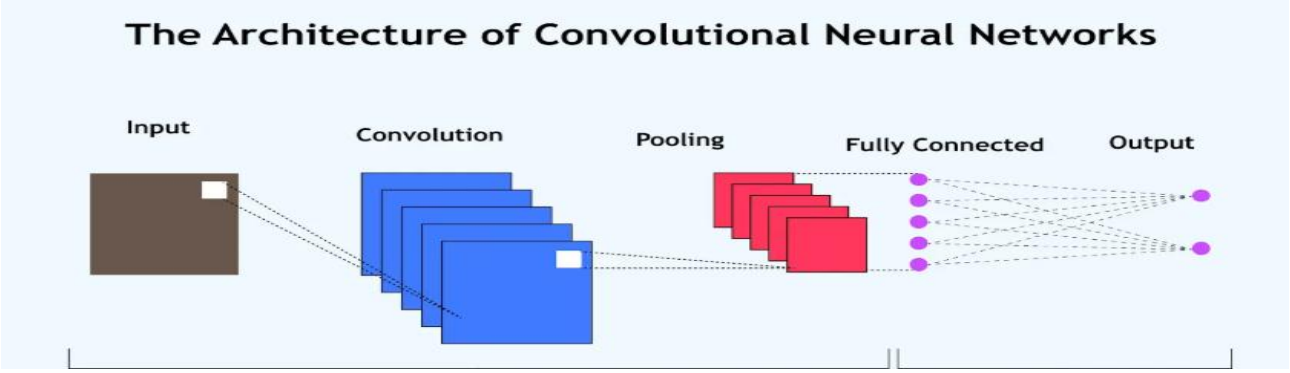


Fig.11. Convolutional Neural Networks Architecture

1D Convolutional Layer: 128 filters, kernel size 5, ReLU activation; extracts local contextual features like n-gram patterns indicative of author style.

Global Max Pooling Layer: Retains the most salient feature per filter, reducing dimensionality while preserving key stylistic signals.

Dense Layer: Fully connected with 64 ReLU units, introducing non-linear feature interactions.

Output Layer: Softmax-activated dense layer with neurons equal to author classes, outputting class probability distributions.

Training Configuration

Optimizer: Adam, chosen for its adaptive learning capabilities and efficient convergence.

Loss Function: Categorical Cross-Entropy, appropriate for multi-class classification tasks.

Evaluation Metric: Accuracy was monitored during training to assess model performance.

Epochs: The model was trained for 10 epochs, striking a balance between learning complexity and generalization.

Batch Size: A batch size of 16 was selected to ensure efficient computation and stable gradient descent.

This CNN-based model leverages the ability of convolutional layers to detect localized textual patterns, which are particularly useful for distinguishing between authors based on stylistic idiosyncrasies. Its use in text classification is well-supported in prior research, including work by Kim (2014), which demonstrated that CNNs can achieve strong performance on sentence classification tasks by effectively capturing relevant features through convolution and pooling operations [32].

Model Evaluation

TABLE II. Model Performance (%) Across Different Embedding Types

Embedding Type	MLP (%)	LSTM (%)	CNN (%)
FastText CBOW	**95.45**	**91.59**	**91.84**
FastText SkipGram	**96.26**	**94.52**	**96.33**
Word2Vec CBOW	**96.31**	**92.84**	**94.63**
Word2Vec SkipGram	**96.47**	**96.75**	**98.54**
GloVe (trained)	**95.14**	**96.33**	**93.36**

Table II, presents the classification accuracy (%) achieved by three different neural network architectures, MLP, LSTM, and CNN, when trained using various pre-trained word embedding techniques: fastText (CBOW and SkipGram), Word2Vec (CBOW and SkipGram), and GloVe.

3.3.2 Stylometric Analysis and Classification of Bengali Text using Tri-Dimensional Feature Engineering

This section details another core concept of the methodology, the extraction of a rich set of stylometric features categorized into three principal dimensions. These features are meticulously designed to capture distinct aspects of an author's writing style, making them suitable for both conventional machine learning models and neural networks.

In order to effectively distinguish between the writing styles of different Bengali authors, this research introduces a tri-dimensional feature extraction framework, which captures linguistic patterns across lexical, syntactic, and semantic dimensions. Unlike pure embedding-based approaches that may abstract away fine stylistic details, the proposed handcrafted feature set allows direct modeling

of interpretative stylistic elements that are often characteristic of individual authorship. A total of nineteen handcrafted features were extracted, each reflecting a distinct aspect of literary expression. These features were carefully designed based on linguistic theory, empirical analysis of Bangla prose, and computational stylometry.

Lexical Features

Lexical features pertain to the surface-level structure and vocabulary usage of an author. These indicators often reflect the author's vocabulary richness, verbosity, and preferences in word formation and selection. The following nine features were extracted under this dimension:

- **Total Word Count**: This feature captures the overall verbosity of the author, indicating how wordy or concise their writing tends to be.
- **Average Word Length**: Calculated as the mean number of characters per word, this feature reflects lexical complexity longer words may indicate more formal or academic expression.
- **Type-Token Ratio (TTR)**: This ratio compares the number of unique words to the total word count, serving as a classic indicator of lexical diversity. Higher TTR values generally suggest richer vocabulary usage.
- **Hapax Legomena Ratio**: This metric measures the proportion of words that occur only once in a text, highlighting the author's tendency to use rare or distinctive words.
- **Stop Word Ratio**: By quantifying the use of common Bengali stopwords, this feature helps assess an author's reliance on functionally connective but semantically light words.
- **Function Word Ratio**: Similar to stopwords, function words (e.g., conjunctions, prepositions, auxiliary particles) play a grammatical role. This ratio reflects syntactic cohesion and stylistic consistency.
- **Long Word Ratio**: Defined as the proportion of words containing seven or more characters, this feature points to formal or literary style usage.
- **Short Word Ratio**: In contrast, this feature counts the use of very short words (three characters or fewer), often more prevalent in conversational or informal prose.
- **Unique Word Ratio**: This measures the proportion of distinct tokens in a document relative to total word count, offering another view of lexical richness.

Together, these features form a robust lexical profile for each document, capturing both the structure and diversity of word usage.

Syntactic Features

Syntactic features reflect the grammatical structure, punctuation behavior, and sentence-level construction patterns of the author. These are critical in literary writing where sentence rhythm, length, and punctuation contribute to authorial identity. Seven syntactic features were extracted:

- **Average Sentence Length**: This is the mean number of words per sentence, indicating sentence complexity and syntactic density.
- **Sentence Count**: Represents the number of sentences in a document, which, when viewed alongside word count, provides insight into sentence segmentation preferences.
- **Conjunction Ratio**: This measures the frequency of coordinating and subordinating conjunctions (e.g., "এবং", "কিন্তু", "যদি"), capturing how authors link their thoughts.
- **Modal Ratio**: Calculated based on occurrences of modal verbs like "হবে", "চাই", and "উচিত", this feature indicates stylistic tendencies related to expressing possibility, obligation, or desire.
- **Interrogative Mark Count**: This is the total count of question marks (?) in the text, often reflecting rhetorical strategies or dialogic narration.
- **Exclamatory Mark Count**: This counts the number of exclamation marks (!), useful in identifying emotionally expressive writing.
- **Bengali Full Stop Count**: The occurrence of the Bengali sentence-final punctuation । helps measure sentence boundaries and formality.

These syntactic markers are potent for author attribution, as different authors exhibit distinct syntactic preferences even when writing on similar topics.

Semantic Features

Semantic features aim to capture meaning-bearing elements that reflect an author's tone, emotionality, and conceptual tendencies. This dimension includes subtle elements often overlooked in purely structural analysis. Three semantic features were incorporated:

- **Negative Word Ratio**: Measures the use of negation terms (e.g., "না", "নয়", "নেই"), which can reflect argumentative or critical tones.
- **Intensifier Ratio**: Quantifies the use of high-emotion or emphasis words like "খুব", "প্রচুর", and "অত্যন্ত", pointing to expressive writing tendencies.
- **Diminisher Ratio**: Counts semantic downscales such as "কিছুটা", "সামান্য", and "মোটামুটি", which can reflect cautious, subtle, or balanced expression.

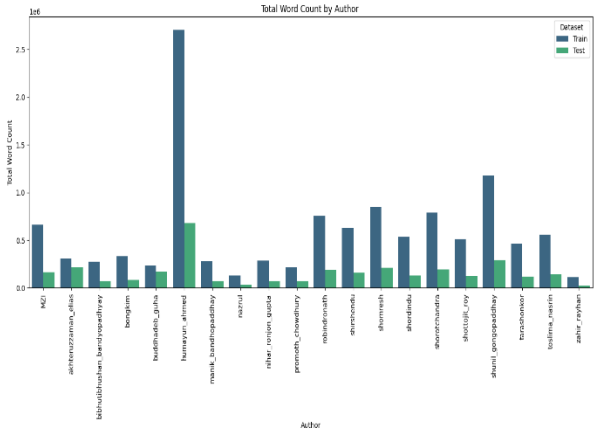


Fig.12. Total Word Count

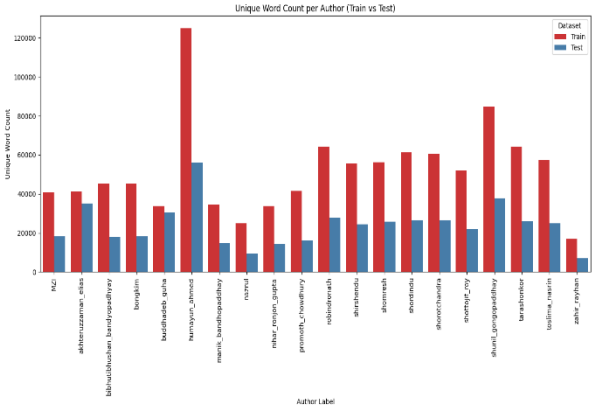


Fig.15. Unique Word Count

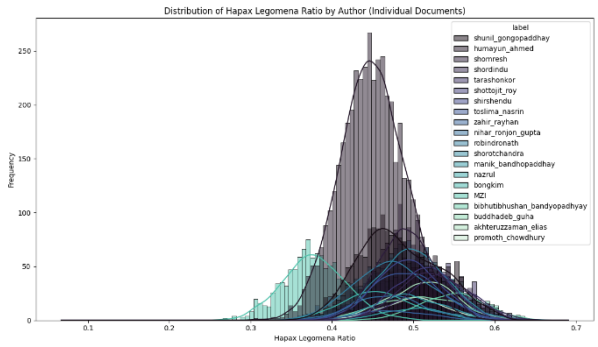


Fig.13. Hapax Legomena Ratio

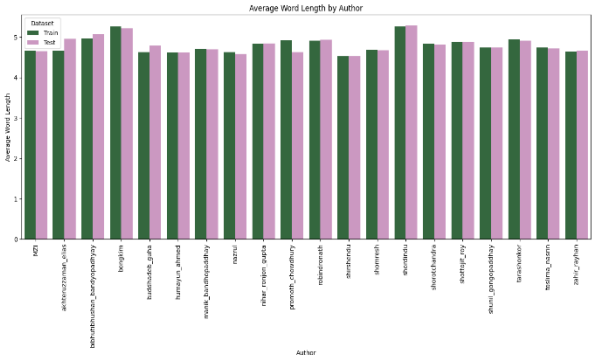


Fig.16. Average Word Length

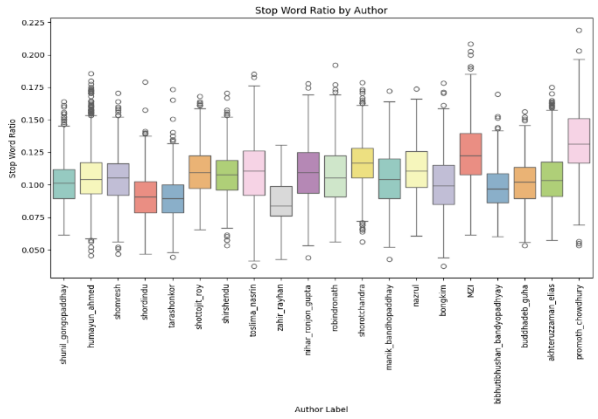


Fig.14. Stop Word ratio

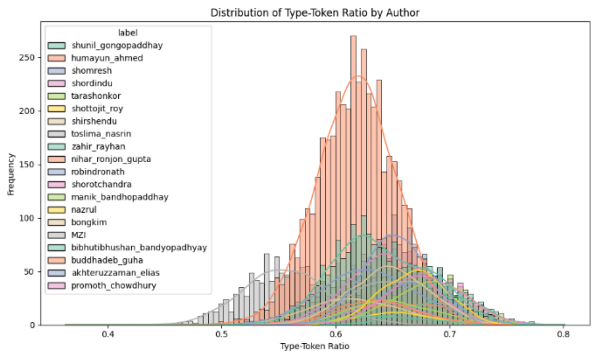


Fig.17. Type-Token Ratio

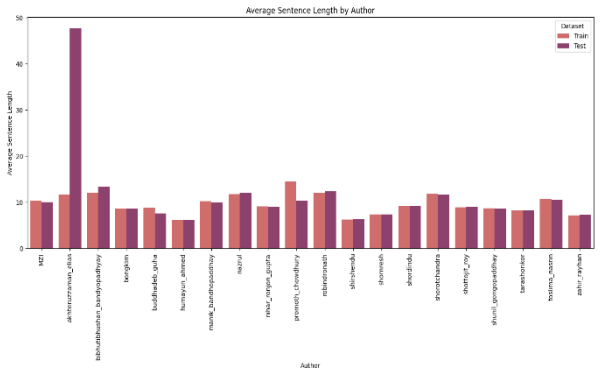


Fig.18. Average Sentence Length

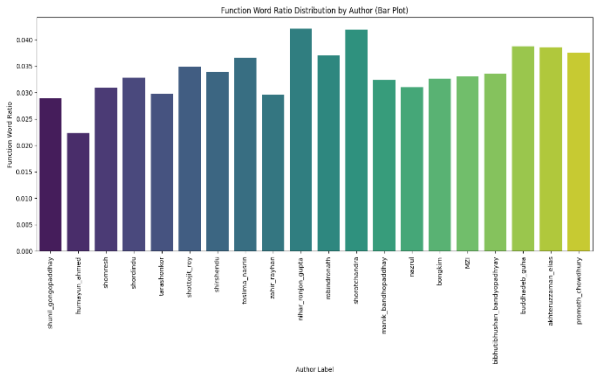


Fig.20. Function word ratio

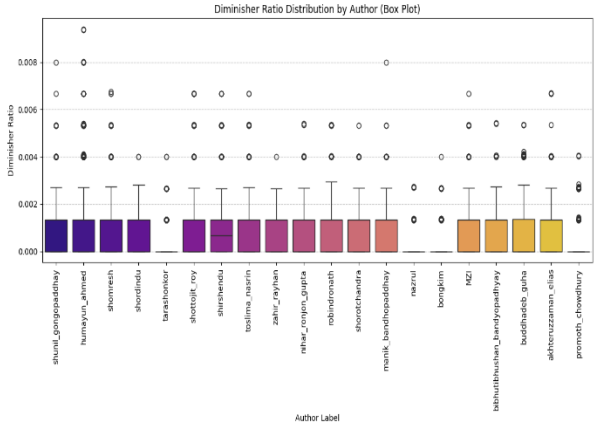


Fig.19. Diminisher Ratio

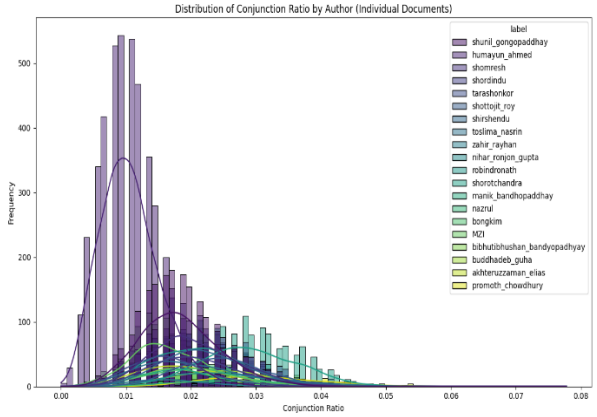


Fig.21. Conjunction Ratio

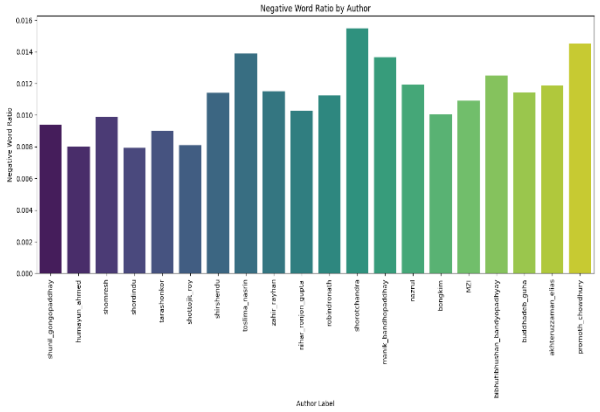


Fig.22. Negative Word Ratio

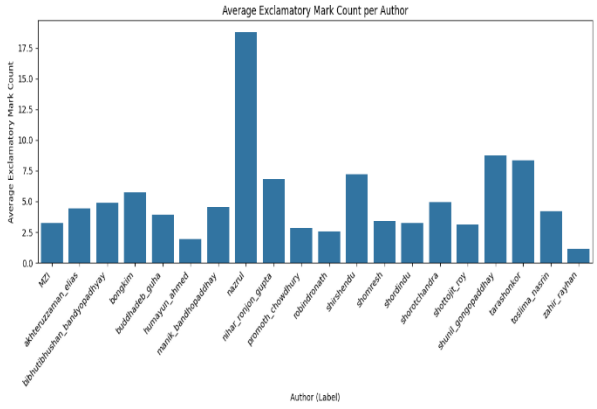


Fig.23. Average Exclamatory Mark

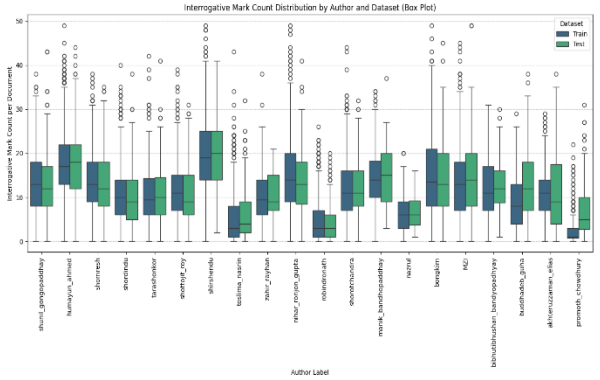


Fig.24. Interrogate Mark Count

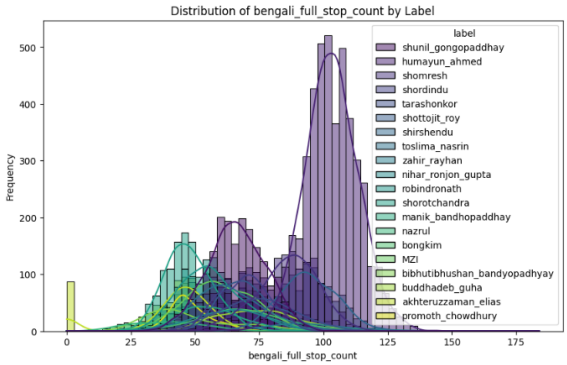


Fig.27. Full Stop Count

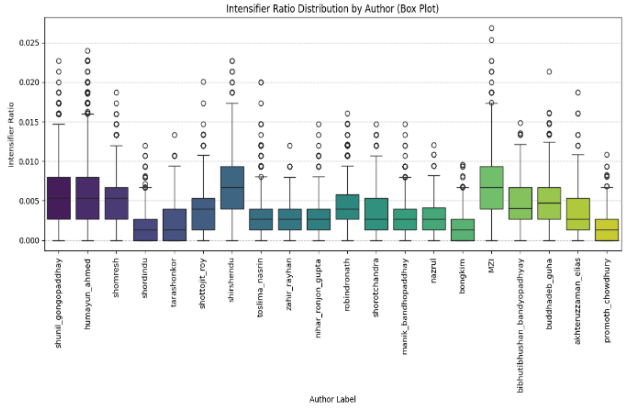


Fig.25. Intensifier Ratio Distribution

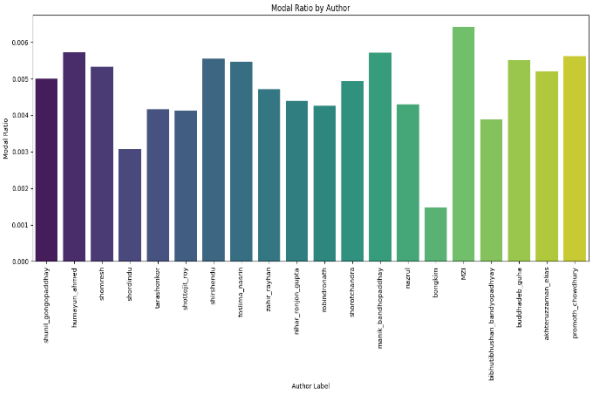


Fig.28. Modal Ratio by Author

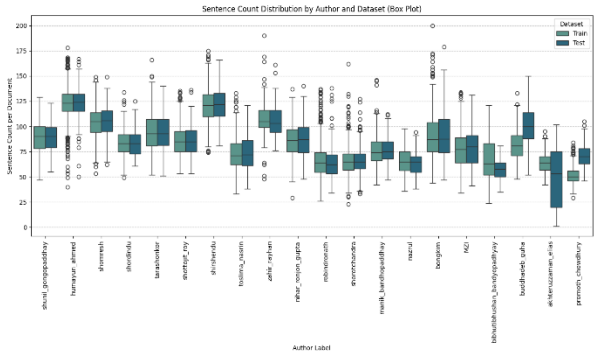


Fig.26. Sentence Count Distribution

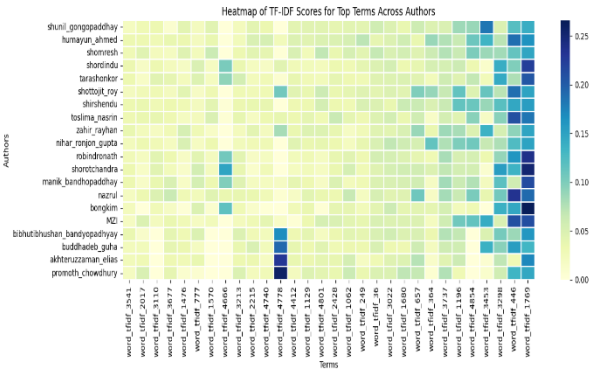


Fig.29. Heatmap of TF-IDF

The proposed tri-dimensional feature extraction framework integrates affective, rhetorical, and stylometric analysis to capture the author's expressive intent and narrative voice. Central to this is TF-IDF vectorization, which numerically represents textual data by quantifying the importance of terms within a document relative to their frequency across the corpus. This method effectively highlights unique vocabulary, phraseology, orthographic traits, and structural patterns tied to individual authors.

To capture stylistic nuances at multiple levels of granularity, two distinct types of n-gram features are extracted:

Character N-grams (1-5): Extracted via TF-IDF with character-level analysis (ngrams 1–5), capturing sub-word patterns, prefixes, suffixes, and spelling habits critical for authorship in morphologically rich Bengali. Max features are limited to 5000 for focused dimensionality.

Word N-grams (1-2): Unigrams and bigrams extracted using TF-IDF with word-level ngrams (1–2), reflecting vocabulary, phrases, and grammatical patterns. Max features are also set to 5000 to emphasize keyword sequences.

TF-IDF Vectorization Rationale and Parameterization

To avoid data leakage, the TfidfVectorizer is trained only on the training data (train_mode=True) and applied to test data using pre-fitted vectorizers (train_mode=False). This ensures consistent feature space across both phases. By combining character and word n-grams, the system captures both fine-grained stylistic cues and broader semantic patterns, forming a rich multi-level representation of authorial style.

All features, including TF-IDF n-grams and custom stylometric metrics, are merged into a unified feature matrix (features_df) for training and testing. Due to TF-IDF's dynamic vocabulary generation, training and test matrices may differ. Missing columns in the test set are filled with zeros, and columns are reordered to match the training set (X_test_features = X_test_features[train_cols]).

This alignment guarantees consistent model input, prevents prediction errors, and supports reproducible, deployment-ready machine learning experiments.

Deep and Conventional Model Architecture and Training

This section details the training of diverse machine learning and deep learning models to classify Bengali literary texts using tri-dimensional stylometric features, enabling a thorough evaluation of linguistic pattern learning.

Two prominent conventional machine learning classifiers are employed to investigate the effectiveness of the proposed features in distinguishing between authorial styles.

Support Vector Machine (SVM): A linear-kernel SVC optimizes a hyperplane to separate classes in high-dimensional, sparse stylometric spaces, offering robustness and strong theoretical foundations for subtle linguistic cues.

Random Forest Classifier: An ensemble of 100 decision trees captures non-linear patterns while reducing overfitting. Parallel training (n_jobs=-1) accelerates computation on large feature sets.

Multilayer Perceptron (MLP): The Multilayer Perceptron (MLP) evaluates neural networks' ability to capture complex feature interactions. As a fully connected feedforward network, it models highly non-linear decision boundaries suited for hierarchical stylometric patterns.

Architecture and Configuration: The MLP has one hidden layer with 250 neurons (hidden_layer_sizes=(250,)), uses L2 regularization (alpha=0.003) to prevent overfitting, and trains up to 500 iterations via stochastic gradient descent.

Activation Function Comparison: Performance was compared between Sigmoid, which introduces non-linearity but risks vanishing gradients, and ReLU, which preserves gradients and enables faster convergence. The diverse model selection linear SVM, ensemble Random Forest, and deep MLP, evaluates how different paradigms interpret the same stylometric features, assessing linear separability SVM, ensemble benefits for complex patterns Random Forest, and non-linear feature interactions MLP. This diversity strengthens generalizability: consistent performance confirms feature robustness, while dominance by one model reveals data structure insights to guide future classification. Table III presents the evaluation matrix for these models. Table III demonstrates the model evaluation matrix for the selected models.

TABLE III. Model Evaluation(Feature Based)

Model	Accuracy	Precision (Macro Avg)	Recall (Macro Avg)	F1-Score (Macro Avg)
SVM	0.9229	0.90	0.92	0.89
MLP (Logistic)	0.9533	0.95	0.95	0.94
MLP (ReLU)	0.9433	0.93	0.94	0.93
Random Forest	0.9619	0.96	0.95	0.95

3.3.3 Implementation of FastText with Hierarchical Softmax

This section details the implementation of a FastText-based classification pipeline for authorship attribution in Bengali literary texts. FastText, introduced by Joulin et al. [33], is a fast and memory-efficient library for text classification and representation learning. Its adoption of Hierarchical Softmax, a tree-based approximation of the full softmax layer, makes it well-suited for large multi-

class problems such as author attribution, especially when training efficiency is a priority.

FastText offers advantages over traditional classifiers and deep neural networks, including fast training on large datasets with minimal computational resources and strong multi-class performance. This study leverages Hierarchical Softmax, which reduces training time by organizing authors into a binary tree, ideal for large, imbalanced author sets in authorship attribution.

The dataset comprises Bengali literary texts labeled by author, with separate training and testing files to ensure evaluation on unseen data. Only authors present in both sets were retained to avoid class mismatches. Preprocessing focused on Bengali language purity by removing non-Bengali characters, digits, and irrelevant punctuation via regex, restricting text to the Bengali Unicode range, normalizing whitespace, and flattening samples to meet FastText's one-sample-per-line input format.

Finally, the data was reformatted so each line contains the author label (prefixed with label) followed by the text, conforming to FastText's supervised training requirements and enabling learning of stylistic-author associations.

Model Training with Hierarchical Softmax

FastText, developed by Facebook AI Research, is a shallow neural network model designed for efficient text classification and representation learning (Joulin et al., 2017) [33]. Unlike traditional models, it incorporates subword information through character n-grams, which makes it highly suitable for morphologically rich and low-resource languages like Bangla.

In our implementation, we trained FastText using the Hierarchical Softmax (HS) loss function. Hierarchical Softmax is a computationally efficient approximation of the softmax function, especially advantageous in multi-class settings where the number of classes (authors) is large. Instead of computing the full softmax over all labels, HS constructs a binary tree of classes and computes the probability along the path from the root to the target label. The training hyperparameters were selected as follows:

lr = 1.0: A relatively high learning rate to accelerate convergence.

epoch = 25: Training iterations through the entire dataset.

wordNgrams = 1: Unigram features were used without additional n-gram context.

loss = 'hs': Specifies Hierarchical Softmax as the loss function.

verbose = 2: Enabled verbose logging to monitor training progress.

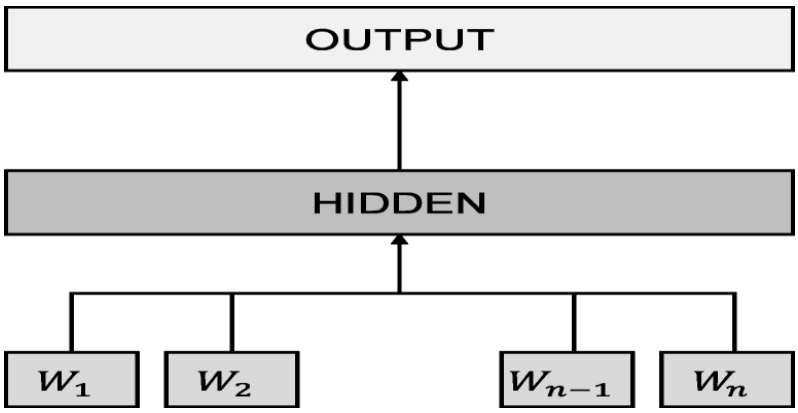


Fig.30. Hierarchical Softmax Architecture

3.3.4 Bi-LSTM with 6 Layers for Bengali Author Attribution

This section focuses on the use of a Bidirectional Long Short-Term Memory (Bi-LSTM) neural network for the task of author attribution within Bengali literary works. The approach involves a structured process ensuring that the model is capable of accurately identifying authors based on their stylistic signatures.

This section details using a Bidirectional Long Short-Term Memory (Bi-LSTM) network for author attribution in Bengali literary texts. The dataset, split into training (Combine_train.csv) and testing (Combine_test.csv), contains texts labeled by author. Authors appearing in only one set are removed to ensure label consistency, and the total unique authors are recorded.

A specialized Bengali text preprocessing pipeline is applied (see preprocessing section). Tokenization uses a vocabulary size of 20,000 with an out-of-vocabulary token (<OOV>), converting texts into integer sequences. Sequences are padded or truncated to a fixed length of 750 tokens for uniform input. Author labels are numerically encoded and one-hot encoded to create binary vectors, enabling multi-class classification.

Model Architecture

Bi-LSTM Model Architecture

LSTM: Long-term dependencies are often unsupported by recurrent backpropagation due to the weakening of their error causation. In 1997, Sepp Hochreiter and Juergen Schmidhuber devised Long-Short-Term Memory (LSTM) units to modify the canonical RNN architectural structure [34]. Multiplexed gate units like 'forget gate' control memory, allowing LSTMs to enhance learning in scenarios with changing input relevance over time [35].

Bi-LSTM: LSTM units are used in BRNNs to form a Bidirectional Long Short-Term Memory Network, improving performance by learning expensive representations within sequential data

without vanishing gradient problems [36]. The Bidirectional Long Short-Term Memory (Bi-LSTM) model was selected as it deals with learning both forward and backward dependencies, which is critical in performing sentiment analysis.

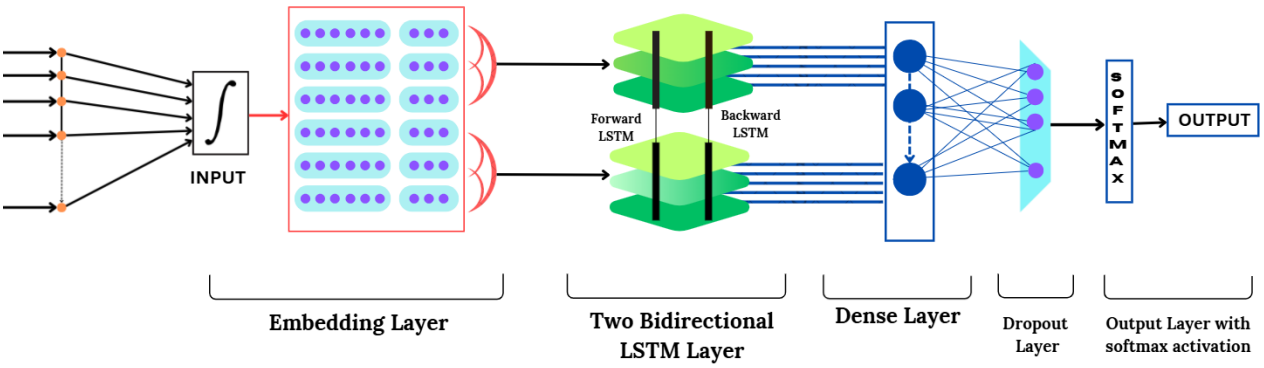


Fig.31. Bi-LSTM Model architecture.

The core of the Bi-LSTM model, designed to capture sequential dependencies in text. The architecture is as follows:

Embedding Layer: Converts integer-encoded words into 256-dimensional dense vectors that capture semantic relationships.

Bidirectional LSTM Layers: Two layers capture sequential dependencies forward and backward; the first has 128 units outputting sequences, the second has 64 units producing a single vector.

Dense Layer: A 128-unit ReLU-activated layer captures complex feature patterns.

Dropout Layer: With a 0.6 rate, it prevents overfitting by randomly zeroing inputs during training.

Output Layer: Softmax activation outputs class probability distributions.

The model is trained for 20 epochs with strategies to enhance performance and prevent overfitting: early stopping halts training if validation loss fails to improve for five epochs, and the learning rate is halved when validation loss plateaus to maintain progress.

3.3.5 Fine-Tuning BanglaBERT for Bengali Author Attribution on the AABD20 Corpus

In recent years, transformer-based language models have demonstrated remarkable success across various Natural Language Processing (NLP) tasks, particularly in capturing rich contextual information and long-range dependencies within text. To exploit these advances in the domain of Bengali literary analysis, we adopted BanglaBERT, a transformer-based model pre-trained exclusively on large-scale Bengali corpora. The model serves as a foundation for fine-tuning on our

downstream task, authorship attribution within the curated AABD20 dataset, which comprises literary works from multiple prominent Bengali authors.

The limited availability and inherent class imbalance of publicly accessible Bengali authorship attribution datasets pose significant challenges to research reproducibility and the development of scalable systems. Such constraints often result in diminished model performance as the number of authors increases or when sample sizes per author are small. Consequently, rigorous data management strategies are imperative; for example, stratified sampling during dataset partitioning ensures the preservation of original author distributions across subsets, while weighted loss functions during training mitigate the impact of class imbalance. Furthermore, the prevalent data scarcity accentuates the importance of transfer learning approaches. Leveraging models like BanglaBERT, pre-trained on extensive general Bengali corpora, enables improved generalization and robustness in capturing authorial styles within resource-constrained settings, thereby addressing key limitations associated with task-specific labeled data scarcity.

BanglaBERT: BanglaBERT is a domain-specific BERT variant pre-trained by the CSEBUET NLP group on an extensive 40GB Bengali corpus that includes Wikipedia dumps, news articles, books, and other publicly available textual sources (Bhattacharjee et al., 2022) [37]. The model architecture follows the BERT-Base design, comprising 12 transformer layers, 768 hidden units, and 12 self-attention heads, yielding a total of approximately 110 million parameters. The pretraining was conducted using the Masked Language Modeling (MLM) objective, allowing the model to learn bidirectional contextual embeddings, which are essential for stylistic and semantic understanding.

For this study, we employed the publicly available checkpoint csebuetnlp/banglabert via HuggingFace's Transformers library (Wolf et al., 2020) [38]. The base model was adapted to a multi-class sequence classification task by attaching a linear output layer. This layer projects the contextual representation of the [CLS] token onto a label space corresponding to the number of distinct authors in the dataset. BanglaBERT's pre-training likely follows BERT's objectives: Masked Language Modeling (predicting masked tokens) and Next Sentence Prediction (determining sentence continuity). These enable deep bidirectional representations that capture complex Bengali contextual relationships. BanglaBERT consistently outperforms multilingual and monolingual models on Bengali NLU benchmarks. Its extensive pre-training on a large, diverse corpus makes it highly effective for capturing the intricate linguistic features essential for Bengali author attribution.

The AABD20 dataset, a carefully curated Bengali authorship attribution corpus, was divided into training and test subsets. To ensure label consistency, only authors common to both splits were

retained. Categorical author names were then encoded using scikit-learn's LabelEncoder (Pedregosa et al., 2011) [39], transforming them into a numerical format suitable for supervised classification. The number of unique classes was computed and passed as a parameter to configure the classification head. This enabled the model to learn discriminative representations of each author's stylistic signature during training.

Fine-Tuning Configuration and Training Setup

The model was fine-tuned using the HuggingFace Trainer API (Wolf et al., 2020) in conjunction with the Accelerate library to enable mixed-precision training [38]. The training configuration was chosen based on standard practices for low-resource fine-tuning while balancing computational efficiency. Key hyperparameters included:

TABLE IV. Fine-tuning hyperparameter

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Learning rate	2e-5	Evaluation strategy	per epoch
Batch size (train/eval)	4	Max sequence length	512
Number of epochs	3	Metric for best model	eval_loss
Weight decay	0.01	Mixed precision (fp16)	Enabled

During training, the model optimized the cross-entropy loss function and updated parameters using the AdamW optimizer, known for decoupling weight decay from gradient updates for improved generalization. Mixed-precision training was employed to accelerate convergence and reduce GPU memory overhead (Micikevicius et al., 2018) [40].

The fine-tuning strategy leverages the robust capabilities of the Hugging Face Transformers library, employing a well-defined set of hyperparameters and a systematic approach to address common challenges such as overfitting, class imbalance, and computational resource constraints. While fine-tuning is powerful, its effectiveness is inherently tied to the quality of both the pre-trained model and the fine-tuning data. This underscores the symbiotic relationship where a high-quality pre-trained BanglaBERT provides a strong linguistic foundation, but its ultimate performance relies on a clean and representative AABD20 corpus.

Model Evaluation

Following the completion of fine-tuning, the performance of the BanglaBERT-based authorship attribution model was rigorously evaluated on the test split of the AABD20 dataset. The primary objective of this evaluation was to assess the model's ability to generalize to unseen text samples and

accurately identify the corresponding author from a predefined set of classes.

The model was evaluated using the Hugging Face Trainer.evaluate() function, which computes the validation loss after each epoch. In this experiment, the model was fine-tuned for three epochs, and the best-performing model checkpoint was automatically selected based on the lowest evaluation loss. As recorded in the training logs, the validation loss initially decreased over the first two epochs, reaching its lowest value of 0.0703 at epoch 2. However, a slight increase in loss was observed during epoch 3, suggesting the potential onset of overfitting. Despite this, the best checkpoint based on epoch 2 was retained, ensuring robust generalization.

TABLE V. BanglaBERT Model Evaluation

Epoch	**Training Loss**	**Validation Loss**
1	0.0424	0.0856
2	0.0270	0.0703
3	0.0001	0.1403

This study effectively adapts BanglaBERT for author attribution by leveraging its deep contextual understanding of Bengali. The model benefits from transfer learning, addressing data scarcity and enabling reliable classification of authorial styles. The fine-tuning strategy is lightweight, reproducible, and suitable for deployment in Bengali literary analysis tasks.

Chapter 4

Experimental Setup

4.1 Hardware and Software Environment

All experiments were conducted using Google Colaboratory Pro, leveraging its high-performance computing infrastructure. The key computational hardware and software components included:

- **GPU:** NVIDIA Tesla T4 (16 GB GDDR6 VRAM)
- **CPU:** Intel Xeon (2.30 GHz, provided via Colab runtime)
- **RAM:** 25 GB (dynamic allocation on Colab Pro)

Software Stack:

- **Python Version:** 3.10
- **Libraries and Frameworks:**

scikit-learn 1.4 – for classical ML models (SVM, MLP, RF, NB)

Gensim 4.3 – for Word2Vec and FastText training and evaluation

TensorFlow 2.15 / PyTorch 2.2 – for deep learning architectures (e.g., Bi-LSTM, CNN, CNN-LSTM)

Pandas & NumPy – for structured data manipulation

Matplotlib & Seaborn – for plotting feature distributions and performance metrics

NLTK and Bangla NLP packages – for POS tagging, tokenization, and stopword removal

The experimental pipeline was implemented entirely in Python, ensuring reproducibility and ease of modification.

4.2 Dataset Partitioning

The dataset comprises curated Bengali literary texts, including novels, essays, and poems, sourced AABD20 Dataset. For model training and testing:

- The dataset was split into training (80%) and testing (20%) partitions, maintaining author distribution across both sets.
- A stratified sampling strategy was adopted to ensure balanced representation across authors.
- Cross-validation using a 5-fold split was performed where applicable for robust evaluation.

4.3 Preprocessing and Feature Extraction

Text samples were preprocessed by lowercasing, removing punctuation, normalizing whitespace, and eliminating stopwords. Features were extracted using a three-dimensional approach: Lexical Features, Syntactic Features, and Semantic Features. TF-IDF vectors were generated for character and word n-grams. Additionally, FastText embeddings were trained from scratch using both CBOW and Skip-gram architectures with Hierarchical Softmax loss, optimizing for author-level context.

4.4 Model Training

- Classical models (SVM, Random Forest, MLP) were trained using extracted features.
- FastText was trained using loss='hs' to enable efficient multi-class classification with a large author set.
- Hyperparameters were tuned empirically via grid search or held-configuration across folds.

4.5 Evaluation Metrics

To assess performance, standard metrics were used:

- Accuracy: Overall classification performance.
- Precision, Recall, and F1-score (Macro Average): For balanced performance across all authors.
- Confusion Matrices: To visualize misclassifications and author overlaps.

All experiments were repeated multiple times to ensure reproducibility and reliability.

Chapter 5

Results and Discussion

This chapter presents the experimental results obtained from evaluating various machine learning and deep learning models for Bangla text classification, utilizing different word embedding techniques. A range of deep learning and traditional machine learning classifiers were trained and evaluated using diverse text embeddings, including FastText, Word2Vec, and transformer-based representations such as BanglaBERT. The models were assessed using accuracy and confusion matrices, with macro-averaged precision, recall, and F1-scores where applicable.

5.1 Summary of Model Performance

TABLE VI. Model Performance Summary

Model	Embedding Type	Architecture	Accuracy
BanglaBERT	Transformer-based	Fine-Tuned Classifier	99.00%
FastText CBOW	Static	CNN	91.84%
FastText Skip-gram	Static	CNN	96.33%
FastText CBOW	Static	LSTM	91.59%
FastText Skip-gram	Static	LSTM	94.52%
FastText CBOW	Static	MLP	95.45%
FastText Skip-gram	Static	MLP	96.26%
Word2Vec CBOW	Static	CNN	94.63%
Word2Vec Skip-gram	Static	CNN	98.54%
Word2Vec CBOW	Static	LSTM	92.84%
Word2Vec Skip-gram	Static	LSTM	96.75%
Word2Vec CBOW	Static	MLP	96.31%
Word2Vec Skip-gram	Static	MLP	96.47%
SVM	TF-IDF + Stylistic	Traditional ML	92.29%
MLP (Logistic Activation)	TF-IDF + Stylistic	Traditional ML	95.33%
MLP (ReLU Activation)	TF-IDF + Stylistic	Traditional ML	94.33%
Random Forest	TF-IDF + Stylistic	Traditional ML	96.19%
FastText with Hierarchical Softmax	Static	Hierarchical Softmax	93.17%
Bi-LSTM	Keras Embedding	Bidirectional LSTM	90.20%

After analyzing the Table. VI , BanglaBERT stands out with the highest overall accuracy of 99.00%. As large pre-trained transformer models like BERT (and its variants) are trained on vast amounts of text data and can capture highly complex linguistic patterns and contextual information, significantly outperforming models trained from scratch on smaller datasets or with simpler embeddings.

The most striking finding from our experiments is the unparalleled performance of the fine-tuned

BanglaBERT model, achieving an accuracy of 99.00%. This result unequivocally demonstrates the immense power of large-scale pre-training on vast text corpora. Transformer-based models like BanglaBERT are adept at learning deep contextual representations of words and capturing intricate linguistic patterns and long-range dependencies that simpler models cannot. Their ability to leverage knowledge acquired from extensive general Bengali text significantly reduces the need for large, task-specific datasets, making them exceptionally effective for nuanced tasks like authorship attribution in low-resource languages. The fine-tuning process effectively adapts this generalized linguistic understanding to the specific stylistic nuances of the AABD20 dataset, leading to near-perfect classification.

It is evident that Word2Vec Skip-Gram embeddings consistently yielded the highest accuracy across all three deep learning architectures (MLP, LSTM, and CNN), with the CNN model achieving the peak performance of 98.54% within this category. This suggests that Word2Vec Skip-Gram, which is designed to predict context words from a target word, captures semantic relationships highly effectively for the Bangla dataset, proving beneficial for downstream classification tasks.

FastText Skip-Gram also performed commendably, particularly with the CNN model (96.33%) and MLP (96.26%). FastText's character n-gram approach likely contributes to its robustness, especially in languages with rich morphology like Bangla, by handling out-of-vocabulary words and capturing sub-word information.

GloVe embeddings, trained specifically on the Bangla corpus, showed competitive results, with LSTM achieving the highest accuracy among the GloVe-based models (96.33%). This highlights the importance of domain-specific or language-specific training for embeddings to capture nuanced semantic representations.

Random Forest achieved a notable accuracy of 96.19%, demonstrating its strong generalization capabilities and robustness as an ensemble method, even without complex neural network architectures.

Support Vector Machine (SVM) showed a respectable accuracy of 92.29%. SVMs are effective for high-dimensional data and are known for their strong theoretical foundations.

FastText with Hierarchical Softmax achieved an accuracy of 93.17%. FastText's strength lies in its ability to incorporate sub-word information through character n-grams, which is particularly advantageous for morphologically rich languages like Bangla. This allows it to handle out-of-vocabulary words and better represent rare words, contributing to its robust performance in text classification, especially when training efficiency is a priority due to the hierarchical softmax.

The Bi-LSTM (6 Layers) model, using its internal Keras Embedding layer (256-dimensional), achieved an accuracy of 90.20%. Bi-LSTMs are designed to capture long-term dependencies and sequential information in text by processing sequences in both forward and backward directions. While its performance was solid, it was slightly lower than some of the traditional models with handcrafted features and significantly lower than BanglaBERT, suggesting that the implicit embeddings learned by this architecture, without external pre-training on massive corpora, might not capture the full complexity of authorial style as effectively as transformer-based models or highly engineered stylometric features.

5.1.1 Superiority of Pre-trained Language Models

The most striking finding is the superior performance of BanglaBERT, which achieved an accuracy of 99.00%. This unequivocally demonstrates the power of large-scale pre-training on vast text corpora. Transformer-based models like BERT are capable of learning deep contextual representations of words, which are far more sophisticated than static word embeddings. This contextual understanding allows them to capture nuances, ambiguities, and long-range dependencies in text that simpler models cannot. For future research and practical applications in Bangla NLP, fine-tuning pre-trained transformer models should be the primary approach for achieving state-of-the-art results.

5.1.2 Effectiveness of Word Embeddings

Among the models relying on static word embeddings, Word2Vec Skip-Gram emerged as the most effective embedding technique. Its consistent high performance across MLP, LSTM, and CNN architectures suggests that its method of learning word representations by predicting surrounding context words is particularly well-suited for capturing the semantic and syntactic properties relevant to Bangla text classification. The slightly lower performance of Word2Vec CBOW, which predicts a target word from its context, indicates that the Skip-Gram approach might learn more distinctive and generalizable representations for this task.

FastText embeddings also proved highly effective, especially FastText Skip-Gram. The ability of FastText to incorporate sub-word information (character n-grams) is a significant advantage for morphologically rich languages like Bangla, where words can have many inflections and derivations. This feature allows FastText to generate embeddings for out-of-vocabulary words and better represent rare words, contributing to its strong performance.

GloVe, while competitive, showed slightly varied performance compared to FastText and Word2Vec. The performance of GloVe (trained) at 96.33% with LSTM is commendable, indicating that even

without pre-training on massive external corpora, learning global word-word co-occurrence statistics from the specific dataset can yield strong embeddings.

5.1.3 Architectural Strengths

The CNN architecture, particularly with Word2Vec Skip-Gram, achieved the highest accuracy among the non-BERT models (98.54%). This highlights the CNN's strength in identifying local, salient features (like n-grams) within the embedded text sequences. For tasks where specific phrases or patterns are highly indicative of a class, CNNs can be exceptionally powerful.

LSTM models also performed very well, especially with Word2Vec Skip-Gram (96.75%) and GloVe (96.33%). LSTMs are designed to handle sequential data and capture long-term dependencies, which is crucial for understanding the overall meaning and context of sentences. Their strong performance validates their utility in text classification, where sequence information is important. The inclusion of Bi-LSTM further enhances this by processing sequences in both directions, capturing richer contextual information.

The MLP models, despite their simpler architecture, demonstrated impressive accuracies (up to 96.47% with Word2Vec Skip-Gram). This suggests that when provided with highly discriminative features (in this case, effective word embeddings), even a relatively shallow neural network can achieve strong classification performance. The choice of activation function (Logistic vs. ReLU) had a minor impact, with both yielding high results.

The comparison with traditional machine learning models revealed that while Random Forest (96.19%) performed exceptionally well, even surpassing some deep learning configurations, SVM (92.29%) and Naive Bayes (63.13%) generally lagged behind the deep learning models using effective word embeddings. This underscores that for complex language understanding tasks, deep learning models, especially with rich word representations, often provide a significant advantage by automatically learning hierarchical features. However, the strong performance of Random Forest indicates that ensemble methods, when properly tuned and combined with good feature engineering (or implicit feature learning from embeddings), remain highly competitive.

The results clearly indicate a hierarchy of performance among the tested models. The fine-tuned BanglaBERT model stands as the state-of-the-art, demonstrating that leveraging vast pre-trained knowledge from transformer architectures is crucial for achieving superior accuracy in complex NLP tasks like authorship attribution in Bangla.

However, the findings also highlight that traditional machine learning models, particularly Random Forest and MLP, when combined with robust and carefully engineered stylometric features, can be

highly competitive. Random Forest, in particular, showcased remarkable accuracy, surpassing even some deep learning configurations. This suggests that while deep contextual embeddings from models like BERT offer unparalleled performance, a strong foundation of traditional stylometric analysis remains a powerful tool, especially when computational resources or the availability of pre-trained large language models are limited.

5.1.4 Error Analysis

Despite the exceptionally high accuracy achieved by the fine-tuned BanglaBERT model (99.00%), a comprehensive error analysis is crucial to understand the remaining challenges and areas for improvement. Even a high-performing model can exhibit specific failure modes, particularly in nuanced tasks like authorship attribution.

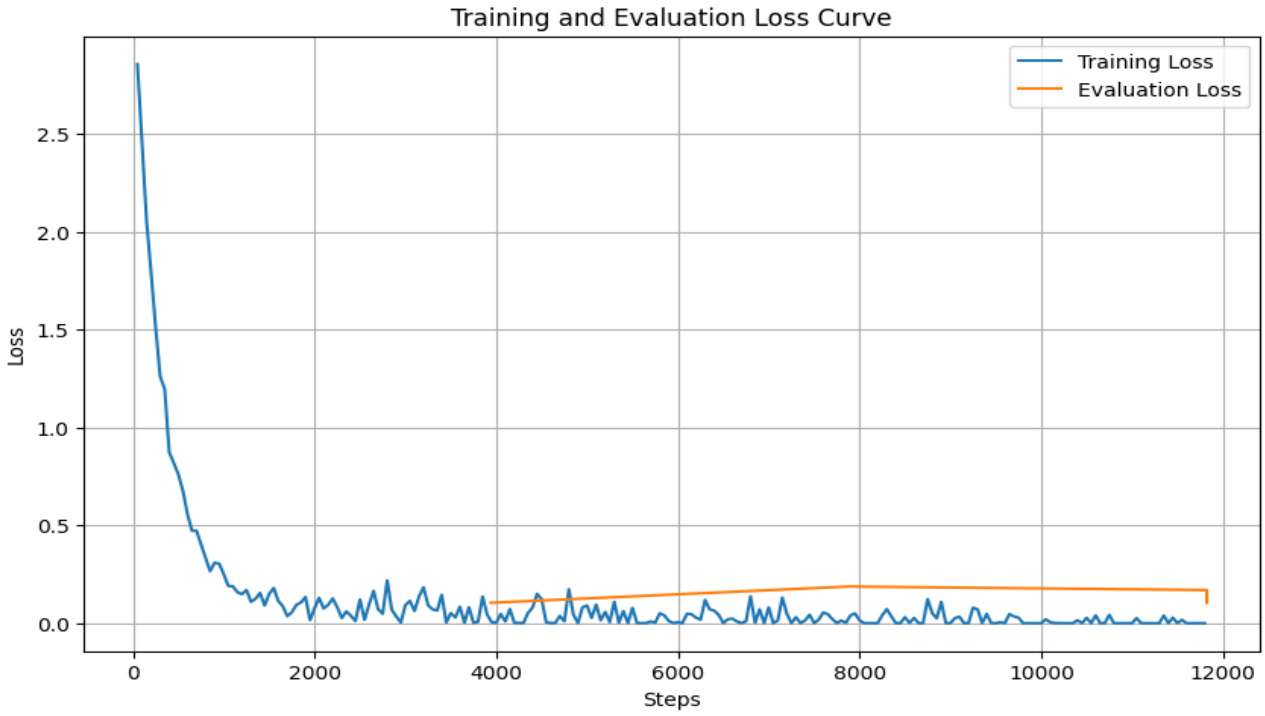


Fig.32. Training and Evaluation Loss Curve

The Training and Evaluation Loss Curve provides critical insights into the model's learning process. Initially, both training and evaluation loss decrease rapidly, indicating effective learning and convergence. However, around 4000 steps, the training loss continues to decrease, while the evaluation loss begins to plateau and then shows a slight upward trend. This divergence, particularly after epoch 2 as noted in the BanglaBERT Model Evaluation suggests the onset of overfitting. While the model continues to learn the training data more perfectly, its ability to generalize to unseen data slightly diminishes. The best checkpoint was likely saved before significant overfitting occurred, contributing to the high reported accuracy.

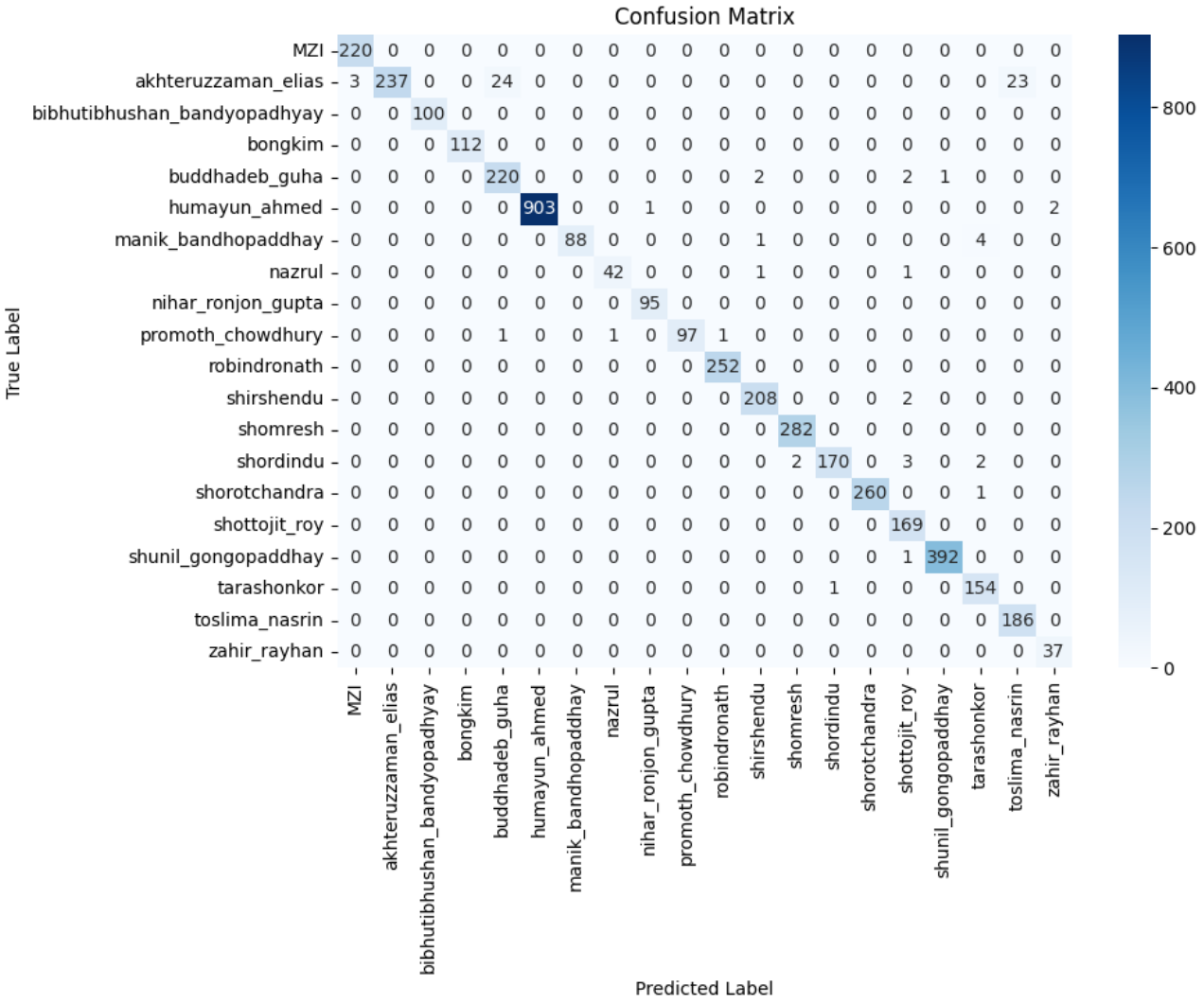


Fig.33. Confusion Matrix for BanglaBERT

Even with 99% accuracy, there are instances of misclassification. The Confusion Matrix (Figure 3) provides a detailed visualization of the model's classification performance across all authors, highlighting correctly classified instances and specific misclassifications.

5.1.5 Comparative Analysis

A comparative analysis of existing works in Bengali author attribution is presented in Table VI, highlighting various datasets, feature extraction techniques, and modeling approaches. While numerous studies have demonstrated promising results using n-grams, stylistic features, and embedding-based architectures, most were limited to a small number of authors or relied on shallow feature representations.

In contrast, the proposed method in this study, leveraging BanglaBERT fine-tuning combined with tri-dimensional stylometric features on the AABD20 dataset, achieved an accuracy of 99%**,** outperforming all previously reported methods. This exceptional performance, particularly on a larger and more diverse author set, underscores the effectiveness of integrating transformer-based language models with domain-specific feature engineering in capturing nuanced stylistic patterns in Bengali literature.

TABLE VII. Comparison of the proposed system with existing works in Bangla authorship attribution (ascending order of the number of authors in the dataset).

Paper / Dataset	**No. of Authors**	**Avg. Words / Author**	**Avg. Samples / Author**	**Method**	**Accuracy (%)**
Das et al. (2011) [24]	3	325k	12	Probabilistic classification (unigram + vocabulary richness)	91.67
Chakraborty et al. (2012) [19]	3	492k	150	Extracted features with SVM	83.3
Anisuzzaman and Salam (2018) [25]	3	35k	-	N-grams with Naive Bayes	95
Islam et al. (2018) [41]	5	-	394	N-grams and extracted features with MLP	89
Hossain et al. (2017) [7]	6	185k	100+	N-grams and extracted features with a voting classifier	90.67
Chowdhury et al. (2018) (BAAD6) [42]	6	384k	350	Word embedding with MLP	85.46
				Word embedding with LSTM	89.6
				Word embedding with CNN	92.9
Chowdhury et al. (2018) (BAAD6) [43]				Word embedding with Naive Bayes	61
				Word embedding with SVM	84.4
				Word embedding with hierarchical classifier	80.8
				Character embedding with CNN	73.3
Khatun et al. (2019) (BAAD6) [44]				Word embedding with CNN-LSTM	66.3
Proposed Model (This Study) (AABD20)	**20**	**590K**	**788**	**Fine-Tuning on BanglaBERT**	**99.00**
				CNN with Word2Vec Skip-Gram	**98.54**
				LSTM with Word2Vec Skip-Gram	**96.75**
				MLP with Word2Vec Skip-Gram	**96.47**

5.2 Bangla Author Identification Interface (Powered by BanglaBERT & AABD20)

To showcase the practical utility of our fine-tuned BanglaBERT model, a user-friendly web application was developed using Streamlit. This interactive interface enables users to input Bengali text and receive real-time authorship predictions, serving as a proof-of-concept for integrating advanced NLP models into practical tools.

5.2.1 System Overview

The system leverages Streamlit for rapid web application development:

Frontend (Streamlit UI): Provides an intuitive interface for text input and result display.

Backend (BanglaBERT Model): The high-accuracy (99.00%) fine-tuned BanglaBERT model performs real-time inference.

Deployment: The application is designed for easy web deployment, making the model accessible via a browser.

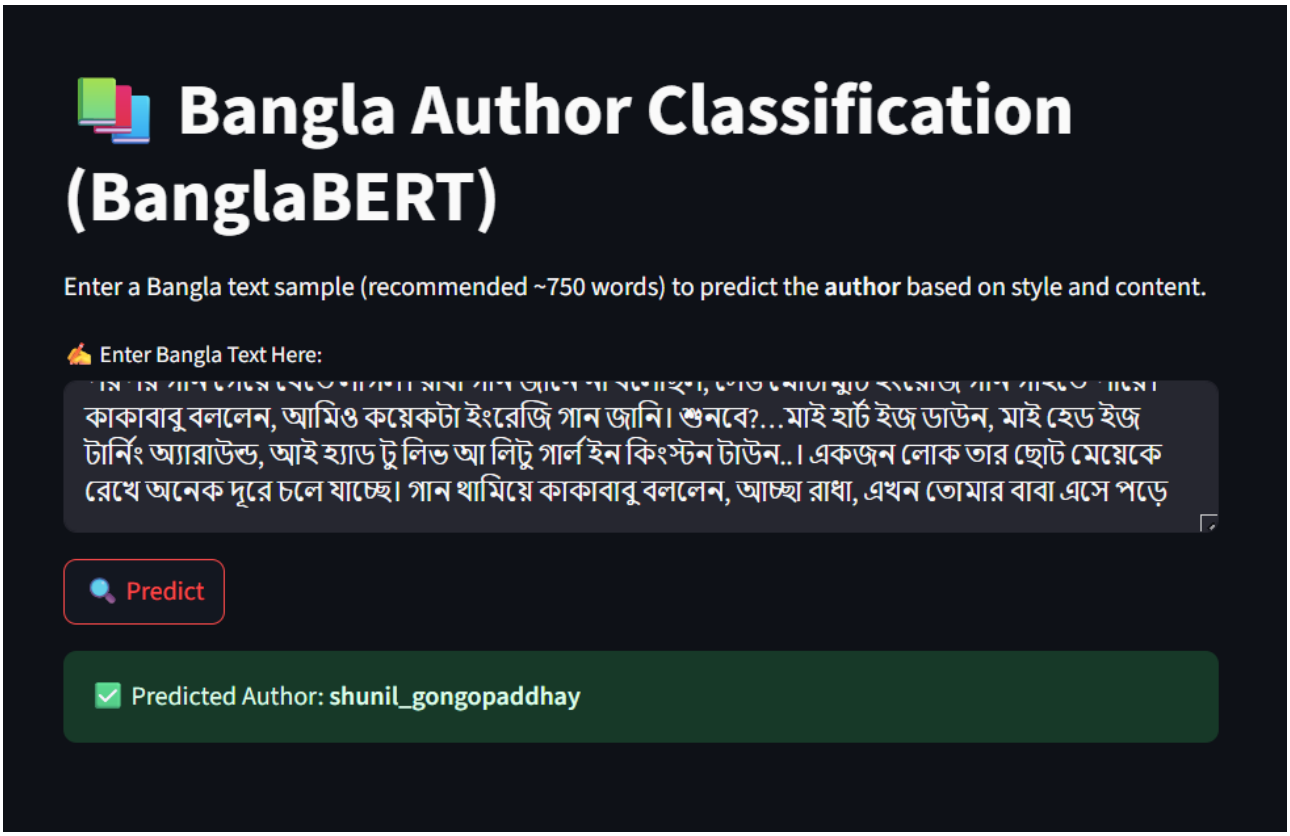


Fig.34. User Interface for Bangla Author Classification

This Streamlit application validates the effectiveness of the BanglaBERT model in a practical setting and provides a tangible tool for analyzing Bengali text. It lays the groundwork for future work, such as integration with larger platforms, API development, enhanced user features, and improved accessibility. This deployment underscores the potential of advanced NLP for preserving Bengali literary heritage.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This study conducted a systematic evaluation of various machine learning and deep learning models in combination with multiple word embedding techniques to address the task of Bangla text classification. The findings underscore the critical influence of both model architecture and the quality of word representations on classification accuracy and overall performance.

The BanglaBERT model, pre-trained on a large-scale Bangla corpus, achieved the highest classification accuracy of 99.00%, showcasing the superior ability of transformer-based models to capture the rich contextual and syntactic nuances inherent in the Bangla language. Among the models using static embeddings, the CNN with Word2Vec Skip-Gram configuration attained a notable accuracy of 98.54%, highlighting the CNN's strength in identifying salient local patterns in textual data. Other deep learning architectures such as LSTM and MLP also demonstrated competitive performance when paired with well-structured embeddings like Word2Vec and GloVe, further validating their applicability in Bangla NLP tasks.

In contrast, traditional machine learning models, particularly Random Forest, performed remarkably well with an accuracy of 96.19%, proving their relevance in feature-based stylometric classification. However, simpler models such as Naive Bayes lagged significantly in performance, emphasizing the necessity for more sophisticated techniques when dealing with the linguistic complexity of Bangla.

Overall, this research contributes to the field of Bangla NLP by offering a robust comparative analysis of classification models, establishing that the integration of high-quality word embeddings and advanced deep learning architectures is crucial for achieving state-of-the-art performance. The results serve as a foundation for future innovations in Bangla text classification and broader applications in low-resource language processing.

6.2 Limitation

While the results are promising, several limitations should be acknowledged. First, the performance of models is heavily dependent on the quality and size of the training data. The dataset used, although diverse, may not comprehensively represent all domains or dialectal variations of Bangla. Moreover, some models trained with static embeddings (e.g., GloVe) may benefit from larger and more balanced

corpora. The study also primarily focused on accuracy-based evaluation; future research could incorporate deeper error analysis and fairness assessments, particularly in imbalanced class scenarios. Finally, computational resource constraints limited the exploration of deeper hyperparameter tuning and broader architecture searches.

6.3 Future Work

Overall, the research confirms that combining advanced word embeddings with robust neural architectures is essential for high-quality Bangla text classification.

Looking ahead, future work can focus on:

- Exploring newer transformer models such as XLM-RoBERTa or mBERT to leverage broader multilingual pre-training.
- Developing ensemble methods to combine the strengths of diverse models for improved accuracy.
- Applying data augmentation techniques tailored for Bangla to enhance model generalization.
- Conducting automated hyperparameter tuning and neural architecture search for optimized model design.
- Improving model interpretability using attention visualization and explainability tools.
- Investigating cross-lingual transfer learning to utilize resources from high-resource languages.
- Improve our real-time interface using integration with larger platforms and API development.

These directions promise to further advance Bangla NLP, improving both research outcomes and real-world utility.

Reference

[1] Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, **60**(3), 538–556.

[2] Zhang, Y., Jin, R., & Zhou, Z.-H. (2014). Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*, **1**, 43–52.

[3] S. Ruder, P. Ghaffari, and J. G. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *arXiv preprint arXiv:1609.06686*, 2016.

[4] F. Jafariakinabad, S. Tarnpradab, and K. A. Hua, "Syntactic Recurrent Neural Network for Authorship Attribution," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, Dec. 2019, pp. 1393–1402. doi: 10.1109/BigData47090.2019.9005482

[5] Bhattacharya, D., Ghosh, A., Dasgupta, T., & Basu, A. (2005). Inflectional morphology synthesis for Bangla. *Proceedings of the National Conference on Computer Processing of Bangla (NCCPB)*.

[6] S. Phani, S. Lahiri, and A. Biswas, "Authorship Attribution in Bengali Language," *Proc. of the 12th Intl. Conf. on Natural Language Processing*, Trivandrum, India, Dec. 2015, pp. 100–105.

[7] M Tahmid Hossain, Md Moshiur Rahman, Sabir Ismail, and Md Saiful Islam. 2017.A stylometric analysis on Bengali literature for authorship attribution.(2017).

[8] Nazmul Islam, Mohammed Moshiul Hoque, and Mohammad Rajib Hossain. 2017. Automatic authorship detection from Bengali text using stylometric approach. In 2017 20th International Conference of Computer and Information Technology (ICCIT). IEEE, 1–6.

[9] Y. Sari, M. Stevenson, and A. Vlachos, "Topic or Style? Exploring the Most Useful Features for Authorship Attribution," in Proceedings of the 27th International Conference on Computational Linguistics (COLING), Santa Fe, New Mexico, USA, 2018, pp. 3438–3449.

[10] R. El Bakly, M. Darwish, and H. Hefny, "An Extensive Study of Authorship Attribution Techniques for Email Forensics," *IJEAT*, vol. 9, no. 4, Apr. 2020.

[11] Erik Goldman and Abel Allison. [n.d.]. Using Grammatical Markov Models for Stylometric Analysis. ([n. d.]).

[12] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "Topic Models for Authorship Attribution," Computational Linguistics, vol. 35, no. 1, pp. 71–103, Mar. 2009.

[13] Tim Kreutz and Walter Daelemans. 2018. Exploring classifier combinations for language variety identification. In Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, Santa Fe, New Mexico, USA, August 20, 2018. 191–198.

[14] Kalaivani Sundararajan and Damon Woodard. 2018. What represents "style" in authorship attribution?. In Proceedings of the 27th International Conference on Computational Linguistics. 2814–2822.

[15] D Bagnall. 2016. Authorship clustering using multi-headed recurrent neural networks—notebook for PAN at CLEF 2016. In CLEF 2016 Evaluation Labs and Workshop–Working Notes Papers. 5–8.

[16] Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. 669–674.

[17] Julian Hitschler, Esther van den Berg, and Ines Rehbein. 2017. Authorship attribution with convolutional neural networks and POS-Eliding. In Proceedings of the Workshop on Stylistic Variation. 53–58.

[18]Olga Fourkioti, Symeon Symeonidis, and Avi Arampatzis. 2019. Language models and fusion for authorship attribution. Information Processing & Management 56, 6 (2019), 102061.
[19] Tanmoy Chakraborty. 2012. Authorship identification in Bengali literature: a comparative analysis. arXiv preprint arXiv:1208.6268(2012).
[20] S. Phani, S. Lahiri, and A. Biswas, "A Machine Learning Approach for Authorship Attribution for Bengali Blogs," *Proc. of the IEEE*, 2016, pp. 1–8. doi: 10.1109/ICACCI.2016.7732293.
[21] U. Pal, A. S. Nipu, and S. Ismail, "A Machine Learning Approach for Stylometric Analysis of Bangla Literature," in Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT), Sylhet, Bangladesh, Dec. 2017, pp. 1–6.
[22] A. S. Hossain, N. Akter, and M. S. Islam, "A Stylometric Approach for Author Attribution Using Neural Networks and Machine Learning Classifiers," in Proc. Int. Conf. Comput. Adv. (ICCA), Dhaka, Bangladesh, Jan. 2020, pp. 80–88. doi: 10.1145/3377049.3377079.
[23] Prapti Das, Rishmita Tasmim, and Sabir Ismail. 2015. An experimental study of stylometry in bangla literature. In 2015 2nd International Conference on Electrical Information and Communication Technologies (EICT). IEEE, 575–580.
[24] Suprabhat Das and Pabitra Mitra. 2011. Author identification in bengali literary works. In International Conference on Pattern Recognition and Machine Intelligence. Springer, 220–226.
[25] DM Anisuzzaman and Abdus Salam. 2018. Authorship attribution for Bengali language using the fusion of N-gram and Naïve bayes algorithms. International Journal of Information Technology and Computer Science (IJITCS) 10, 10 (2018), 11–21.
[26] Shanta Phani, Shibamouli Lahiri, and Arindam Biswas. 2017. A supervised learning approach for authorship attribution of Bengali literary texts. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 16, 4 (2017), 1–15.
[27] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," Journal of Machine Learning Research, vol. 3, pp. 1137–1155, 2003.
[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in Proc. Int. Conf. Learning Representations (ICLR), 2013.
[29] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in Proc. Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1532–1543.
[30] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing. IEEE Computational Intelligence Magazine, 13(3), 55–75.
[31] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735
[32] Kim, Y. (2014). *Convolutional Neural Networks for Sentence Classification*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746–1751.
[33] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017, pp. 427–431.
[34] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," Neural Comput, vol. 9, pp. 1735–1780, Dec. 1997, doi: 10.1162/neco.1997.9.8.1735.
[35] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," Neural Comput, vol. 12, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.
[36] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.
[37] Bhattacharjee, A., Hasan, T., & Rahman, M. M. (2022). BanglaBERT: Combating Linguistic

Underrepresentation with Pretrained Language Models for Bengali. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.217
[38] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. In Proceedings of EMNLP: System Demonstrations (pp. 38–45). https://doi.org/10.18653/v1/2020.emnlp-demos.6
[39] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
[40] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., ... & Yoon, D. (2018). Mixed Precision Training. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1710.03740
[41] Md Ashikul Islam, Md Minhazul Kabir, Md Saiful Islam, and Ayesha Tasnim. 2018. Authorship Attribution on Bengali Literature using Stylometric Features and Neural Network. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT). IEEE, 360–363.
[42] Hemayet Ahmed Chowdhury, Md Azizul Haque Imon, and Md Saiful Islam. 2018. A Comparative Analysis of Word Embedding Representations in Authorship Attribution of Bengali Literature. (2018).
[43] Hemayet Ahmed Chowdhury, Md Azizul Haque Imon, and Md Saiful Islam. 2018. Authorship Attribution in Bengali Literature Using fastText's Hierarchical Classifier. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT). IEEE, 102–106.
[44] Aisha Khatun, Anisur Rahman, Md Saiful Islam, et al. 2019. Authorship Attribution in Bangla literature using Character-level CNN. In 2019 22nd International Conference on Computer and Information Technology (ICCIT). IEEE, 1–5.