Benchmarking Classical Machine Learning and Transformer Architectures for Bangla–English News Classification: A Large-Scale Unified Evaluation with Morphological Complexity Analysis

***Abstract*—Automated news categorization is a core enabling technology for content recommendation, media monitoring, misinformation triage, and digital archiving, yet reliable guidance on model selection is scarce for low-resource, morphologically rich languages such as Bangla. This paper presents a large-scale comparative study of classical machine learning and deep learning approaches for news text classification in Bangla and English under a single, unified preprocessing and evaluation pipeline. Five classical models (Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, and XGBoost), three feature extraction strategies (Bag-of-Words, One-Hot Encoding, and TF-IDF), and three deep architectures (BiLSTM, BERT, and a hybrid LSTM+BERT) are benchmarked on more than half a million native news articles drawn from a 15-category Bangla corpus and the 4-category AG News benchmark. In addition, a quantitative morphological complexity metric is introduced, showing that the overwhelming majority of Bangla articles exhibit medium-to-high inflectional density and providing an empirical explanation for the widening advantage of contextual deep models on morphologically rich text. The results demonstrate that fine-tuned monolingual BERT delivers the strongest performance in both languages (92.95% accuracy for Bangla and 94.50% for English), that the hybrid LSTM+BERT yields only marginal additional benefit, and that SVM with TF-IDF remains the most competitive and computationally efficient classical baseline. These findings translate directly into practical, evidence-based deployment guidelines: transformer fine-tuning where accuracy is paramount, and linear TF-IDF models where GPU resources are unavailable—recommendations relevant to newsrooms, content platforms, and public-sector information systems operating in multilingual and low-resource settings.**

***Index Terms*—Bangla natural language processing, deep learning, low-resource languages, morphological complexity, text classification, transformers.**

I. Introduction

The rapid proliferation of digital content across many languages has created strong demand for automated text classification systems capable of processing and categorizing information at scale [1], [2]. Text classification—the assignment of predefined labels to documents—underpins news categorization, sentiment analysis, spam and toxic-content filtering, fake-news detection, and personalized recommendation [1]–[4]. While English benefits from large annotated datasets and mature NLP pipelines, the landscape for low-resource languages remains comparatively underdeveloped [2], [5]. Bangla (Bengali), spoken by over 230 million native speakers and commonly ranked among the top six most spoken languages in the world [5], [6], is still treated as a low-resource language: its resources and benchmarks remain sparse and fragmented across tasks, domains, and annotation schemes [5], [7].

Bangla poses distinctive challenges for text classification. It exhibits complex inflectional morphology, rich derivational processes, and productive compounding, alongside a continued scarcity of large, high-quality labeled corpora [5], [8], [9]. These characteristics complicate tokenization, feature extraction, and robust generalization, and they often exacerbate class imbalance and domain shift in practice [8]–[10]. Recent Bangla studies—spanning news categorization [9], [10], sentiment analysis [11], [12], hate-speech identification [13], hyperpartisan-news detection [14], and explainable transformer-based news-title classification [15]—consistently report that performance remains highly sensitive to preprocessing choices, domain coverage, and label distributions, and that transformer models now define the state of the art for the language. Beyond text classification, the broader BNLP landscape has diversified rapidly to include tasks such as gender inference from Bangla names [16], Bangla Sign Language recognition [17], and neural machine translation between English and Bangla [18], underscoring both the pace of progress in the field and the continued fragmentation of resources and benchmarks across tasks.

Methodologically, two model families dominate. Classical algorithms—Logistic Regression [19], Naïve Bayes [20], Support Vector Machines (SVM) [1], [21], Random Forests [22], and gradient-boosted trees such as XGBoost [23]—remain strong when combined with sparse representations such as bag-of-words or TF-IDF [1], [10]. Deep architectures, notably Long Short-Term Memory (LSTM) networks [24], their bidirectional variants [25], and Transformer-based models such as BERT [2], [26], instead learn distributed, context-sensitive representations over sequences, supported by pretrained embeddings [27]–[29] and cross-lingual encoders such as XLM-R [30] and LASER [31]. In this work we focus on LSTM with learned embeddings and monolingual BERT models fine-tuned for Bangla and English, respectively.

This paper addresses the absence of a large-scale, unified comparison of these families across native Bangla and English news corpora; the specific research gaps are formalized in Section II-F. Our main contributions are as follows:

1) **Unified Multilingual Framework:** A single preprocessing and evaluation pipeline—covering tokenization, feature extraction (Bag-of-Words, One-Hot Encoding, TF-IDF), and stratified splitting—applied identically to both

languages, enabling controlled and reproducible cross-language comparison.
2) **Large-Scale Benchmarking:** Eight models (Logistic Regression, Naïve Bayes, SVM, Random Forest, XGBoost, BiLSTM, BERT, and a hybrid LSTM+BERT) benchmarked on 434,563 Bangla and 127,600 English news articles with a stratified 80:10:10 split—one of the largest comparative evaluations in multilingual text classification to date.
3) **Morphological Complexity Analysis:** A quantitative inflectional-density metric for Bangla showing that 98.8% of articles fall into medium-to-high complexity ranges, offering empirical evidence that explains the performance advantage of sequential and contextual models on Bangla.
4) **Cross-Language Insights and Practical Guidelines:** BERT achieves the highest accuracy for both Bangla (92.95%) and English (94.50%); the hybrid LSTM+BERT provides only marginal gains for Bangla ($\Delta = 0.02\%$) and slightly lower performance for English (94.20%); and SVM is the strongest classical model for both languages, confirming the advantage of transformer-based approaches across resource conditions.

The remainder of this paper is organized as follows. Section II reviews related work and formalizes the research gaps and positioning of this work. Section III describes the methodology, including dataset collection, preprocessing, feature extraction, and model architectures. Section IV details the experimental setup, including a dedicated hyperparameter justification. Section V presents the results. Section VI provides an expanded discussion with a comparison to recent state-of-the-art studies, and Section VII concludes.

II. Related Work

A. Classical Machine Learning for Text Classification

Early text classification systems relied on sparse, handcrafted representations such as bag-of-words and TF-IDF combined with statistical learning algorithms. Logistic Regression traces back to the regression analysis of binary sequences introduced by Cox [19] and remains widely used for multi-class text classification with one-versus-rest strategies. Naïve Bayes, whose probabilistic indexing foundations were established by Maron [20], is effective for document classification due to its simplicity and low computational cost in high-dimensional spaces. Support Vector Machines, formulated by Cortes and Vapnik [21], were shown by Joachims to be a strong baseline for large-scale text categorization with high-dimensional sparse features [1]. Random Forests, introduced by Breiman [22], combine bootstrap-aggregated decision trees with feature randomness to reduce variance, while gradient-boosted decision trees—most notably XGBoost by Chen and Guestrin [23]—extend boosting with regularization and optimized tree construction.

B. Deep Neural Architectures for Text Classification

Neural embedding methods drove the shift from sparse features to dense distributed representations: word2vec [27], GloVe [28], and FastText with subword information [29], the latter proving particularly useful for morphologically rich languages. On top of such embeddings, convolutional networks [3] and Hierarchical Attention Networks [4] achieved strong sentence- and document-level results. Recurrent architectures, especially LSTM networks introduced by Hochreiter and Schmidhuber [24], model long-range sequential dependencies through gating; bidirectional recurrent processing, introduced by Schuster and Paliwal [25], extends this to both temporal directions and is well suited to languages with flexible word order and complex morphology.

C. Pretrained Transformers and Multilingual Models

The Transformer architecture of Vaswani *et al.* [26] and the bidirectional pretraining paradigm of BERT [2] substantially advanced text classification, with variants such as RoBERTa and fine-tuning strategies pioneered by ULMFiT further improving downstream performance. In multilingual settings, XLM-R [30] and LASER [31] extend representation learning across many languages. Systematic evaluations show that word-order and morphological typology correlate with transformer performance [32], and industry studies analyze monolingual versus multilingual trade-offs in deployment [33]. Hybrid designs that stack recurrent layers over BERT representations have been explored for sentiment and service-request classification [34], [35], generally reporting modest gains that motivate the controlled evaluation performed here.

D. Bangla NLP and Text Classification

Bangla text classification historically relied on Naïve Bayes and SVM over bag-of-words and TF-IDF features, with recurrent models later improving on these baselines. Dhar *et al.* proposed a hybrid classical–neural approach for Bangla news categorization [10]; the Potrika corpus (664,880 articles in eight categories) enabled evaluations of LSTM, BiLSTM, GRU, and CNN models with several embedding schemes [36]; and a recent PLOS ONE study evaluated machine learning, deep learning, and hybrid stacking meta-classifiers on 118,404 Bangla news articles [9]. Transformer-based methods now dominate: Alam *et al.* fine-tuned multilingual transformers across several Bangla classification domains and reported state-of-the-art results [37]; monolingual BanglaBERT [38] provides a dedicated pretrained encoder; and the BNLP toolkit supplies preprocessing resources [7]. The survey of Alam *et al.* [5] concludes that transformers significantly outperform earlier methods while remaining constrained by data and compute. Most recently, transformer-based Bangla systems have been extended to explainable news-title classification (2025) [15], semi-supervised hyperpartisan news detection reaching 95.65% binary accuracy (2025) [14], hate-speech identification in the BLP-2025 shared task [13], and large newly curated news corpora such as BNAD (2024) [39].

E. Multilingual and Cross-Lingual Text Classification

Multilingual transformers such as mBERT and XLM-R narrow the gap between high- and low-resource languages via shared parameters and subword vocabularies [30], with

evaluations analyzing typological effects and training-data imbalance [32], [33]. In the Bangla context, multilingual transformers have been evaluated alongside other languages [37], and a comparative Bangla–English sentiment study on e-commerce reviews examined machine learning, deep learning, and hybrid models, but relied on translated English data and modest corpus sizes [40]. Unified evaluations of classical and deep models on large native Bangla and English news corpora remain scarce.

F. Research Gaps and Positioning of This Work

Synthesizing the literature above, three concrete research gaps emerge.

- **Gap 1 (No unified cross-language benchmark).** Existing Bangla news-classification studies evaluate only subsets of classical or deep models and rarely include a matched English benchmark under identical preprocessing and evaluation conditions [9], [10], [36]. Conversely, multilingual transformer evaluations compare transformer families alone, without strong classical baselines on large-scale Bangla news data [32], [33].
- **Gap 2 (Domain and data mismatch in cross-language comparisons).** The closest cross-language comparison operates on e-commerce sentiment with translated—rather than native—English data and considerably smaller corpora [40], leaving open how conclusions transfer to large native news corpora.
- **Gap 3 (No quantitative link between morphology and model performance).** Although morphological richness is repeatedly invoked to explain Bangla results [5], [8], no prior Bangla news-classification study quantifies article-level morphological complexity and relates it empirically to the performance of different model families.

Positioning and Novelty. This work closes all three gaps simultaneously. Specifically, it:

(i) benchmarks five classical and three deep architectures, including a hybrid LSTM+BERT model, under a unified experimental pipeline on 434,563 native Bangla and 127,600 native English news articles;
(ii) avoids translation artifacts entirely by using native corpora in both languages; and
(iii) introduces a quantitative inflectional-density metric together with a complexity-stratified accuracy analysis that, for the first time in Bangla news classification, empirically connects morphological complexity to the widening advantage of contextual deep models.

Consequently, this study provides both the largest unified Bangla–English news-classification benchmark to date and a principled, data-driven explanation of when and why transformer-based models justify their computational cost in low-resource, morphologically rich settings.

III. Methodology

A. Overview

This study proposes a unified, language-agnostic pipeline for multilingual text classification, applied consistently to both Bangla and English, enabling controlled and reproducible cross-language comparison. The pipeline comprises six sequential stages, as illustrated in Fig. 1: dataset collection, data preprocessing, feature extraction, model training, morphological complexity analysis (Bangla-specific, run in parallel with evaluation), and evaluation on a stratified held-out test set using accuracy, precision, recall, and weighted F1-score.

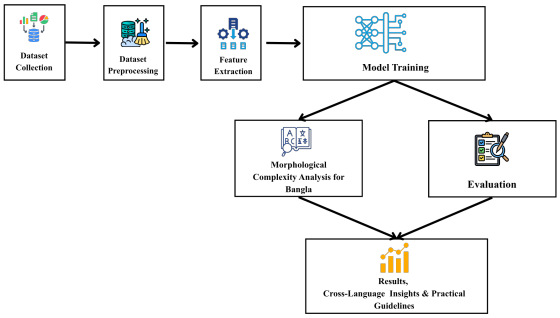


Fig. 1. Overall workflow of the proposed methodology for multilingual text classification. The pipeline comprises six sequential stages applied to both Bangla and English datasets. Stage 5 (morphological complexity analysis) is Bangla-specific and runs in parallel with Stage 6 (evaluation).

- **Dataset Collection:** Large-scale news corpora are curated from publicly available sources for each language, comprising 434,563 Bangla articles across 15 categories and 127,600 English articles across 4 categories.
- **Data Preprocessing:** Raw text is passed directly to downstream feature extraction tools, which apply implicit lowercasing, regex-based tokenization, and minimum document frequency filtering for classical models. For deep learning models, the Keras `Tokenizer` handles sequence tokenization for LSTM, while HuggingFace `AutoTokenizer` applies WordPiece subword tokenization for BERT and the hybrid architecture.
- **Feature Extraction:** Numerical representations are derived from preprocessed text using Bag-of-Words (BoW), One-Hot Encoding, and Term Frequency–Inverse Document Frequency (TF-IDF) for classical models, and sequence-based learned or pretrained contextual embeddings for deep learning models.
- **Model Training:** Five classical machine learning models—Logistic Regression, Naïve Bayes, SVM, Random Forest, and XGBoost—and three deep learning architectures—BiLSTM, BERT, and a hybrid LSTM+BERT—are trained and evaluated under identical experimental conditions across both languages.
- **Morphological Complexity Analysis:** A quantitative inflectional density metric is introduced for Bangla to measure morphological complexity at the article level and examine its relationship with downstream classification performance.
- **Evaluation:** All models are assessed on a held-out test set using a stratified 80:10:10 train-validation-test split, reporting accuracy, precision, recall, and weighted F1-score.

TABLE I
SUMMARY STATISTICS OF THE BANGLA AND ENGLISH DATASETS

Property	Bangla	English
Source	Prothom Alo	AG News
Total Articles	434,563	127,600
Categories	15	4
Training Set	347,650 (80%)	102,080 (80%)
Validation Set	43,456 (10%)	12,760 (10%)
Test Set	43,457 (10%)	12,760 (10%)
Splitting	Stratified 80:10:10	

B. Dataset Collection

Two datasets representing contrasting resource conditions were employed in this study: a large-scale Bangla news corpus and a well-established English news benchmark. Table I summarizes the key statistics of both datasets.

Bangla Dataset. The Bangla corpus consists of news articles collected from *Prothom Alo*, one of Bangladesh's largest and most widely circulated national daily newspapers [41]. The raw corpus comprises 437,948 records, of which 437,870 are valid articles spanning 32 category labels. Following standard practice, categories with fewer than 1,000 samples were excluded, yielding a final corpus of 434,563 articles across 15 categories: Bangladesh, Sports, International, Entertainment, Economy, Opinion, Technology, Lifestyle, Education, Durporobash (expatriate affairs), North America, Pachmisheli (miscellaneous topics), We-Are (community section), Onnoalo (lifestyle supplement), and Roshalo (entertainment supplement). The dataset was sourced from the publicly available Kaggle repository.

English Dataset. The English corpus is drawn from the AG News dataset [42], a widely adopted benchmark for English news classification. It comprises 127,600 articles uniformly distributed across four categories: World, Sports, Business, and Science/Technology. The dataset was obtained via the Hugging Face `datasets` library.

Data Splitting. Both datasets were partitioned using stratified sampling to preserve class distribution across all subsets. A fixed 80:10:10 train-validation-test split was applied, as summarized in Table I.

C. Data Preprocessing

Rather than implementing a custom multi-stage preprocessing pipeline, this study adopts an implicit preprocessing strategy in which normalization and tokenization are handled by the downstream feature-extraction tools. This design ensures reproducibility, eliminates preprocessing-induced variance across model comparisons, and allows each model family to use its native tokenization mechanism. The preprocessing strategy differs across the three model families, and Algorithm 1 formalizes the complete procedure.

Classical Machine Learning Models. Raw text was fed directly into scikit-learn's `TfidfVectorizer` and `CountVectorizer`, which automatically apply:

- **Lowercasing:** All text is converted to lowercase using the default `lowercase=True` parameter.
- **Tokenization:** Tokens are extracted using the built-in regular expression `(?u)\b\w\w+\b`, selecting word tokens containing at least two characters while implicitly removing punctuation and single-character tokens.
- **Minimum-frequency filtering:** Terms appearing in fewer than two documents are discarded using `min_df=2`, reducing vocabulary noise caused by extremely rare words.

LSTM Model. Text was tokenized using the Keras `Tokenizer` fitted on the training corpus, which performs whitespace-based tokenization and lowercasing. An out-of-vocabulary token (`<OOV>`) was reserved for unseen words. Tokenized sequences were converted into integer indices and post-padded or truncated to fixed sequence lengths of 500 tokens for Bangla and 200 tokens for English.

BERT and Hybrid LSTM+BERT. Text was tokenized using the HuggingFace `AutoTokenizer` corresponding to each pretrained model—`csebuetnlp/banglabert` for Bangla and `bert-base-uncased` for English. The tokenizer applies WordPiece subword tokenization, inserts the special tokens `[CLS]` and `[SEP]`, generates attention masks, and pads or truncates all sequences to a maximum length of 128 tokens, consistent with standard BERT fine-tuning practice [2].

D. Feature Extraction

Feature extraction transforms preprocessed text into numerical representations suitable for model input. Three sparse representations were constructed for classical machine learning models, and two sequence-based representations were used for deep learning architectures. All vectorizers were fitted exclusively on the training split and applied to validation and test sets without refitting, to prevent data leakage. Fig. 2 illustrates the complete feature-extraction workflow: preprocessed text follows one of three branches—sparse vectorization feeding the five classical models, integer sequences with learned embeddings feeding the BiLSTM, and WordPiece subword tokenization producing contextual token embeddings for BERT and the hybrid LSTM+BERT.

1) Bag-of-Words (BoW): The Bag-of-Words model represents each document as a vector of raw term frequencies over a fixed vocabulary. A vocabulary of 10,000 terms was retained using a unigram and bigram range (n-gram range $(1, 2)$) with a minimum document frequency of `min_df=2` to discard terms appearing in fewer than two documents.

2) One-Hot Encoding: One-Hot Encoding is a binary variant of BoW in which each entry indicates the presence or absence of a term in a document, irrespective of frequency. A vocabulary of 5,000 unigram terms was used (`binary=True`). This representation serves as a simple baseline to isolate the effect of term weighting.

3) TF-IDF: Term Frequency–Inverse Document Frequency (TF-IDF) reweights raw term counts to amplify discriminative terms while down-weighting terms that appear frequently across many documents. In its standard formulation, for a term t in document d over corpus $\mathcal{D}$:

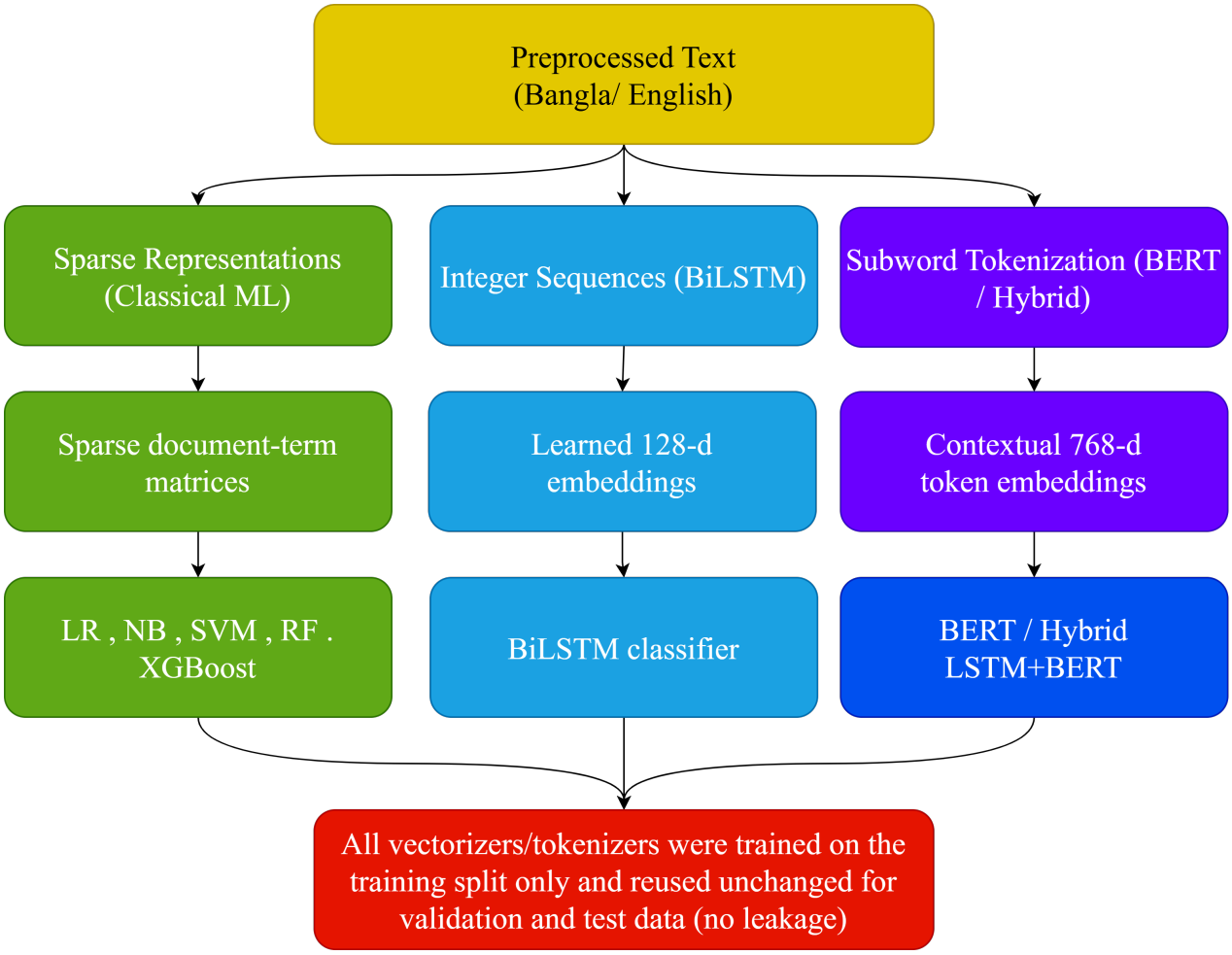


Fig. 2. Feature extraction workflow. Preprocessed text follows one of three branches: sparse vectorization (TF-IDF, Bag-of-Words, One-Hot) feeding the five classical models; integer sequences with learned 128-dimensional embeddings feeding the BiLSTM; and WordPiece subword tokenization producing contextual 768-dimensional token embeddings for BERT and the hybrid LSTM+BERT. All fitting is confined to the training split.

$$\text{TF-IDF}(t, d, \mathcal{D}) = \text{tf}(t, d) \times \log \frac{|\mathcal{D}|}{1 + |\{d' \in \mathcal{D} : t \in d'\}|} \quad (1)$$

In implementation, scikit-learn's smoothed variant of the inverse document frequency, $\text{idf}(t) = \log\left((1 + |\mathcal{D}|)/(1 + \text{df}(t))\right) + 1$ with L2 row normalization, was used. Sublinear TF scaling (`sublinear_tf=True`) replaces raw term frequency with $1 + \log(\text{tf}(t, d))$, compressing the influence of high-frequency terms. A vocabulary of 10,000 terms with bigram range $(1, 2)$ and `min_df=2` was used. TF-IDF was selected as the primary representation for all five classical ML models based on its well-established superiority over raw frequency counts for multi-class text classification tasks [1].

4) Sequence Representations for LSTM: For the LSTM model, text was encoded using a Keras `Tokenizer` fitted on the training corpus. Vocabulary sizes were set to 50,000 tokens for Bangla and 30,000 tokens for English, and maximum sequence lengths were set to 500 and 200 tokens respectively. The larger vocabulary and sequence length for Bangla reflect its higher morphological diversity, where a single root word can generate dozens of inflected surface forms, resulting in a substantially larger unique token space and longer average article length compared to English. An out-of-vocabulary token (`<OOV>`) was reserved for unseen words at inference time. Tokenized sequences were converted to integer indices and padded or truncated to fixed maximum lengths using post-sequence padding.

5) Subword Tokenization for BERT and Hybrid: For BERT and the hybrid LSTM+BERT model, text was tokenized using the HuggingFace `AutoTokenizer` corresponding to each pretrained model: `csebuetnlp/banglabert` for Bangla and `bert-base-uncased` for English. The tokenizer applies WordPiece subword tokenization, inserts `[CLS]` and `[SEP]` special tokens, and generates attention masks to distinguish real tokens from padding. All sequences were padded to a maximum length of 128 tokens with truncation for longer inputs, consistent with standard BERT fine-tuning practice [2].

Table II summarizes all feature extraction configurations.

E. Classical Machine Learning Models

Five classical machine learning models were implemented using scikit-learn [43]. All five models were trained and evaluated

TABLE II
FEATURE EXTRACTION CONFIGURATIONS

Method	Models	Vocab.	n-gram	Key Parameters
TF-IDF	LR, NB, SVM, RF, XGB	10,000	(1,2)	`min_df=2`, `sublinear_tf=True`
Bag-of-Words	LR, NB, SVM, RF, XGB	10,000	(1,2)	`min_df=2`, raw counts
One-Hot	LR, NB, SVM, RF, XGB	5,000	(1,1)	`binary=True`, `min_df=2`
Keras Seq. (Bangla)	BiLSTM	50,000	—	`max_len=500`, `<OOV>`, post-padding
Keras Seq. (English)	BiLSTM	30,000	—	`max_len=200`, `<OOV>`, post-padding
AutoTokenizer	BERT, Hybrid	Pretrained	—	WordPiece, `max_len=128`, `[CLS]/[SEP]`, attention masks

Algorithm 1 Unified Implicit Data Preprocessing

Require: Raw corpus $D = \{(x_i, y_i)\}$, language $L \in \{\text{Bangla}, \text{English}\}$, model family $M \in \{\text{Classical}, \text{LSTM}, \text{BERT/Hybrid}\}$
Ensure: Model-ready inputs X and labels y for training, validation, and testing
Remove invalid records and discard classes containing fewer than 1,000 samples.
Split D into $(D_{\text{train}}, D_{\text{val}}, D_{\text{test}})$ using stratified sampling (80:10:10, random seed = 42).
if $M = \text{Classical}$ **then**
    Fit `CountVectorizer`/`TfidfVectorizer` on D_{train} (`lowercase=True`, token pattern `(?u)\b\w\w+\b`, `min_df=2`)
    Transform D_{train}, D_{val}, and D_{test} without refitting.
else if $M = \text{LSTM}$ **then**
    Fit the Keras `Tokenizer` with an `<OOV>` token on D_{train}.
    Set vocabulary size to 50,000 for Bangla or 30,000 for English.
    Convert texts into integer sequences and post-pad/truncate to a maximum length of 500 tokens (Bangla) or 200 tokens (English).
else
    Initialize the HuggingFace `AutoTokenizer` (`csebuetnlp/banglabert` for Bangla or `bert-base-uncased` for English).
    Apply WordPiece tokenization, insert `[CLS]` and `[SEP]`, generate attention masks, and pad/truncate all sequences to 128 tokens.
end if
return (X, y) for the training, validation, and test sets.

across all three feature representations (TF-IDF, BoW, and One-Hot Encoding). TF-IDF was used as the primary representation for cross-model performance comparison, as it consistently yielded the highest validation accuracy across all models and both languages. A fixed random seed (`random_state=42`) was applied to all stochastic models to ensure reproducibility. Table III summarizes the hyperparameter configurations for all five models.

Logistic Regression (LR) [19]. Logistic Regression models the posterior probability of each class using a softmax function over a linear combination of input features. The model was configured with L2 regularization ($C = 1.0$), the L-BFGS solver with multi-class handling set to `auto`, and a maximum of 1,000 iterations. Parallel processing was enabled via `n_jobs=-1`.

Naïve Bayes (NB) [20]. Multinomial Naïve Bayes applies Bayes' theorem under the conditional independence assumption, modelling term frequencies as multinomially distributed. The smoothing parameter was set to $\alpha = 1.0$, making it the simplest and fastest model in the suite.

Support Vector Machine (SVM) [1], [21]. A Linear Support Vector Classifier (`LinearSVC`) was employed, which optimizes a linear hinge loss objective with L2 regularization ($C = 1.0$). The dual formulation was used with a maximum of 2,000 iterations. `LinearSVC` is well-suited to high-dimensional sparse TF-IDF representations due to its linear complexity with respect to the number of samples and features.

Random Forest (RF) [22]. An ensemble of 100 decision trees was constructed via bootstrap aggregation (`n_estimators=100`). Trees were grown to full depth (`max_depth=None`) using default splitting criteria (`min_samples_split=2`, `min_samples_leaf=1`). Parallel tree construction was enabled using all available CPU cores (`n_jobs=-1`).

XGBoost [23]. Gradient boosted trees were trained using the XGBoost native API with DMatrix format for efficient sparse matrix handling. A multi-class softmax objective was used with 100 maximum boosting rounds and early stopping triggered after 10 rounds without validation loss improvement. Key hyperparameters include maximum tree depth of 6, learning rate ($\eta = 0.1$), and stochastic subsampling of 0.8 for both rows (`subsample`) and columns per tree (`colsample_bytree`).

F. Deep Learning Models

All three deep learning models were implemented in PyTorch and trained on an NVIDIA H100 GPU. Mixed-precision training via `torch.cuda.amp.GradScaler` was enabled for BERT and the hybrid model to reduce memory consumption and accelerate training. The best model checkpoint was selected based on minimum validation loss, and final evaluation was performed on the held-out test set.

1) Bidirectional LSTM (BiLSTM) [24], [25]: A Bidirectional Long Short-Term Memory network was implemented as a custom `nn.Module` (class `BiLSTMClassifier`). Input token sequences are first passed through a learned `Embedding`

TABLE III
HYPERPARAMETER CONFIGURATIONS FOR CLASSICAL ML MODELS

Model	Parameter	Value
Logistic Regression [19]	Regularization (C)	1.0
	Solver	L-BFGS
	Max iterations	1,000
Naïve Bayes [20]	Smoothing (α)	1.0
SVM (LinearSVC) [21]	Regularization (C)	1.0
	Max iterations	2,000
	Formulation	Dual
Random Forest [22]	No. of trees	100
	Max depth	None (full)
	Min samples split	2
XGBoost [23]	Boosting rounds	100
	Learning rate (η)	0.1
	Max depth	6
	Subsample	0.8

layer of dimension 128, followed by `Dropout(0.3)`. The embedded sequence is then processed by a two-layer Bidirectional LSTM with hidden size 128 per direction, producing a combined hidden state of dimension $128 \times 2 = 256$ by concatenating the final forward and backward hidden states. This representation is passed through a three-layer classification head: `Linear(256, 128)` $\rightarrow$ `ReLU` $\rightarrow$ `Dropout(0.3)` $\rightarrow$ `Linear(128, 64)` $\rightarrow$ `ReLU` $\rightarrow$ `Dropout(0.2)` $\rightarrow$ `Linear(64, C)`, where C denotes the number of output classes (15 for Bangla, 4 for English). This yields 7.1M trainable parameters for Bangla and 4.5M for English.

Training used the Adam optimizer (lr $= 0.001$, $\lambda = 10^{-5}$) with cross-entropy loss. A `ReduceLROnPlateau` scheduler (factor=0.5, patience=2) was applied to reduce the learning rate when validation loss stagnated. Training ran for a maximum of 15 epochs with a batch size of 256 and early stopping with patience 3, yielding best checkpoints at epoch 6 for Bangla and epoch 12 for English.

2) BERT-Based Classifier [2], [26]: Two monolingual pretrained transformer models were fine-tuned for sequence classification using HuggingFace's `AutoModelForSequenceClassification` API: the `csebuetnlp/banglabert` model [38] for Bangla (ELECTRA-based, 110.6M parameters) and `bert-base-uncased` [2] for English (110.0M parameters). Both models use the built-in HuggingFace classification head, which applies a linear projection over the `[CLS]` token representation to produce class logits.

Fine-tuning used the AdamW optimizer (lr $= 2 \times 10^{-5}$, weight decay $= 0.01$) with a `OneCycleLR` scheduler (`pct_start`=0.1, `anneal_strategy=linear`). Training ran for a maximum of 5 epochs with a batch size of 64 and early stopping with patience 2, yielding best checkpoints at epoch 4 for Bangla and epoch 3 for English. Mixed-precision training was enabled throughout.

TABLE IV
HYPERPARAMETER CONFIGURATIONS FOR DEEP LEARNING MODELS

Parameter	BiLSTM	BERT	Hybrid
Framework	PyTorch	PyTorch	PyTorch
Batch size	256	64	32
Max epochs	15	5	5
Early stop patience	3	2	2
Learning rate	10^{-3}	2×10^{-5}	2×10^{-5}
Optimizer	Adam	AdamW	AdamW
LR scheduler	ReduceLROnPlateau	OneCycleLR	OneCycleLR
Mixed precision	No	Yes	Yes
Bangla best epoch	6	4	5
English best epoch	12	3	5

3) Hybrid LSTM+BERT Architecture [34], [35]: The proposed hybrid model (`HybridBERTLSTM`) integrates a pretrained BERT backbone with a Bidirectional LSTM to explicitly model sequential dependencies over BERT's contextual token representations. The architecture operates as follows: the full token sequence output from BERT (shape $T \times 768$, where T is the sequence length) is passed into a two-layer Bidirectional LSTM with hidden size 256 per direction. The final hidden states from both directions are concatenated to yield a $256 \times 2 = 512$ dimensional representation, which is then passed through: `Dropout(0.3)` $\rightarrow$ `Linear(512, 256)` $\rightarrow$ `ReLU` $\rightarrow$ `Dropout(0.3)` $\rightarrow$ `Linear(256, C)`.

To balance representational capacity with training efficiency, the BERT backbone was partially frozen: the bottom layers were kept fixed while the top 4 transformer layers remained trainable. This yields 32.2M trainable and 81.7M frozen parameters for Bangla (113.9M total), and 32.8M trainable and 80.5M frozen for English (113.3M total). Training used the same AdamW optimizer and `OneCycleLR` scheduler as the standalone BERT model (lr $= 2 \times 10^{-5}$), with batch size 32, a maximum of 5 epochs, and early stopping with patience 2, yielding best checkpoints at epoch 5 for both languages.

Table IV summarizes the hyperparameter configurations for all three deep learning models.

G. Morphological Complexity Analysis

Bangla is a morphologically rich language in which a single root word can generate numerous surface forms through inflectional processes involving tense, aspect, number, case, and gender agreement. To quantify this property at the document level and examine its relationship with downstream classification performance, we introduce a word-level inflectional density metric applied to the entire Bangla corpus.

Metric Definition. For each article d, we define the morphological complexity score $\mathcal{C}(d)$ as the proportion of word tokens that contain at least one suffix from a predefined set of Bangla inflectional markers $\mathcal{M}$:

$$\mathcal{C}(d) = \frac{|\{w \in d : \exists m \in \mathcal{M},\ m \subseteq w\}|}{|d|} \tag{2}$$

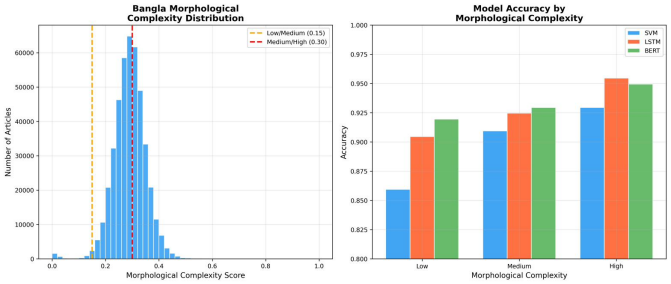


Fig. 3. Bangla morphological complexity analysis. *Left:* Distribution of article-level complexity scores with Low/Medium/High thresholds (0.15, 0.30). *Right:* Mean complexity score per news category.

where w denotes a word token, m a suffix marker, and $|d|$ the total number of tokens in article d.

Inflectional Marker Set. The marker set $\mathcal{M}$ comprises 17 Bangla inflectional suffixes drawn from verbal, nominal, and aspectual paradigms, including present and past tense verbal endings (*-chhi*, *-chhe*, *-chhil*, *-echi*, *-eche*), progressive markers (*-cchhi*, *-cchhe*), future tense markers (*-bo*, *-be*), locative and instrumental case endings (*-te*, *-ye*), perfective endings (*-lam*, *-len*), and nominal definiteness and plurality suffixes (*-ta*, *-ti*, *-gulo*).

Complexity Categories. Articles are assigned to one of three complexity levels based on empirically motivated thresholds:

- **Low:** $\mathcal{C}(d) < 0.15$
- **Medium:** $0.15 \leq \mathcal{C}(d) < 0.30$
- **High:** $\mathcal{C}(d) \geq 0.30$

Applied to the full Bangla corpus of 434,563 articles, the metric yields a mean complexity score of $\mu = 0.2905$ ($\sigma = 0.0594$, median $= 0.2913$), with 4,739 articles (1.1%) classified as Low, 241,764 (55.6%) as Medium, and 187,912 (43.2%) as High complexity. Consequently, 98.8% of all articles fall into the medium-to-high range, confirming the pervasive morphological richness of the corpus. Complexity scores vary across content categories, with Technology ($\bar{\mathcal{C}} = 0.3289$) and Economy ($\bar{\mathcal{C}} = 0.3284$) exhibiting the highest mean complexity and Education ($\bar{\mathcal{C}} = 0.2104$) the lowest. The distribution of complexity scores and per-category breakdown are presented in Fig. 3.

H. Evaluation Metrics

All models were evaluated on the held-out test set using four standard classification metrics: Accuracy, Precision, Recall, and F1-score. For multi-class settings, Precision, Recall, and F1-score were computed using weighted averaging, where each class contributes proportionally to its support (number of true instances). This averaging strategy accounts for the class imbalance present in the Bangla dataset, where the *bangladesh* category alone contributes approximately 53.5% of all test samples.

Let TP_c, FP_c, and FN_c denote the true positives, false positives, and false negatives for class c, respectively, and let n_c denote the number of test samples in class c. The metrics are defined as:

TABLE V
SOFTWARE ENVIRONMENT

Package	Version
Python	3.12
PyTorch	2.8.0 (CUDA 12.8)
TensorFlow	2.21.0
Transformers	4.40.0
XGBoost	3.2.0
NumPy	1.26.4
Pandas	2.1.4
scikit-learn	1.x

$$\text{Accuracy} = \frac{\sum_c TP_c}{N} \tag{3}$$

$$\text{Precision}_w = \sum_c \frac{n_c}{N} \cdot \frac{TP_c}{TP_c + FP_c} \tag{4}$$

$$\text{Recall}_w = \sum_c \frac{n_c}{N} \cdot \frac{TP_c}{TP_c + FN_c} \tag{5}$$

$$\text{F1}_w = \sum_c \frac{n_c}{N} \cdot \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{6}$$

where $N = \sum_c n_c$ is the total number of test samples. The weighted F1-score serves as the primary ranking metric throughout this study, as it jointly captures both precision and recall while accounting for class distribution.

IV. EXPERIMENTAL SETUP

A. Computing Infrastructure

Experiments were conducted across two cloud platforms depending on the computational requirements of each model family. Classical machine learning models (Logistic Regression, Naïve Bayes, SVM, and Random Forest) were trained on Google Colaboratory with a T4 GPU backend, using CPU-bound scikit-learn operations. XGBoost and all deep learning models (BiLSTM, BERT, and Hybrid LSTM+BERT) were trained on the Lightning AI cloud platform equipped with a single NVIDIA H100 80GB HBM3 GPU (85 GB GPU memory, 168.4 GB system RAM, CUDA 12.8).

B. Software Environment

The complete software stack is summarized in Table V. Classical ML models used scikit-learn within a TensorFlow 2.21.0 environment on Google Colab. Deep learning models were implemented in PyTorch 2.8.0 on Lightning AI. HuggingFace Transformers 4.40.0 was used for pretrained model loading and tokenization.

C. Hyperparameter Selection and Justification

Hyperparameters were determined through a three-tier strategy. First, for classical models, community-standard defaults with strong published track records were adopted, preserving comparability with prior benchmarks and avoiding per-model tuning bias in an eight-model comparison. Second, deep-learning optimization settings followed the original publications and established fine-tuning recipes—for example, the 2×10^{-5} AdamW learning rate recommended for BERT fine-tuning [2]—and were verified on the validation split. Third, capacity- and language-dependent parameters (vocabulary sizes, sequence lengths, batch sizes) were selected from corpus statistics and hardware memory constraints, again validated on the held-out validation set through early stopping and checkpointing on minimum validation loss. No test-set information influenced any selection. Table VI summarizes each key hyperparameter, its value, and its justification.

D. Reproducibility

All stochastic components were fixed to a global random seed of 42. For PyTorch operations, `torch.manual_seed(42)`, `torch.cuda.manual_seed(42)`, and `torch.backends.cudnn.deterministic = True` were set prior to training. Scikit-learn models used `random_state=42`, XGBoost used `seed=42`, and NumPy and TensorFlow random seeds were set to 42. Dataset splitting used stratified sampling with the same seed, ensuring identical class distributions across all train, validation, and test sets.

E. Model Selection and Checkpointing

A checkpoint-based system was employed to enable fault-tolerant training across sessions. For all deep learning models, the best model weights were saved based on minimum validation loss and reloaded for final test evaluation. Early stopping was applied with patience of 3 epochs for BiLSTM and 2 epochs for BERT and Hybrid models. Training times for all models are summarized in Table VII.

V. Results and Analysis

This section presents the experimental results for all eight models across both Bangla and English datasets. Table VIII reports the primary performance metrics (Accuracy, Precision, Recall, and weighted F1-score) for each model using TF-IDF features for classical ML models and sequence-based representations for deep learning models. Fig. 4 visualises the accuracy and F1-score comparison across all models and languages.

A. Overall Model Performance

The results in Table VIII reveal several notable patterns. For Bangla, the proposed Hybrid LSTM+BERT architecture achieves the highest accuracy (92.97%) and F1-score (92.95%), marginally exceeding standalone BERT (92.95%, $\Delta = 0.02\%$). For English, BERT achieves the highest performance (94.50%), outperforming the Hybrid by 0.30%. Among classical models, SVM achieves the highest accuracy for both Bangla (90.95%) and English (91.03%) with TF-IDF features. Notably, the best deep learning model outperforms the best classical model by 2.02% for Bangla and 3.46% for English, indicating a larger advantage of transformer-based models for the resource-rich English setting. Random Forest yields the weakest performance overall, falling 10.24% below SVM for Bangla and 3.49% below for English, likely due to the poor handling of high-dimensional sparse features under bootstrap subsampling.

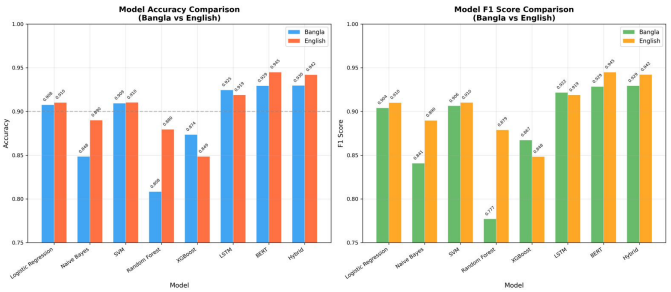


Fig. 4. Accuracy and weighted F1-score comparison across all eight models for Bangla (blue) and English (orange).

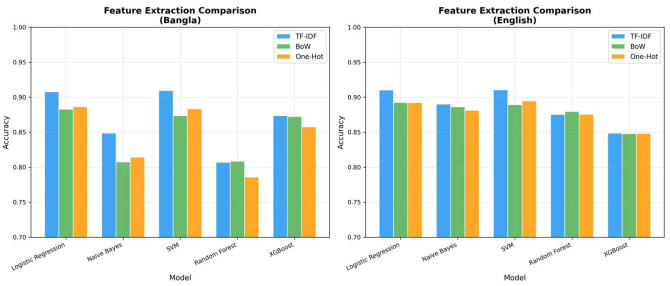


Fig. 5. Accuracy comparison across feature extraction strategies (TF-IDF, BoW, One-Hot) for all five classical ML models on Bangla (left) and English (right).

B. Feature Extraction Comparison

Table IX and Fig. 5 compare the accuracy of all five classical models across the three feature representations: TF-IDF, BoW, and One-Hot Encoding.

TF-IDF consistently outperforms BoW and One-Hot Encoding for almost all models and both languages, confirming its well-established superiority for multi-class text classification. The sublinear term frequency scaling and inverse document frequency weighting effectively amplify discriminative terms while suppressing corpus-wide common terms. The sole exception is Random Forest, for which BoW slightly outperforms TF-IDF on both Bangla (0.8084 vs 0.8071) and English (0.8795 vs 0.8755), possibly because TF-IDF's continuous reweighting introduces additional split complexity for tree-based learners. One-Hot Encoding consistently ranks lowest or comparably to BoW, as the absence of frequency information is particularly harmful for morphologically rich Bangla text.

C. Deep Learning vs. Classical Machine Learning

Fig. 6 shows the Precision, Recall, and F1-score for all models. Deep learning models consistently outperform classical

TABLE VI
HYPERPARAMETER VALUES AND THEIR JUSTIFICATION

Component	Hyperparameter	Value	Justification
Vectorizers	Vocabulary / n-gram / `min_df`	10,000 / (1,2) / 2	Bigram range captures inflected collocations; `min_df=2` removes noise; 10k vocabulary caps memory at 434k documents
TF-IDF	`sublinear_tf`	True	Log-scaled TF compresses high-frequency terms; standard for multi-class text classification [1]
LR [19]	C / solver / max iter.	1.0 / L-BFGS / 1,000	scikit-learn defaults with proven performance on sparse text; L-BFGS handles the multinomial objective efficiently
NB [20]	α	1.0	Laplace smoothing default; robust to unseen terms
SVM [21]	C / max iter.	1.0 / 2,000	Default margin softness; iteration cap doubled to guarantee convergence on 434k documents
RF [22]	Trees / depth	100 / full	Standard ensemble size; full depth lets trees exploit sparse splits
XGBoost [23]	η / depth / subsample / rounds	0.1 / 6 / 0.8 / 100, early stop 10	Author-recommended defaults; subsampling regularizes; early stopping on validation loss prevents overfitting
BiLSTM [24], [25]	Embed / hidden / layers / dropout	128 / 128×2 / 2 / 0.3	Balances capacity within a 7.1M-parameter budget; dropout counters overfitting on frequent classes
BiLSTM optim.	lr / batch / epochs	10^{-3} / 256 / $\leq$15 (patience 3)	Adam default learning rate; large batch exploits H100 memory; validation-loss early stopping selects the checkpoint
BERT [2]	lr / batch / epochs	2×10^{-5} / 64 / $\leq$5 (patience 2)	Canonical BERT fine-tuning recipe [2]; OneCycleLR warm-up (`pct_start`=0.1) stabilizes early fine-tuning
Hybrid	Trainable layers / LSTM hidden / batch	Top 4 / 256×2 / 32	Partial freezing (32M trainable) balances adaptation with efficiency; smaller batch fits the 113.9M-parameter model in memory
Sequences	Max length (Bn / En)	500 / 200 (LSTM); 128 (BERT)	Matches corpus length statistics; 128 subword tokens follows standard BERT practice [2]
Global	Random seed	42	Fixed across NumPy, TensorFlow, PyTorch, scikit-learn, and XGBoost for full reproducibility

TABLE VII
TRAINING TIMES FOR ALL MODELS (TF-IDF FOR CLASSICAL ML)

Model	Platform	Bangla	English
Logistic Regression	Colab T4	4m 12s	7s
Naïve Bayes	Colab T4	5s	<1s
SVM	Colab T4	—*	—*
Random Forest	Colab T4	—*	—*
XGBoost	H100	1h 51m	9m 36s
BiLSTM	H100	5m 29s	1m 14s
BERT	H100	15m 10s	4m 13s
Hybrid LSTM+BERT	H100	32m 58s	9m 25s

* Loaded from saved checkpoint; exact time not recorded.

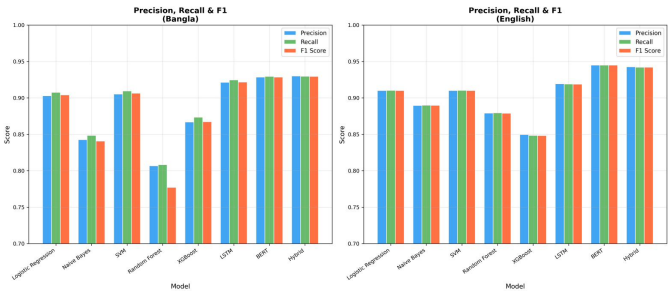


Fig. 6. Precision, Recall, and weighted F1-score for all models on Bangla (left) and English (right).

ML across all three metrics for both languages. The BiLSTM surpasses SVM by 1.51% accuracy for Bangla and 0.86% for English, demonstrating the value of sequential modelling even without pretraining. BERT improves upon SVM by 2.00% for Bangla and 3.46% for English, with the larger English gain reflecting the richer pretraining data available for English BERT (`bert-base-uncased`) compared to the Bangla-specific BanglaBERT. The Hybrid LSTM+BERT adds only 0.02% over BERT for Bangla ($\Delta = 0.0002$) and underperforms BERT by 0.30% for English, suggesting that BERT's self-attention mechanism already captures most of the sequential and morphological structure relevant for classification, with the additional recurrent layers providing negligible benefit.

Among classical models, the performance ordering is consistent across both languages: SVM > LR > NB for the top three models, reflecting the well-established efficacy of linear classifiers for high-dimensional sparse text representations. XGBoost underperforms LR by 3.41% for Bangla despite its strength on tabular tasks, consistent with findings in prior work that gradient-boosted trees struggle with sparse, high-dimensional feature matrices [23]. Random Forest ranks last among classical models for Bangla (80.71%), with a notably low weighted F1-score (77.52%) indicating disproportionate misclassification of minority categories.

D. Training Convergence

Fig. 7 presents the training and validation accuracy curves for all six deep learning runs. Several patterns are notable. The BiLSTM on Bangla converges steadily to a best validation checkpoint at epoch 6, while the English BiLSTM requires 12 epochs to reach its optimum. BERT fine-tuning converges quickly for both languages, reaching best validation loss at

TABLE VIII
CLASSIFICATION PERFORMANCE OF ALL MODELS (TF-IDF FOR CLASSICAL ML; SEQUENCE FOR DEEP LEARNING). BEST RESULT PER METRIC PER LANGUAGE IS **BOLD**; BEST CLASSICAL RESULT IS UNDERLINED. LR TIES SVM ON ENGLISH ACCURACY AND F1.

Model	Bangla				English			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Logistic Regression	0.9077	0.9029	0.9077	0.9041	0.9103	0.9102	0.9103	0.9102
Naïve Bayes	0.8485	0.8427	0.8485	0.8408	0.8900	0.8896	0.8900	0.8897
SVM	0.9095	0.9053	0.9095	0.9065	0.9103	0.9102	0.9103	0.9102
Random Forest	0.8071	0.8085	0.8071	0.7752	0.8755	0.8751	0.8755	0.8749
XGBoost	0.8736	0.8668	0.8736	0.8673	0.8485	0.8497	0.8485	0.8483
BiLSTM	0.9246	0.9213	0.9246	0.9217	0.9190	0.9195	0.9190	0.9189
BERT	0.9295	0.9286	0.9295	0.9286	**0.9450**	**0.9450**	**0.9450**	**0.9450**
Hybrid (LSTM+BERT)	**0.9297**	**0.9301**	**0.9297**	**0.9295**	0.9420	0.9428	0.9420	0.9422

TABLE IX
ACCURACY BY FEATURE EXTRACTION STRATEGY FOR CLASSICAL ML MODELS

Model	Bangla			English		
	TF	BoW	OH	TF	BoW	OH
LR	**.9077**	.8827	.8864	**.9103**	.8924	.8923
NB	**.8485**	.8076	.8142	**.8900**	.8861	.8813
SVM	**.9095**	.8735	.8835	**.9103**	.8891	.8946
RF	.8071	**.8084**	.7857	.8755	**.8795**	.8756
XGB	**.8736**	.8724	.8576	**.8485**	.8479	.8480

TF = TF-IDF, BoW = Bag-of-Words, OH = One-Hot.

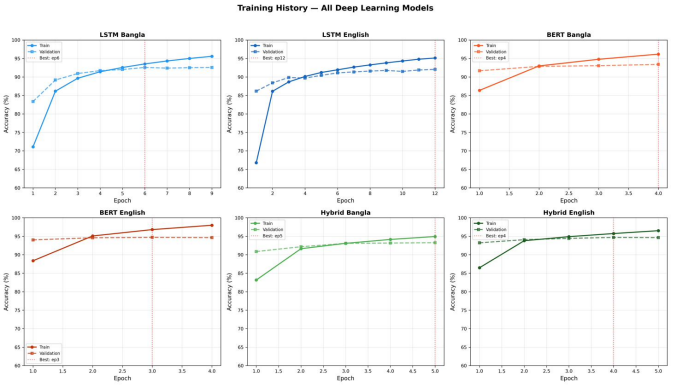


Fig. 7. Training and validation accuracy curves for all six deep learning training runs (BiLSTM, BERT, and Hybrid for Bangla and English). Dotted vertical lines indicate the best checkpoint epoch.

epoch 4 (Bangla) and epoch 3 (English), consistent with the typical fast convergence observed in fine-tuning large pretrained models. The Hybrid model requires the most epochs (5 for both languages), as the partial BERT fine-tuning combined with LSTM training introduces a longer optimization trajectory. In all cases, training accuracy continues to increase after the best checkpoint, indicating that early stopping effectively prevents overfitting.

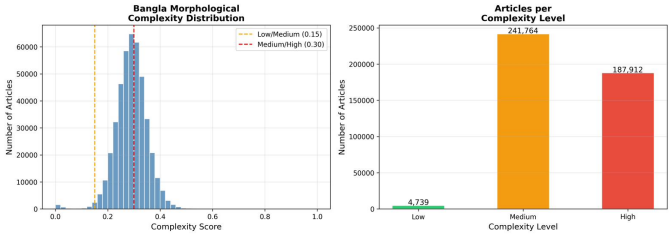


Fig. 8. *Left:* Distribution of article-level Bangla morphological complexity scores with Low/Medium/High thresholds (0.15, 0.30). *Right:* Model accuracy stratified by morphological complexity level for SVM, BiLSTM, and BERT.

E. Morphological Complexity Analysis

The morphological complexity analysis yields several important insights (Fig. 8). The distribution of complexity scores across 434,563 Bangla articles is approximately bell-shaped with mean $\mu = 0.2905$ ($\sigma = 0.0594$, median $= 0.2913$), concentrated in the Medium range (55.6%) with a substantial High tail (43.2%). Only 1.1% of articles are classified as Low complexity, confirming the pervasive morphological richness of Bangla news text.

Accuracy stratified by complexity level reveals that all three models benefit from higher morphological complexity, with deep learning models showing a more pronounced improvement. SVM accuracy increases from approximately 86% at Low complexity to 93% at High complexity, while BiLSTM rises from approximately 90% to 95% and BERT from 92% to 95% across the same range. The widening gap between SVM and deep learning models at higher complexity levels provides empirical support for the hypothesis that sequential and contextual representations are particularly advantageous for morphologically rich text, where surface-form sparsity limits the discriminative power of bag-of-words representations. Complexity also varies by content category: Technology ($\bar{C} = 0.3289$) and Economy ($\bar{C} = 0.3284$) exhibit the highest mean complexity, while Education ($\bar{C} = 0.2104$) is the least morphologically complex, likely reflecting differences in formal versus informal written register.

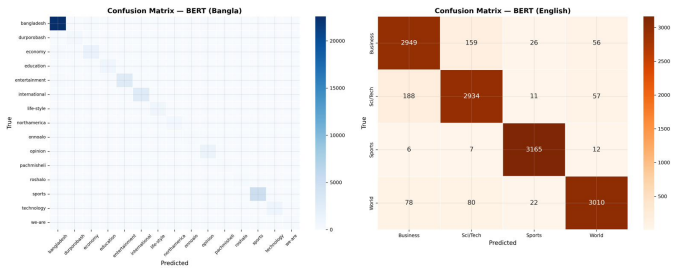


Fig. 9. Confusion matrices for BERT on Bangla (left, 15 classes) and English (right, 4 classes) test sets.

F. Confusion Matrix Analysis

Fig. 9 shows the confusion matrices for the best-performing BERT model. For English, BERT achieves near-perfect diagonal structure across all four categories. The highest confusion occurs between Business–Sci/Tech (159 misclassifications) and World–Business (78), reflecting genuine semantic overlap between these domains.

For Bangla, the dominant *bangladesh* class (23,250 test samples) is classified with high precision (96.84%) and recall (97.16%), benefiting from its large support and distinct topical vocabulary. The most challenging categories are minority classes: *we-are* (300 samples, F1: 33.64%), *pachmisheli* (344 samples, F1: 48.68%), and *onnoalo* (270 samples, F1: 68.35%), all of which exhibit editorial or lifestyle content with significant vocabulary overlap with other categories. The *northamerica* category (699 samples, F1: 67.69%) is also challenging, likely due to its topical similarity to the larger *international* category. These results suggest that performance for the Bangla dataset is substantially constrained by class imbalance rather than model capacity.

G. Cross-Language Comparison

Comparing model performance across languages reveals consistent structural patterns. First, all deep learning architectures perform higher on English than on Bangla: BERT achieves 94.50% on English versus 92.95% on Bangla (gap: 1.55%), attributable to the richer pretraining corpus and simpler 4-class task structure. Second, for classical models, the pattern is reversed for several models: SVM achieves nearly identical accuracy on both languages (91.03% English, 90.95% Bangla, gap: 0.08%), and LR similarly (91.03% vs 90.77%, gap: 0.26%). This suggests that TF-IDF representations with linear classifiers are more robust to language-specific morphological differences than one might expect, likely because the unigram-plus-bigram vocabulary partially captures inflectional surface forms.

Third, XGBoost shows a striking reversal: it ranks fifth among eight models for Bangla (87.36%) but last for English (84.85%), suggesting that gradient-boosted trees are particularly ill-suited to high-feature-count sparse matrices in the 4-class balanced English setting. Fourth, an unexpected reversal occurs for BiLSTM, which is stronger on Bangla (92.46%) than English (91.90%), plausibly because the far larger Bangla training set (347,650 vs. 102,080 samples) benefits a from-scratch sequential learner. Finally, the deep learning advantage over classical ML is larger for English (3.46%) than for Bangla (2.02%), indicating that the rich English pretraining data amplifies the transformer advantage, while the morphological complexity of Bangla partially narrows the gap by favouring sequential models even at the classical level.

VI. Discussion

This section interprets the experimental findings in the context of the research questions posed in this study, critically compares them with recent state-of-the-art studies, and derives the strengths, remaining limitations, and practical implications for multilingual and low-resource NLP.

A. Transformer Models vs. Classical Machine Learning

The results consistently demonstrate the superiority of transformer-based models over classical machine learning approaches for both Bangla and English text classification. The best deep learning model outperforms the best classical model by 2.02% for Bangla and 3.46% for English. However, the *magnitude* of this advantage is informative: the smaller gap for Bangla (2.02% vs. 3.46%) suggests that the deep learning advantage is partially attenuated by the morphological complexity of Bangla, where TF-IDF representations with bigram features partially capture inflectional surface forms that would otherwise require contextual modelling.

Among classical models, SVM with TF-IDF achieves the strongest performance for both languages (Bangla: 90.95%, English: 91.03%), consistent with its well-established efficacy for high-dimensional sparse text representations [1]. The near-identical SVM performance across languages (gap: 0.08%) is particularly striking, suggesting that linear classifiers are robust to the morphological and structural differences between Bangla and English when combined with sublinear TF-IDF weighting. This finding has practical implications: for resource-constrained deployments where deep learning inference is prohibitive, SVM with TF-IDF represents a strong, language-agnostic baseline.

Random Forest performs considerably worse than SVM for Bangla (80.71% accuracy, 77.52% weighted F1), a gap of 10.24%. This disparity is attributable to the well-known difficulty of tree-based models in high-dimensional sparse feature spaces: feature subsampling in Random Forest selects a random subset of features at each split, which in sparse TF-IDF matrices results in many trees being trained on uninformative zero-valued features. The lower weighted F1 (77.52%) relative to accuracy (80.71%) further indicates systematic misclassification of minority categories. XGBoost, despite its generally strong performance on structured tasks, underperforms Logistic Regression by 3.41% for Bangla and ranks last among all models for English (84.85%), consistent with prior findings that gradient boosting struggles with sparse, high-dimensional bag-of-words representations [23].

B. BERT vs. Hybrid LSTM+BERT: Marginal Gains

One of the most theoretically significant findings of this study is the marginal performance gain of the proposed Hybrid LSTM+BERT architecture over standalone BERT. For Bangla,

the Hybrid achieves 92.97% accuracy compared to BERT's 92.95% ($\Delta = 0.02\%$, or 0.0002 in absolute terms). For English, the relationship reverses: BERT outperforms the Hybrid by 0.30% (94.50% vs. 94.20%), suggesting that the additional recurrent layers over BERT's contextual representations provide no benefit and introduce a modest regularization cost for English.

These findings suggest that BERT's multi-head self-attention mechanism already captures the sequential and morphological dependencies in both languages, leaving little residual sequential information for the BiLSTM layers to exploit. The marginally positive result for Bangla ($\Delta = 0.02\%$) may reflect the slightly richer sequential modelling afforded by explicit recurrence over the morphologically denser Bangla token sequences, where local inflectional patterns may benefit from the inductive bias of recurrent processing. However, given the negligible magnitude of this gain, we cannot definitively attribute it to architectural superiority over statistical noise without further ablation studies. The negative result for English reinforces the interpretation that the Hybrid's additional parameters introduce unnecessary complexity for the simpler 4-class English task.

These findings are consistent with the broader literature on hybrid BERT–BiLSTM architectures for sentiment and service-request classification [34], [35], which generally report modest, task-dependent gains over standalone BERT. Our results extend this observation to a low-resource, morphologically rich language setting and provide useful negative evidence that the stacked-recurrence design does not meaningfully improve classification accuracy at this scale—steering future work toward alternative fusion mechanisms (e.g., cross-attention) rather than stacking.

C. Morphological Complexity as an Explanatory Factor

The morphological complexity analysis provides a principled, data-driven explanation for several observed performance patterns. The finding that 98.8% of Bangla articles exhibit medium-to-high morphological complexity ($\mu = 0.2905$, $\sigma = 0.0594$) confirms that inflectional richness is not an isolated characteristic of particular categories but a pervasive property of Bangla news text.

The stratified accuracy analysis (Fig. 8) reveals that the performance gap between SVM and deep learning models *widens* as morphological complexity increases. At low complexity, SVM achieves approximately 86% accuracy, only modestly below BiLSTM (≈90%) and BERT (≈92%). At high complexity, the gap grows: SVM reaches approximately 93% while BiLSTM and BERT achieve approximately 95–96%. This pattern is consistent with the hypothesis that bag-of-words representations suffer disproportionately from morphological diversity, as inflected surface forms generate distinct vocabulary entries for semantically equivalent tokens, fragmenting the feature space. In contrast, BERT's subword tokenization via WordPiece partially decomposes inflected tokens into shared subword units, and its self-attention mechanism captures contextual relationships between inflected forms that sparse TF-IDF vectors cannot.

The category-level variation in complexity is also noteworthy. Technology ($\bar{\mathcal{C}} = 0.3289$) and Economy ($\bar{\mathcal{C}} = 0.3284$) are the most morphologically complex categories, which may contribute to the higher classification difficulty observed in the confusion matrix analysis. Education ($\bar{\mathcal{C}} = 0.2104$), as the least complex category, likely benefits from a more constrained and consistent vocabulary, making it easier to classify even with sparse representations.

D. Comparison with Recent State-of-the-Art Studies

Table X positions this work against recent, closely related studies. Direct numeric comparison must be interpreted with care because tasks, taxonomies, and class counts differ: binary tasks such as hyperpartisan detection [14] are intrinsically easier than the 15-class, heavily imbalanced problem addressed here, and headline-only settings [15] carry far less signal than full articles. Within those caveats, three observations follow. First, the present study is, to our knowledge, the largest unified Bangla–English news-classification benchmark: it evaluates more model families than [9], [36] on a larger native Bangla corpus than [9], and—unlike [40]—uses native rather than translated English data. Second, our BERT results (92.95% on 15 imbalanced classes) are competitive with transformer results reported on easier task formulations, e.g., 95.65% binary accuracy in [14], while additionally quantifying the cost–benefit of hybridization that most prior studies leave unexamined. Third, no prior study in Table X connects morphology to model performance quantitatively; the complexity-stratified analysis is unique to this work and explains mechanistically why transformers dominate precisely where classical models weaken.

E. Why the Proposed Approach Performs Well, and Its Strengths

Why performance is strong. Three factors drive the strong headline results. First, *monolingual pretraining*: BanglaBERT's language-specific vocabulary and pretraining avoid the subword over-fragmentation that multilingual models exhibit on Bangla, while `bert-base-uncased` benefits from massive English pretraining. Second, *subword tokenization*: WordPiece decomposes inflected forms into shared units, directly countering the surface-form sparsity that the morphological analysis shows to be pervasive—explaining why the transformer advantage grows exactly where classical models degrade. Third, *scale with clean supervision*: 347,650 Bangla training documents with stratified splitting allow both fine-tuned transformers and from-scratch BiLSTMs to exploit the data fully, which also explains the counterintuitive Bangla-over-English BiLSTM result.

Strengths. The study's principal strengths are its controlled unified pipeline (identical splits, seeds, and evaluation across eight models and two languages), its use of native corpora in both languages, its breadth (three feature representations $\times$ five classical models plus three deep architectures), its full reproducibility protocol (fixed seeds across all libraries, checkpointing, documented software stack), and its explanatory morphological analysis, which converts an often hand-waved linguistic claim into a measurable, falsifiable relationship.

TABLE X
COMPARISON WITH RECENT STATE-OF-THE-ART STUDIES. REPORTED RESULTS ARE AS STATED BY THE ORIGINAL AUTHORS; TASKS AND CLASS STRUCTURES DIFFER, SO VALUES ARE NOT DIRECTLY COMMENSURABLE.

Study (year)	Language(s)	Corpus / task	Models compared	Best reported result	Key difference vs. this work
Alam *et al.* [37] (2020)	Bangla	Six benchmarks, multi-domain	Multilingual transformers	SOTA on six benchmarks	Transformers only; no classical baselines; no English counterpart
Ahmad *et al.* [36] (2022)	Bangla	Potrika: 664,880 articles, 8 categories	ML + LSTM/BiLSTM/GRU/CNN	Strong DL results on balanced subset	No transformer/hybrid; single language
Dhar *et al.* [10] (2023)	Bangla	News categorization	Hybrid classical–neural	Improved over individual baselines	Smaller scale; no cross-language evaluation
Das *et al.* [40] (2023)	Bangla + English	E-commerce review sentiment	ML, DL, hybrid	Hybrid DL best on sentiment	Translated English data; different domain; smaller corpora
Hossain *et al.* [9] (2025)	Bangla	118,404 news articles	ML, DL, hybrid stacking	Stacking meta-classifier best	Single language; $\sim$3.7$\times$ smaller corpus; no morphology analysis
Naeen *et al.* [15] (2025)	Bangla	News-title classification (6,150 titles)	Transformers + ML/DL + LIME	Transformer best with explainability	Headline-only; small corpus; explainability focus
Hasan *et al.* [14] (2025)	Bangla	Hyperpartisan detection (binary)	BanglaBERT + classical ML	95.65% accuracy / 95.44% F1	Binary task; semi-supervised; not category classification
This work	Bangla + English (native)	434,563 + 127,600 articles; 15/4 classes	5 classical + BiLSTM + BERT + Hybrid; 3 feature sets	BERT 92.95% (Bn) / 94.50% (En); SVM best classical	Unified pipeline; largest native bilingual benchmark; morphological complexity analysis

F. Practical Implications

The findings of this study offer several evidence-based guidelines for practitioners working on multilingual or low-resource text classification:

- **For resource-constrained deployments:** SVM with TF-IDF (bigram, sublinear scaling) provides a strong, computationally efficient baseline that achieves over 90% accuracy for both Bangla and English without requiring GPU hardware, pretrained models, or fine-tuning infrastructure.
- **For maximum accuracy:** BERT fine-tuning with a monolingual pretrained model is recommended. For Bangla, `csebuetnlp/banglabert` provides strong performance (92.95%) with relatively fast fine-tuning (15 minutes on an H100 GPU). The marginal gain of the Hybrid model ($\Delta = 0.02\%$) does not justify the additional computational cost (32 minutes vs. 15 minutes), making standalone BERT the preferred choice in practice.
- **For morphologically rich languages:** Transformer-based models with subword tokenization are strongly preferred over classical bag-of-words approaches, as the performance advantage of deep learning widens with increasing morphological complexity. BiLSTM provides a useful intermediate option when full BERT fine-tuning is not feasible, outperforming SVM by 1.51% for Bangla at a fraction of BERT's parameter count.
- **For class-imbalanced datasets:** Weighted F1-score should be the primary evaluation metric rather than accuracy, as the dominant *bangladesh* category (53.5% of Bangla test samples) artificially inflates accuracy for all models. Practitioners should further consider category-specific performance for deployment decisions, as minority categories such as *we-are* (F1: 33.64%) and *pachmisheli* (F1: 48.68%) remain challenging even for BERT.

G. Limitations

This study has several limitations that should be considered when interpreting the results. First, the morphological complexity metric is based on a curated list of 17 inflectional suffixes and does not capture the full range of Bangla morphological processes, including derivational morphology, sandhi rules, or compound word formation. A more comprehensive morphological analyser would provide a richer characterization of linguistic complexity.

Second, the study uses a single Bangla news source (Prothom Alo), which may introduce domain and editorial bias. The 15 categories reflect the publication's section structure, not a universal Bangla news taxonomy, and the severe class imbalance (the *bangladesh* category dominates 53.5% of the corpus) limits the generalizability of per-category findings to other Bangla corpora.

Third, the English AG News corpus is relatively small (127,600 articles) and limited to four broad categories, making cross-language comparisons sensitive to task difficulty asymmetry. A larger, more granular English benchmark with a comparable category structure would enable a more controlled cross-language evaluation.

Fourth, the Hybrid model was evaluated only with the specific architecture described (two-layer BiLSTM over BERT sequence output, top-4 BERT layers trainable). Alternative

hybrid designs—such as attention pooling over LSTM outputs, different BERT layer freezing strategies, or cross-attention between BERT and LSTM representations—may yield different performance profiles and are left for future investigation.

Fifth, results are reported from single seeded runs without multi-seed variance estimates or statistical significance testing, so small inter-model differences (such as the 0.02% Hybrid–BERT delta) cannot be distinguished from run-to-run noise. Finally, no comparison against multilingual or instruction-tuned large language models in zero- or few-shot settings is included; both are natural extensions of this benchmark.

VII. Conclusion

This paper presented a large-scale, unified comparative study of classical machine learning and deep learning architectures for news text classification in Bangla and English, benchmarking eight models over more than half a million native news articles under identical experimental conditions, together with a novel quantitative morphological complexity analysis.

Four principal conclusions emerge. First, transformer-based models consistently outperform classical approaches in both languages, with the advantage larger in the resource-rich English setting and partially narrowed in Bangla by TF-IDF's ability to capture inflectional surface forms. Second, SVM with TF-IDF is the strongest and most language-robust classical baseline, remaining a practical choice for resource-constrained deployments. Third, the hybrid LSTM+BERT provides no meaningful benefit over standalone BERT despite substantially higher training cost, indicating that self-attention already captures the sequential structure relevant to this task. Fourth, morphological complexity is pervasive in Bangla news text, and the advantage of contextual deep models widens systematically with it—empirically grounding the case for subword tokenization and contextual representations in morphologically rich, low-resource languages.

Beyond establishing strong multilingual baselines, these findings offer directly actionable model-selection guidance for practitioners in newsrooms, content platforms, and public-sector systems operating across resource conditions. Future work includes extending the morphological metric to derivational processes and compounding, mitigating class imbalance for more reliable minority-class evaluation, generalizing the framework to other morphologically rich low-resource languages (e.g., Hindi, Urdu, Tamil, Amharic), incorporating multilingual and instruction-tuned large language models in zero- and few-shot settings, and exploring cross-attention-based hybrid designs.

Data Availability Statement

The datasets analyzed in this study are publicly available. The Bangla news corpus (Prothom Alo) is available from the Kaggle repository "Text Classification Based on the BD Newspaper" at https://www.kaggle.com/datasets/tahmidmir/text-classification-based-on-the-bd-newspaper [41]. The English AG News benchmark [42] is available through the Hugging Face Datasets library at https://huggingface.co/datasets/ag_news. The derived experimental artifacts (per-model results and training summaries) generated during this study are available from the corresponding author upon reasonable request.

References

[1] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. Eur. Conf. Mach. Learn. (ECML)*, 1998, pp. 137–142.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[3] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.

[4] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. NAACL-HLT*, 2016, pp. 1480–1489.

[5] F. Alam *et al.*, "A review of Bangla natural language processing tasks and the utility of transformer models," *arXiv preprint arXiv:2107.03844*, 2021.

[6] Wikipedia contributors, "Bengali language," https://en.wikipedia.org/wiki/Bengali_language, 2025, accessed: 2025-05-01.

[7] M. S. Islam *et al.*, "BNLP: A toolkit and resource collection for Bengali language processing," https://github.com/sagorbrur/bnlp, 2019, accessed: 2025-05-01.

[8] M. Rafi-Ur-Rashid, M. Mahbub, and M. A. Adnan, "Breaking the curse of class imbalance: Bangla text classification," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 2022.

[9] T. Hossain, A.-R. Islam, M. H. K. Mehedi, A. A. Rasel, M. Abdullah-AL-Wadud, and J. Uddin, "BanglaNewsClassifier: A machine learning approach for news classification in Bangla newspapers using hybrid stacking classifiers," *PLOS ONE*, 2025, bangla news classification on 118,404 articles with ML, DL, and hybrid stacking models.

[10] A. Dhar, H. Mukherjee, K. Roy, K. C. Santosh, and N. S. Dash, "Hybrid approach for text categorization: A case study with Bangla news article," *J. Inf. Sci.*, 2023.

[11] M. N. Hoque, U. Salma, M. J. Uddin, M. M. Ahamad, and S. Aktar, "Exploring transformer models in the sentiment analysis task for the under-resource Bengali language," *Nat. Lang. Process. J.*, 2024.

[12] M. S. Islam and K. M. Alam, "Sentiment analysis of Bangla language using a new comprehensive dataset BangDSA and the novel feature metric SkipBangla-BERT," *Nat. Lang. Process. J.*, 2023.

[13] S. Saha, K. N. Asib, and M. M. Hoque, "Retriv at BLP-2025 Task 1: A transformer ensemble and multi-task learning approach for Bangla hate speech identification," in *Proc. BLP Workshop*, 2025.

[14] M. M. Hasan, F. B. Hassan, M. A. Jubair, Z. Ahmed, S. Yeakin, and M. M. Billah, "Bangla BERT for hyperpartisan news detection: A semi-supervised and explainable AI approach," *arXiv preprint arXiv:2507.21242*, 2025.

[15] M. J. Naeen, S. K. Das, S. A. Jisan, S. A. Khushbu, N. C. Saha, and Ohidujjaman, "Explainable detection: A transformer-based language modeling approach for Bengali news title classification with comparative explainability analysis using ML and DL," *Front. Artif. Intell.*, vol. 8, p. 1537432, 2025.

[16] M. H. Kabir, F. Ahmad, M. A. M. Hasan, and J. Shin, "Gender recognition of Bangla names using deep learning approaches," *Applied Sciences*, vol. 13, no. 1, p. 522, 2023.

[17] M. H. Kabir, A. S. M. Miah, M. Hadiuzzaman, and J. Shin, "Combining state-of-the-art pre-trained deep learning models: A novel approach for Bangla sign language recognition using max voting ensemble," *Systems and Soft Computing*, vol. 7, p. 200230, 2025.

[18] A. A. Shiam, S. M. Redwan, M. H. Kabir, and J. Shin, "A neural attention-based encoder-decoder approach for English to Bangla translation," *Computer Science Journal of Moldova*, vol. 31, no. 1, pp. 70–85, 2023.

[19] D. R. Cox, "The regression analysis of binary sequences," *J. Roy. Statist. Soc. Ser. B*, vol. 20, no. 2, pp. 215–232, 1958.

[20] M. E. Maron, "Automatic indexing: An experimental inquiry," *J. ACM*, vol. 8, no. 3, pp. 404–417, 1961.

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[22] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794.

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

[25] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR, Workshop Track*, 2013.

[28] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.

[29] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, 2017.

[30] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proc. ACL*, 2020, pp. 8440–8451.

[31] H. Schwenk and V. Chaudhary, "LASER: Language-agnostic sentence representations," in *Proc. NAACL-HLT, Vol. 1*, 2019, pp. 90–96.

[32] N. Groenwold *et al.*, "Evaluating transformer-based multilingual text classification," in *Proc. ACL*, 2020.

[33] C. Wang and M. Banko, "Practical transformer-based multilingual text classification," in *Proc. NAACL: Industry Papers*, 2021.

[34] R. Cai, B. Qin, Y. Chen, L. Zhang, R. Yang, S. Chen, and W. Wang, "Sentiment analysis about investors and consumers in energy market based on BERT-BiLSTM," *IEEE Access*, vol. 8, pp. 171 408–171 415, 2020.

[35] Y. Xiong, G. Chen, and J. Cao, "Research on public service request text classification based on BERT-BiLSTM-CNN feature fusion," *Appl. Sci.*, vol. 14, no. 14, p. 6282, 2024.

[36] I. Ahmad, F. AlQurashi, and R. Mehmood, "Machine and deep learning methods with manual and automatic labelling for news classification in Bangla language," *arXiv preprint arXiv:2210.10903*, 2022, introduces the Potrika dataset of 664,880 articles in eight categories.

[37] T. Alam, A. M. Khan, and F. Alam, "Bangla text classification using transformers," *arXiv preprint arXiv:2011.04446*, 2020.

[38] A. Bhattacharjee, T. Hasan, W. U. Ahmad, K. Samin, M. S. Islam, A. Iqbal, M. S. Rahman, and R. Shahriyar, "BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla," *arXiv preprint arXiv:2101.00204*, 2021.

[39] A. A. Saad, U. N. Mahi, M. S. Salim, and S. I. Hossain, "Bangla news article dataset," *Data in Brief*, vol. 57, p. 110874, 2024.

[40] R. K. Das, M. Islam, M. M. Hasan, S. Razia, M. Hassan, and S. A. Khushbu, "Sentiment analysis in multilingual context: Comparative analysis of machine learning and hybrid deep learning models," *Heliyon*, 2023.

[41] T. Mir, "Text classification based on the BD newspaper," https://www.kaggle.com/datasets/tahmidmir/text-classification-based-on-the-bd-newspaper, 2023, accessed: 2025-05-01.

[42] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, aG News benchmark dataset.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.