A Unified BERT-CNN-BiLSTM Framework for Simultaneous Headline Classification and Sentiment Analysis of Bangla News

Mirza Raquib, Munazer Montasir Akash, Tawhid Ahmed, Saydul Akbar Murad, Farida Siddiqi Prity, Mohammad Amzad Hossain, Asif Pervez Polok, Nick Rahimi**

***Abstract*—In our daily lives, newspapers are an essential information source that impacts how the public talks about present-day issues. However, effectively navigating the vast amount of news content from different newspapers and online news portals can be challenging. Newspaper headlines with sentiment analysis tell us what the news is about (e.g., politics, sports) and how the news makes us feel (positive, negative, neutral). This helps us quickly understand the emotional tone of the news. This research presents a state-of-the-art approach to Bangla news headline classification combined with sentiment analysis applying Natural Language Processing (NLP) techniques, particularly the hybrid transfer learning model BERT-CNN-BiLSTM. We have explored a dataset called BAN-ABSA of 9014 news headlines, which is the first time that has been experimented with simultaneously in the headline and sentiment categorization in Bengali newspapers. Over this imbalanced dataset, we applied two experimental strategies: technique-1, where undersampling and oversampling are applied before splitting, and technique-2, where undersampling and oversampling are applied after splitting on the In technique-1 oversampling provided the strongest performance, both headline and sentiment, that is 78.57% and 73.43% respectively, while technique-2 delivered the highest result when trained directly on the original imbalanced dataset, both headline and sentiment, that is 81.37% and 64.46% respectively. The proposed model BERT-CNN-BiLSTM significantly outperforms all baseline models in classification tasks, and achieves new state-of-the-art results for Bangla news headline classification and sentiment analysis. These results demonstrate the importance of leveraging both the headline and sentiment datasets, and provide a strong baseline for Bangla text classification in low-resource.**

***Index Terms*—Bangla, Headline, Sentiment, Simultaneously, Hybrid, NLP.**

**Corresponding author.

Mirza Raquib: Department of CCE, International Islamic University Chittagong, Chattogram, Bangladesh; Department of ICE, Noakhali Science and Technology University, Noakhali, Bangladesh. Email: mirzaraquib@iiuc.ac.bd

Munazer Montasir Akash: Department of CSE, Banladesh University of Engineering and Technology, Dhaka, Bangladesh; Dept. of CSE, IIUC, Chattogram, Bangladesh. Email: 0423052104@grad.cse.buet.ac.bd

Tawhid Ahmed: Department of CSE, Banladesh University of Engineering and Technology, Dhaka, Bangladesh. Email: tawhidahmed98@gmail.com

Saydul Akbar Murad: School of CSE, University of Southern Mississippi, Hattiesburg, MS, USA. Email: saydulakbar.murad@usm.edu

Farida Siddiqi Prity: Department of CSE, Netrokona University, Netrokona, Bangladesh. Email: prity@neu.ac.bd

Mohammad Amzad Hossain: Department of ICE, NSTU, Noakhali, Bangladesh. Email: amzad@nstu.edu.bd

Asif Pervez Polok: mPower Social Enterprise. Email: asif-polok.research@gmail.com

Nick Rahimi: School of Computing Sciences and Computer Engineering, University of Southern Mississippi, Hattiesburg, MS, USA. Email: nick.rahimi@usm.edu

I. Introduction

The rapid growth of digital content and the internet has necessitated robust natural language processing (NLP) systems that can analyze and comprehend human language properly. For instance, a language like Bangla, which is one of the most spoken languages in the world, has remained mostly overlooked as compared to English and other well-resourced languages. Newspapers continue to be one of the most significant information sources and the headlines play a crucial role by providing a quick idea of news content. At such times, headlines often convey a mood that can impact how readers interpret and react to news. Thus, not only the category of a headline (e.g., politics, sports, religion or other), but also its sentiment (e.g., positive/negative/ neutral) is an essential factor to help readers in understanding news content. Motivated by the above scenario, an exploration of advanced models is desired for Bangla text classification, mainly targeting headline categorization and sentiment analysis.

Text categorization, one of the classic problems under NLP, is a type of supervised learning in which predefined categories are assigned to free text documents on the basis of some linguistic and contextual information. Classical machine learning (ML) models, including Support Vector Machines (SVM), Naïve Bayes, and Random Forest, have long been serving for classification, which is based on hand-designed features. Nonetheless, these techniques frequently do not fully represent the semantic complexity and sequential dynamics of natural language. More recently, with the advent of deep learning (DL), models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit networks have demonstrated state-of-the-art performance by learning increasingly higher representations of text in an end-to-end manner. However, even after these developments, Bangla is still in an under-resourced state in NLP, with some challenges like morphological complexity and scarcity of annotated data, along with the absence of domain-specific pre-trained models. The main challenge that is addressed in this paper is how to build a robust and precise classification model for Bangla news headlines and sentiment analysis where all these issues come into consideration.

Machine Learning and deep learning methods have been applied in contrary or in hybrid with other for better classification. Solutions based on Machine Learning (ML) [1] [2]

[3] [4] perform well with small data, yet they don't scale and present a poor semantic understanding. The DL-based solutions, such CNN, BiLSTM and GRU models [5] [6] [7] [8], achieved a higher accuracy degree but tended to suffer from the overfitting problem and low generalization performance on the lowresource-learning condition. Hybrid models that perform feature extraction in combination with sequential modeling like CNN-LSTM, CNN-BiGRU and BERT-based models [9] [10] [11] [12] have also raised the bar for performance. However, most previous works either cannot handle the class imbalance problem effectively or tend not to make full use of the contextual embeddings provided by transformer-based models. Moreover, existing work has addressed for the most part headline classification or sentiment analysis alone and with cursory attempts toward jointly considering both tasks within a single model. This gap motivates us to propose a more hybrid model that should collectively capture headline categorization along with sentiment analysis of the Bagla news headline.

In order to fill in the gaps, this paper introduces a hybrid deep learning approach that combines pretrained BERT embeddings with CNN and BiLSTM networks for Bangla text classification. The proposed work has utilized two types of data sources (Bangla news headlines, Bangla sentiment reviews). All the datasets are evaluated in three forms (imbalanced, undersampled and oversampled) to have a fair evaluation on different data distributions. To ensure stability, k-fold cross-validation is utilized and model evaluation is carried out extensively using multiple evaluation metrics such as Accuracy, Precision, Recall and F1-score. Furthermore, it is enhanced with Local Interpretable Model-agnostic Explanations (LIME) for interpretability of model predictions.

Major Contributions of This Work

- Unified Analysis: We have developed a fusion of BERT-CNN-BiLSTM for the first time on Bengali headline and sentiment analysis simultaneously.
- Robust Dataset Evaluation: We have experimented with a publicly available dataset that has not been explored in any previous work.
- We present two methods for balancing: Technique-1, which involves oversampling and undersampling the entire dataset, and Technique-2, which only applies balancing to the training split.
- Rigorous Validation: Using 5-fold cross-validation, the result can be measured to provide an accurate performance estimate over several dataset variants.
- Comprehensive Metrics and Visualization: There are Accuracy, Precision, Recall, and F1-score are calculated with XAI to visualize the performance.
- Model Interpretability: Explainable predictive model using LIME for transparent and trustful decision support.
- Advancing Bangla NLP: Creating a strong baseline for Bangla text classification and contributing to low-resource language processing research.

The rest of this article is organized as follows: the related works are summarized in Section II. In Section III, we describe the proposed methodology in details, which consists of the dataset collection, processing, and train validation test dataset, and we propose a neural network architecture. In Section IV, the recognition experimental results are presented with a detailed explanation of the evaluation criteria of the proposed model. Finally, our conclusion is given in Section V.

II. Literature Review

Headline classification and sentiment analysis of Bangla news have become more significant issues recently, with the growth of digital media platforms. Headline classification is more concerned with detecting the topic or category of a news article, while sentiment analysis identifies the underlying emotional tone in the text. A lot of research has been carried out on these tasks individually in recent years; however, the growing existence of Bangla news portals emphasizes the need for such combined analysis towards a better understanding. 'This kind of integration allows readers to get the thematic focus and the affective impact of a news headline at one stroke. In Bangla news headline datasets, the class imbalance is handled by using undersampling and oversampling methods with a hybrid BERT-CNN-BiLSTM model for better classification, which was not studied in previous work on existing datasets.

Some earlier works on automatic classification of Bengali news headlines highlighted the challenges involved in this area, primarily because of unavailability of datasets and good word embeddings. An early endeavour using an LSTM-based deep learning model was made on a self-collected dataset of 4,580 Bangla headlines and it turned out to be a useful baseline for Bangla headline classification [20]. Based on this understanding, the later studies attempted to study deep learning and hybrid methods as well. For example, by comparing CNN, BiLSTM, and a hybrid model of BiLSTM+SVM with classical ML models on TF-IDF and Word2Vec embeddings, the authors found that the novel approach outperforms others with an accuracy of 94% and F1 score 94% [21]. Another work also implemented various classical ML classifiers, i.e, SVM, Naïve Bayes, Logistic Regression and Random Forest, along with a neural network, achieving 90% accuracy as the best performing by the neural network among all [22]. GRU-based models have been tested in this scenario as well, and the authors show that GRU attained an accuracy of 84% (best among 4 ML and 3 DL models) across eight categories of headlines [23]. In addition, CNN-LSTM with GloVe-based vectorisation outperformed numerous other models, achieving an accuracy of 87% [24]. Similarly, more complex designs have been suggested; an example of such is NewsNet, a hybrid CNN + RNN (BiLSTM with parameterised GRU) model that achieved 94.57% accuracy while being able to maintain precision, recall and F1 above 94% [25]. Hybrid models like attention-enhanced CNN-BiLSTM, introduced recently, focus on capturing both local and sequential features and attain 84.04% accuracy on a 136K Bangla news articles dataset, where GRU was very close to the result with its score of 83.91% [26].

After focusing on headline classification, more previous works have also been done in the area of Bangla text sentiment classification from web-based comments using some traditional ML algorithms (LR, SGD, SVM, NB, Decision Tree,

RF) with a new proposed Model of LSTM. The author used Count Vectorizer and TF-IDF on 4000 Bangla comments(2000 positive and 2000 negative), which were done for evaluation. In comparison, they reported an accuracy of 84.25% with the LSTM model, which is higher than the results of most previous Bangla sentiment analyses [13]. Extending this direction, a hybrid deep learning model, namely CNN-LSTM model, was proposed for sentiment analysis by utilizing various forms of word embedding, including Word2Vec, GloVe, CBOW and an Embedding layer that concentrated on the identification of emotions in Bangla text [14]. In their dataset, there are 32,000 Bangla Facebook comments and 5640 of them are annotated with six emotions. This entailed only three classes due to class imbalance: happiness, sadness and anger. The best performance for the authors hybrid CNN-LSTM model with Word2Vec embedding was an accuracy of 90.49% and F1-score of 92.83% which outperformed all other deep learning (CNN, LSTM) and traditional ML (SVM, NB, KNN) models. In another work, TF-IDF was also utilized for text representation, followed by an overall classification using a machine learning algorithm on Bangla product reviews [15]. Comparative Results out of SVM, GaussianNB, and MultinomialNB, the accurate machine learning model was Gaussian NB with an accuracy of 88.23%, followed by SVM and Multinomial NB with the accuracies of 83.04% and 86.37%, respectively. Continuing the lexical sentiment scoring, the Bangla Text Sentiment Score (BTSC) was proposed, which was then incorporated with deep learning using Word2Vec representations [16]. They have tuned several DL models, such as CNN, RNN, Bi-LSTM, HAN-LSTM, D-CAPSNET-Bi-LSTM, and BERT-LSTM, based on a categorical aspect-based Bangla cricket dataset annotated with three-fold polarities (positive, negative, and neutral). The attention-based hybrid BERT-LSTM model achieved the highest accuracy of 84.18% compared to all other models. In Bangla, the work in multi-class sentiment analysis is also limited, where large majority of works have focused only for ternary (positive/negative/neutral) classification and even their achieved results are not satisfactory. A hybrid CNN-LSTM (CLSTM) was proposed, and when tested using a dataset of 42,036 Facebook comments containing text according to the four sentiment classes, it performed quite well against six ML and four DL models [17]. Out of all these models, their CLSTM hybrid model performed the best at 85.8% accuracy (0.86 F1 score) over baseline ML models. In another contribution, the author examined sentiment in Bangladeshi front-page newspaper articles, creating a custom dataset over a period of six months and from five major newspapers [18]. The study used six ML classifiers, i.e., Logistic Regression, Multinomial NB, SVM, SVC, Random Forest and K-NN and found an accuracy (between 58.78%: K-NN; 85.19%: Logistic Regression) for three-class sentiment classification. A recent study, however, highlighted the issue of media bias and sentiment in relation to transparency in journalism in Bangladesh. Research comparing traditional ML models (SVM, LSTM) with transformer-based NLP models (one recent standard: BERT [19] The Transformer-based model beat the ML baselines by a wide margin with an accuracy of 93%(F1 – 92%).

While past studies have made contributions to Bangla headline and sentiment classification, they also have some limitations. Most previous work done in headline classification is based on relatively small or domain-specific datasets, which have an imbalanced class distribution. The underlying representations in these studies rely on static embeddings (TF-IDF, Word2Vec, GloVe) obtained from restricted knowledge graphs and may lead to dense clusters in the embedding space that compromise the semantic richness, such as transformer architectures with contextual embeddings via attention mechanisms. Furthermore, most of the approaches either focused on headline category or sentiment classification individually, without attempting a joint headline category and sentiment dimension analysis. Likewise, in sentiment analysis, previous works were either limited to binary or few-class classification, restricted to small or imbalanced datasets, did not have enough resources to preprocess abbreviations (e.g., comprehensive stopword/abbreviation lists or lemmatisation tools) and presented common overfitting in hybrid approaches. Additionally, the existing works on sentiment evaluation have been dependent on either shallow ML models or single DL frameworks, which have limited their power to cover the diverse complexity of Bangla text.

In response to these limitations, we have proposed a novel hybrid deep learning framework, BERT-CNN-BiLSTM, for Bangla news headline classification and sentiment analysis. Our model fuses transformer-based contextual embedding (BERT), convolutional feature extraction (CNN) and sequential dependency modelling (BiLSTM) to combine local and global semantics in a transformer-based approach for headline classification. The same architecture uses BERT contextual representation combined with CNN for local semantics and BiLSTM for long-term dependencies to provide excellent results in sentiment classification. In these studies, the publicly available BAN-ABSA imbalanced dataset that has not been explored so far in this context is used. Class distribution is balanced using undersampling and oversampling methods for bias reduction and better model performance. In this study, the hybrid model BERT-CNN-BiLSTM is used for headline classification and sentiment analysis that can handle both subject and sentiment simultaneously. Mask-Predicted: To test the robustness and universality of our model, we also conduct experiments on two other datasets, which include headline classification and sentiment analysis.

III. Methodology

In this framework, the local and global context information are effectively integrated for news headline classification and sentiment analysis. The imbalanced data is pre-processed in a more detailed way by cleansing, removing stop words, stemming, tokenization, padding, etc., and the words are encoded to represent the meaningful vectors. Such prepared inputs are then passed to a BERT model that can extract embeddings with rich context designed to extract semantic characteristics of the sentences. We then use a Convolutional Neural Network (CNN) to extract local, semantic patterns, and a Bidirectional LSTM (BiLSTM) layer to capture long-term

Study	Dataset					DL Models	Hybrid Models	Metrics						Limitations
	Headline	Sentiment	Other Dataset	Head. Classes	Sent. Classes			Acc.	Pre.	Rec.	F1	Cross Val.	XAI	
[20]	✓	✗	✗	5	✗	LSTM	✗	✓	✓	✓	✓	✗	✗	Small dataset, weak embeddings, no lemmatiser
[21]	✓	✗	✗	8	✗	CNN, BiLSTM	BiLSTM+SVM	✓	✓	✓	✓	✗	✗	Imbalanced dataset (economic vs tech)
[22]	✓	✗	✗	10	✗	SVM, NB, LR, RF, NN	✗	✓	✗	✗	✗	✗	✗	Small dataset, limited categories
[23]	✓	✗	✗	8	✗	GRU	✗	✓	✓	✓	✓	✗	✗	Moderate accuracy
[24]	✓	✗	✗	10	✗	LSTM, CNN, SVM, BiLSTM, ANN	CNN+LSTM	✓	✗	✗	✓	✗	✗	No transformer embeddings (static GloVe only)
[25]	✓	✗	✗	8	✗	BiLSTM, GRU, CNN	NewsNet (CNN+RNN)	✓	✓	✓	✓	✗	✗	Single dataset only
[26]	✓	✗	✗	6	✗	CNN, BiLSTM, GRU	CNN+BiLSTM+Attention	✓	✓	✓	✓	✗	✗	Limited to 6 categories, moderate accuracy
[13]	✗	✓	✗	✗	2	LSTM, LR, SGD, SVM, NB, DT, RF	✗	✓	✓	✓	✓	✗	✗	Binary only, small dataset, no lexicons
[14]	✗	✓	✗	✗	3	CNN, LSTM	BERT+LSTM	✓	✓	✓	✓	✗	✗	Only 3 emotions used (imbalance)
[15]	✗	✓	✗	✗	2	SVM, Gaussian NB, Multinomial NB	✗	✓	✓	✓	✓	✗	✗	Shallow ML only, binary only
[16]	✗	✓	✗	✗	3	CNN, LSTM (Word2Vec, CBOW, GloVe)	CNN+LSTM	✓	✓	✓	✓	✗	✗	Small, imbalanced dataset
[17]	✗	✓	✗	✗	4	CNN, LSTM, BiLSTM, GRU	CNN+LSTM (CLSTM)	✓	✓	✓	✓	✗	✗	Overfitting, imbalance
[18]	✗	✓	✗	✗	3	LR, SVM, NB, RF, KNN	✗	✓	✓	✓	✓	✗	✗	English only, no DL for Bangla
[19]	✗	✓	✗	✗	3	SVM, LSTM, BERT	✗	✓	✓	✓	✓	✗	✗	Focus on bias detection
Our Model	✓	✓	✓	4	3	CNN, BiLSTM	**BERT-CNN-BiLSTM**	✓	✓	✓	✓	✓	✓	The framework is restricted to two segments (i.e. headline classification and sentiment analysis), but extension to further ones (e.g. topic categorisation, emotion detection, sarcasm identification, aspect-based sentiment analysis) are needed for more generalisation.

Legend: Acc = Accuracy, Pre = Precision, Rec = Recall, F1 = F1-score, Cross Val. = Cross Validation, XAI = Explainable AI, ✓= Yes, ✗= No

TABLE I: Comparison of related studies with the proposed BERT-CNN-BiLSTM hybrid model.

Technique	Dataset Type	Aspect				Polarity		
		Other	Politics	Religion	Sports	Negative	Positive	Neutral
Technique 1	Imbalanced	3013	2663	1755	1583	4723	2622	1669
	Undersampling	1583	1583	1583	1583	1669	1669	1669
	Oversampling	3013	3013	3013	3013	4723	4723	4723
Technique 2	Train-Undersampling	1259	1259	1259	1259	1274	1274	1274
	Train-Oversampling	2231	2231	2231	2231	3677	3677	3677

TABLE II: Class distribution for Aspect and Polarity under Technique-1 and Technique-2.

dependencies over the sequence. The output from these layers is concatenated to form a featureset that is more powerful for classification.

The overall workflow of the proposed framework for headline classification and headline sentiment analysis is illustrated in Figure 1.

A. Dataset Collection

For this research, we utilized a Bangla news headline dataset BAN-ABSA from Kaggle [27]. It has three attributes: headline (post), category (aspect), and sentiment (polarity). Distribution of number of headlines per headline category (4 total) and sentiment label (3 total) — 9014 total.

Formally, if the dataset is denoted as

$$D = \{(x_i, y_i^{head}, y_i^{sent}) \mid i = 1, \ldots, N\}, \quad N = 9014$$

where x_i represents the headline text, $y_i^{head} \in \{1, 2, 3, 4\}$ represents the aspect label, and $y_i^{sent} \in \{1, 2, 3\}$represents the polarity label, then the distribution of samples per class is imbalanced as shown above.

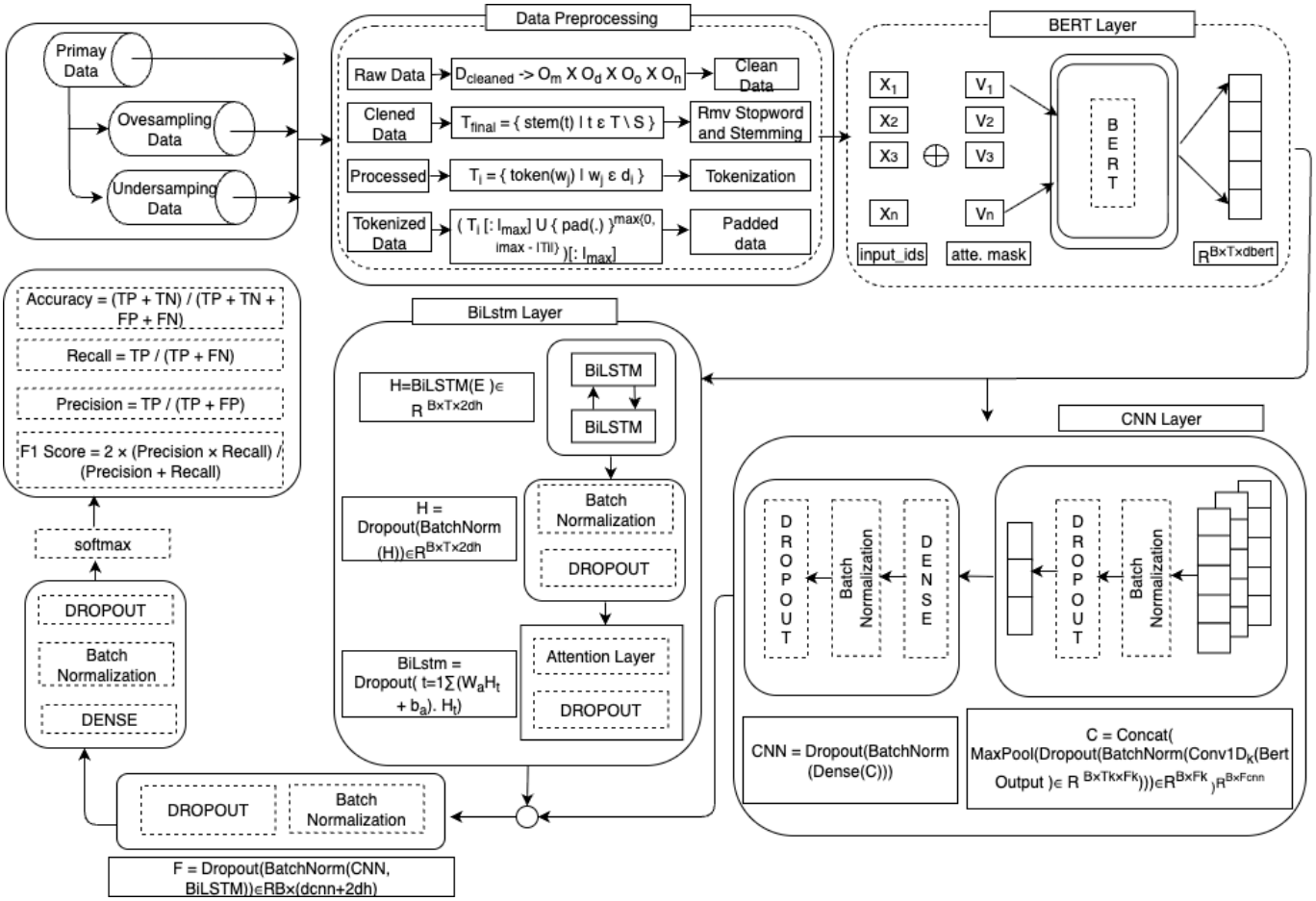


Fig. 1: Flow Chart of the headline classification with sentiment analysis

B. Undersampling and Oversampling

The class imbalance exists in both the headline category and sentiment label datasets. We utilized undersampling and oversampling techniques to generate balanced datasets for the credible training and testing. Aspect and polarity distributions of all imbalanced, under sampled and over sampled datasets both technique 1 and technique 2 are shown in Table II.

If $n_c$denotes the number of samples in class c, resampling transforms the distribution as:

$$n_c^{\text{under}} = \min(n_c)$$

$$D^{under} = \{(x_i, y_i^{head}, y_i^{sent}) \mid i = 1, \ldots, N\}, \quad N = n_c^{under}$$

$$n_c^{\text{over}} = \max(n_c)$$

$$D^{over} = \{(x_i, y_i^{head}, y_i^{sent}) \mid i = 1, \ldots, N\}, \quad N = n_c^{over}$$

This ensures all classes are equally represented in both undersampled and oversampled datasets.

C. Data Preprocessing

Data preprocessing includes several key steps: data cleaning, stopword removal and stemming, tokenization, padding, word encoding and sentence representation [35]. Each step transforms the textual input into a structure and a suitable format for the model. The following sections describe each step individually, including relevant mathematical equations.

- **Clean Data:** At the initial stage of imbalanced data contains different metadata, digits and some special symbols. Now removing this noise and irrelevant information from this imbalanced data to make it cleaned data ($D_cleaned$).

$$D_{cleaned} = D_m \times D_d \times D_o \times D_n$$

  Where D_m removes meta characters, O_d removes digits, D_o removes special characters, and D_n normalises text.
- **Stopword Removal and Stemming** The next step is the removal of stopwords, i.e, the high-frequency low-semantic words, after cleaning the imbalanced data. In addition, stemming reduces the words to their morphological root form, which further reduces the lexical redundancy. Additionally, to ensure that queries do not comprise of meaningless tokens, we eliminate sentences with a length of less than 2 words.

$$T_{final} = stem(t) \mid t \in T \setminus S$$

  Where T = token set, S = stopword list, and stem(t) = stem(t) applies stemming to token t.
- **Tokenization:** By dividing the cleaned and processed text into smaller linguistic units called tokens, the process of

tokenization converts them into numerical representations that deep learning models can process. BERT's tokenizer, which can handle sophisticated Bangla morphology and capture contextual nuances, was used for subword tokenization in this study.

$$T_i = token(w_j) \mid w_j \in d_i$$

where d_i is the ith headline and w_j are its words.

- **Padding:** By limiting the maximum length of all tokenized sequences to 300, padding maintains uniformity in the length of Bangla news headlines and postings, which may vary. A sequence is post-padded with zeros if it contains fewer than 300 tokens. It is reduced to the first 300 tokens if it is longer in order to keep all samples consistent. Formally, the padded version of a tokenized sequence T_i is given by:

$$P_i = Pad(T_i, maxlen)$$

where T_i is the tokenized input text, and

$$P_i = p_1, p_2, \ldots, p_{maxlen}$$

such that:

$$p_j = w_j for j <= k, and p_j = 0 for j > k$$

when the sequence length k ¡ 300. For sequences longer than 300,

$$P_i = w_1, w_2, \ldots, w_300$$

After tokenization and padding, the final dataset is represented as a fixed-length matrix:

$$D = P_1, P_2, \ldots, P_N, P_i \in R_300$$

where N is the total number of headlines.

D. Proposed Model Architecture Overview

The overall architecture of the proposed hybrid model is illustrated in Figure 2. The model begins with a BERT layer to generate contextualised embeddings from input sequences, which are then processed by two parallel branches: a CNN module that captures local n-gram features and a BiLSTM+Attention module that models long-range dependencies. The feature representations from both branches are fused and passed through fully connected layers for the final classification. In the following subsections, we describe each component of the model in detail, beginning with the BERT layer.

*1) **BERT Layer**:* In this phase, we have employed BERT (Bidirectional Encoder Representations from Transformers) as the foundational embedding layer for our hybrid classification model BERT-CNN-BiLSTM. BERT is a pre-trained transformer model that captures contextualized representations of words by considering both left and right contexts simultaneously. Unlike traditional word embeddings such as Word2Vec or GloVe, which produce a single static vector per word, BERT generates embeddings that vary according to the surrounding textual context. Then BERT is adopted in the architecture to produce context-dependent embedding for each token from a news headline. These embeddings are input into the CNN and BiLSTM models with attention to capture local patterns and global dependencies, leading to higher accuracy for both the headline category and sentiment label prediction. For our BERT layer, the input parameters are:

- Maximum sequence length (T): 300 tokens
- Hidden size (d_{bert}): 768
- Batch size (B): depends on training configuration
- Token IDs ($X \in \mathbb{R}^{B \times T}$)
- Attention masks ($M \in \{0,1\}^{B \times T}$) indicating valid tokens
- **Input Embeddings** Before passing the tokenized text to the BERT encoder, each token is transformed into a dense vector representation that combines three components: token embeddings, positional embeddings, and segment embeddings. For a given token at position $t \in 1, \ldots, T$ in sample b, the input embedding is defined as [37]:

$$h_{b,t}^{(0)} = e_{b,t}(tok) + e_t(pos) + e_{b,t}(seg) \in R^{d_{\text{bert}}}$$

Now, for sample b, the input representation is obtained by stacking all of the token embeddings for a sequence.

$$H_b^{(0)} = [h_{b,1}^{(0)}, h_{b,2}^{(0)}, \ldots, h_{b,T}^{(0)}] \in R^{T \times d_{\text{bert}}}$$

- **Transformer Encoder** The transformer encoder in BERT consists of L stacked layers, each combining multi-head self-attention and a position-wise feed-forward network, with residual connections and layer normalization. This enables the model to capture long-range dependencies and contextual information for each token.
At layer ℓ, the output of the encoder can be represented compactly as [38]:

$$H^{(\ell)} = \text{LayerNorm}\Big(\text{FFN}\big(\text{MHA}(H^{(\ell-1)}, M)\big) + H^{(\ell-1)}\Big) \in \mathbb{R}^{B \times T \times d_{\text{bert}}}$$

Where, $\text{MHA}(\cdot, M)$ is the masked multi-head self-attention, which computes contextualized token representations while ignoring padded positions indicated by M.FFN$(\cdot)$ is the position-wise feed-forward network, applied independently to each token embedding.
After L layers, the final contextual embeddings are:

$$H = H^{(L)} \in R^{B \times T \times d_{\text{bert}}}$$

These embeddings capture both local token-level patterns and global sequence-level dependencies, and can be used directly for downstream tasks such as classification.

- **Dropout Layer** To reduce overfitting and improve generalization, we apply a dropout layer immediately after the BERT embeddings. Specifically, a dropout rate of 0.35 is used:

$$H_{drop} = Dropout(0.35)(H)$$

Here, H is the BERT output $H \in R^{B \times T \times d_{\text{bert}}}$ and H_{drop} is passed to the CNN and BiLSTM modules for downstream feature extraction.

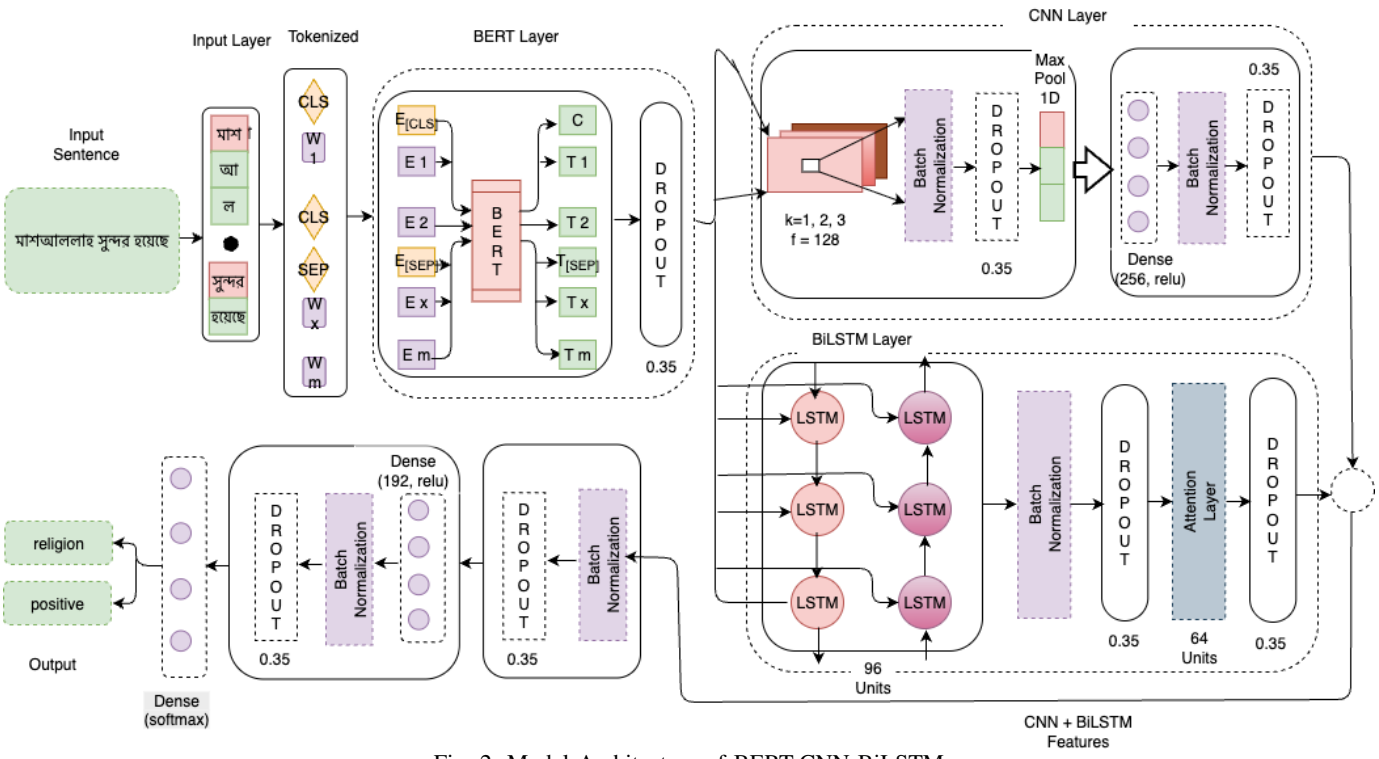


Fig. 2: Model Architecture of BERT-CNN-BiLSTM

*2) **CNN Layer**:* After obtaining contextual embeddings from BERT, it is essential to capture local patterns and n-gram features that may be indicative of specific aspects or sentiments in news headlines. While BERT embeddings encode rich semantic and contextual information, a CNN layer helps extract position-invariant local features from these embeddings, which are crucial for identifying short-range lexical patterns that can influence classification.

To achieve this, we employ a 1D convolutional branch with multiple kernel sizes. This allows the model to capture local patterns of different lengths across the token sequence $H_{drop} \in R^{B\times T\times d_{\text{bert}}}$.

- **Multi-kernel convolution** During convolution, the filter count is 128, ReLU activation is employed, L2 regularization is added, and batch normalization and dropout (p=0.35) follow [39]:

$$Z_{b,t,j}^{(k)} = \sigma\left(\sum_{i=1}^{d_{\text{bert}}}\sum_{m=1}^{k} W_{i,m,j}^{(k)} \cdot H_{\text{drop}}[b, t+m-1, i] + b_j^{(k)}\right)$$

where $W^{(k)} \in \mathbb{R}^{d_{\text{bert}}\times k\times 128}$ are convolutional weights, $b^{(k)} \in \mathbb{R}^{128}$ biases, and $\sigma(\cdot)$ denotes ReLU activation. Batch normalization and dropout($p = 0.35$) are then performed after convolution + activation:

$$H^{(k)} = \text{Dropout}\left(\text{BatchNorm}(Z^{(k)})\right).$$

- **Pooling and concatenation** Global MaxPooling is used to extract the most notable characteristics from each convolution. It pools across the sequence length while picking the maximum activations for each filter:

$$P_{b,j}^{(k)} = \max_t H_{b,t,j}^{(k)}, j \in \{1, \ldots, 128\}$$

Then each of the branches outputs a pooled feature vector:

$$P^{(k)} \in \mathbb{R}^{B\times 128}$$

These pooled features are then concatenated to form a unified feature vector:

$$C = \text{Concat}\left(P^{(2)}, P^{(3)}, P^{(4)}\right) \in \mathbb{R}^{B\times 384}.$$

- **Fully connected projection with regularization** This concatenated vector is then passed through the dense layer, followed by batch normalization and dropout. The mapping of C to a 256-D representation by the dense layer by the following:

$$U_b = \phi\left(C_b W_{\text{fc}} + b_{\text{fc}}\right), \quad W_{\text{fc}} \in \mathbb{R}^{384\times 256},\ b_{\text{fc}} \in \mathbb{R}^{256},$$

where $\phi(\cdot)$ represents batch normalization and ReLU, followed by dropout, $W^{(k)}$ is onvolutional weights, $b^{(k)}$ is convolutional biases, W_{fc} is dense projection weights and b_{fc} is biases.
Thus, the CNN representation is:

$$CNN = \text{Dropout}\left(\text{BatchNorm}(U_b)\right) \in \mathbb{R}^{B\times 256}.$$

This CNN feature vector is specific to local lexical patterns and is later concatenated with global features from BiLSTM into a single entity that is classified at the end. **Algorithm 1 (CNNBranch)** describes the CNN part of the hybrid BERT–CNN–BiLSTM model aimed at capturing local lexical features from BERT embeddings. Regularization: Learn dropouts, learn batch norm, L2, etc — all help prevent overfitting and improve generalization.

Algorithm 1 CNN Feature Extraction from BERT Embeddings

Require:
BERT embeddings: $H_{\text{drop}} \in \mathbb{R}^{B \times T \times d_{\text{bert}}}$
Kernel sizes: $K = \{2, 3, 4\}$
Number of filters per kernel: $F_k = 128$
Dropout probability: $p = 0.35$
Ensure:
CNN feature vector: CNN $\in \mathbb{R}^{B \times 256}$

for $k \in \{2, 3, 4\}$ **do**
  for $j = 1$ to F_k **do**
    for $b = 1$ to B **do**
      for $t = 1$ to $T - k + 1$ **do**
        $Z[b, t, j] \leftarrow \text{ReLU}\Big($
        $\sum_{i=1}^{d_{\text{bert}}} \sum_{m=1}^{k} W[k][i, m, j] \cdot H_{\text{drop}}[b, t + m - 1, i]$
        $+ b[k][j]\Big)$
      end for
    end for
  end for
  $H_k \leftarrow \text{Dropout}(\text{BatchNorm}(Z))$
  for $b = 1$ to B **do**
    for $j = 1$ to F_k **do**
      $P[k][b, j] \leftarrow \max_t H_k[b, t, j]$
    end for
  end for
end for

Concatenate pooled features from all kernels
for $b = 1$ to B **do**
  $C[b] \leftarrow \text{Concat}(P[2][b], P[3][b], P[4][b])$ ▷ Dimension $B \times 384$
end for

Fully connected projection with batch norm and dropout
for $b = 1$ to B **do**
  $U[b] \leftarrow C[b] \cdot W_{\text{fc}} + b_{\text{fc}}$ ▷ Dense layer projection
  $\text{CNN}[b] \leftarrow \text{Dropout}(\text{BatchNorm}(U[b]))$ ▷ Final CNN feature vector
end for
return CNN

*3) **BiLSTM Layer with Attention Layer**:* While the CNN branch captures local n-gram patterns, it is also important to learn global contextual dependencies across the entire sequence. To achieve this, we use a Bidirectional LSTM (BiLSTM) combined with an attention mechanism. The BiLSTM encodes sequential dependencies in both forward and backwards directions, while attention highlights the most informative tokens for the classification task.

- **BiLSTM layer input** Given BERT embeddings $H_{drop} \in R^{B \times T \times d_{bert}}$, this BiLSTM calculates the hidden states processing the sequence in both forward and backward directions. **Forward LSTM**
- **Forget Gate**:
$$\vec{f_t} = \sigma\big(W_f \vec{x}_t + U_f \vec{h}_{t-1} + b_f\big)$$
- **Input Gate**:
$$\vec{i_t} = \sigma\big(W_i \vec{x}_t + U_i \vec{h}_{t-1} + b_i\big)$$
- **Candidate Cell State**:
$$\tilde{\vec{c}}_t = \tanh\big(W_c \vec{x}_t + U_c \vec{h}_{t-1} + b_c\big)$$
- **Cell State Update**:
$$\vec{c}_t = \vec{f}_t \odot \vec{c}_{t-1} + \vec{i}_t \odot \tilde{\vec{c}}_t$$
- **Output Gate**:
$$\vec{o}_t = \sigma\big(W_o \vec{x}_t + U_o \vec{h}_{t-1} + b_o\big)$$
- **Hidden State**:
$$\vec{h}_t = \vec{o}_t \odot \tanh(\vec{c}_t)$$

Backward LSTM ($\overleftarrow{h_t}$)

The backward LSTM computes the same set of gates
$$\{\overleftarrow{f_t},\ \overleftarrow{i_t},\ \overleftarrow{\tilde{c}_t},\ \overleftarrow{c_t},\ \overleftarrow{o_t},\ \overleftarrow{h_t}\}$$
but processes the sequence in reverse, from $T \to 1$.
Now the BiLSTM processes the sequence in both forward and backward directions:

$$\overrightarrow{h_t} = \text{LSTM}_f(x_t, \overrightarrow{h_{t-1}}), \quad \overleftarrow{h_t} = \text{LSTM}_b(x_t, \overleftarrow{h_{t+1}})$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}], \quad h_t \in \mathbb{R}^{2d_h}$$

$$H = [h_1, h_2, \ldots, h_T], \quad H \in \mathbb{R}^{B \times T \times 2d_h}$$

To stabilise training and reduce overfitting, we apply batch normalization and dropout:

$$H' = \text{Dropout}(\text{BatchNorm}(H)), \quad H' \in \mathbb{R}^{B \times T \times 2d_h}$$

- **Attention mechanism** The attention layer computes an importance score for each hidden state H'_t, allowing the model to focus on the most relevant tokens:

$$u_t = \tanh(W_a h'_t + b_a), \quad u_t \in \mathbb{R}^{d_a}$$

$$\alpha_t = \frac{\exp(u_t^\top u_w)}{\sum_{k=1}^{T} \exp(u_k^\top u_w)}, \quad \alpha_t \in \mathbb{R}$$

The final attended representation is the weighted sum of hidden states with dropout:

$$\text{BiLSTM}_{\text{att}} = \sum_{t=1}^{T} \alpha_t h'_t, \quad \text{BiLSTM}_{\text{att}} \in \mathbb{R}^{B \times 2d_h}$$

Here, $W_a \in \mathbb{R}^{d_a \times 2d_h}$, $b_a \in \mathbb{R}^{d_a}$, and $u_w \in \mathbb{R}^{d_a}$ are trainable parameters of the attention layer.

BiLSTMBranch in **Algorithm 2** denotes the BiLSTM portion of the hybrid BERT–CNN–BiLSTM model, designed to capture global sequential dependencies and the attention mechanism in this layer is used to focus and draw global context information from BERT embeddings.

Algorithm 2 BiLSTM + Attention Feature Extraction

Require:
BERT embeddings: $H_{\text{drop}} \in \mathbb{R}^{B\times T\times d_{\text{bert}}}$
Hidden units: $d_h = 128$
Attention dimension: d_a
Ensure:
BiLSTM attention vector: $\text{BiLSTM}_{\text{att}} \in \mathbb{R}^{B\times 2d_h}$
for $b = 1$ to B **do**
  for $t = 1$ to T **do**
    Forward LSTM:
    $f_t^{\rightarrow} = \sigma(W_f^{\rightarrow} x_t + U_f^{\rightarrow} h_{t-1}^{\rightarrow} + b_f^{\rightarrow})$
    $i_t^{\rightarrow} = \sigma(W_i^{\rightarrow} x_t + U_i^{\rightarrow} h_{t-1}^{\rightarrow} + b_i^{\rightarrow})$
    $\tilde{c}_t^{\rightarrow} = \tanh(W_c^{\rightarrow} x_t + U_c^{\rightarrow} h_{t-1}^{\rightarrow} + b_c^{\rightarrow})$
    $c_t^{\rightarrow} = f_t^{\rightarrow} \odot c_{t-1}^{\rightarrow} + i_t^{\rightarrow} \odot \tilde{c}_t^{\rightarrow}$
    $o_t^{\rightarrow} = \sigma(W_o^{\rightarrow} x_t + U_o^{\rightarrow} h_{t-1}^{\rightarrow} + b_o^{\rightarrow})$
    $h_t^{\rightarrow} = o_t^{\rightarrow} \odot \tanh(c_t^{\rightarrow})$
    Backward LSTM:
    $f_t^{\leftarrow} = \sigma(W_f^{\leftarrow} x_t + U_f^{\leftarrow} h_{t+1}^{\leftarrow} + b_f^{\leftarrow})$
    $i_t^{\leftarrow} = \sigma(W_i^{\leftarrow} x_t + U_i^{\leftarrow} h_{t+1}^{\leftarrow} + b_i^{\leftarrow})$
    $\tilde{c}_t^{\leftarrow} = \tanh(W_c^{\leftarrow} x_t + U_c^{\leftarrow} h_{t+1}^{\leftarrow} + b_c^{\leftarrow})$
    $c_t^{\leftarrow} = f_t^{\leftarrow} \odot c_{t+1}^{\leftarrow} + i_t^{\leftarrow} \odot \tilde{c}_t^{\leftarrow}$
    $o_t^{\leftarrow} = \sigma(W_o^{\leftarrow} x_t + U_o^{\leftarrow} h_{t+1}^{\leftarrow} + b_o^{\leftarrow})$
    $h_t^{\leftarrow} = o_t^{\leftarrow} \odot \tanh(c_t^{\leftarrow})$
    $h_t = \text{Concat}(h_t^{\rightarrow}, h_t^{\leftarrow})$
  end for
  $H_b' = \text{Dropout}(\text{BatchNorm}([h_1, h_2, \ldots, h_T]))$
  Attention mechanism:
  for $t = 1$ to T **do**
    $u_t = \tanh(W_a \cdot H_b'[t] + b_a)$
    $e_t = u_w^{\top} \cdot u_t$
  end for
  $\alpha_t = \text{softmax}([e_1, e_2, \ldots, e_T])$
  $\text{BiLSTM}_{\text{att}}[b] = \sum_{t=1}^{T} \alpha_t \cdot H_b'[t]$
end for
$\text{BiLSTM}_{\text{att}} = \text{Dropout}(\text{BiLSTM}_{\text{att}})$
return $\text{BiLSTM}_{\text{att}}$

*4) **Feature Combination and Classification Layer**:* The strengths of both branches are complementary in that the local features in CNN cooperate with the global contextual features in BiLSTM+Attention. This fusion enables the model to capture both local lexical patterns and long-range dependencies in a single, unified representation.

- **Feature Combination** The CNN feature vector ($\in R^{Bd_{cnn}}$) and BiLSTM-attention feature vector ($\in R^{B2d_h}$) are concatenated:

$$F = \text{Dropout}\Big(\text{BatchNorm}\big(\text{Concat}(\text{CNN}, \text{BiLSTM}_{att})\big)\Big) \in \mathbb{R}^{B\times(d_{\text{cnn}}+2d_h)}$$

Batch normalization stabilizes training, while dropout (p=0.35) improves generalization.

- **Fully connected layers** The combined feature vector is then passed through a dense layer with 192 units, followed by batch normalization and dropout for additional regularization:

$$x = Dropout(BatchNorm(Dense_{192}(F)))$$

- **Output layer** Finally, classification is performed using a softmax layer over C classes, where C corresponds to the number of target labels (e.g., 4 for aspect classification or 3 for polarity classification) [36]:

$$\hat{y} = \text{Softmax}(W_o x + b_o) \in \mathbb{R}^{B\times C}$$

Where for each class c ∈ 1,…,C:

$$\hat{y}_{b,c} = \frac{\exp\big((W_o x_b + b_o)_c\big)}{\sum_{j=1}^{C} \exp\big((W_o x_b + b_o)_j\big)}, \quad \forall c \in \{1, \ldots, C\}$$

- **Optimization and loss function** The model is trained with the AdamW optimizer (learning rate lr, weight decay = 0.01, gradient clipping at 1.0) to improve stability and generalization. To further reduce overconfidence in predictions, categorical cross-entropy loss with label smoothing (∈=0.2) is applied:

$$L = -\frac{1}{B}\sum_{b=1}^{B}\sum_{c=1}^{C}\Big[(1-\epsilon)y_{b,c} + \frac{\epsilon}{C}\Big]\log \hat{y}_{b,c}$$

With label smoothing (ϵ), the target distribution $y_{b,c}^{smooth}$ is defined as:

$$y_{b,c}^{smooth} = (1-\epsilon)\cdot y_{b,c} + \frac{\epsilon}{C}$$

This HybridClassifier is responsible for taking the outputs of both CNNBranch and BiLSTMBranch and performing the final classification. First, the input sentence is encoded using a BERT tokenizer and encoder to get contextual embeddings for the words. Local lexical patterns are modeled using the embeddings with **Algorithm 1** (CNNBranch), while **Algorithm 2**(BiLSTMBranch) models global sequence-level dependencies and uses attention to emphasize salient contextual features. After obtaining the feature vectors from the two branches, they are fused together, normalized, and regularized and then sent to fully connected layers for classification. The complete flow of this hybridization process is illustrated in **Algorithm 3** (HybridClassifier), where the final prediction is made by integrating both CNN and BiLSTM branches, along with overfitting reduction via dropout and batch normalization.

E. Performance Analysis

We evaluated the performance of the models using multiple metrics, including loss, accuracy, precision, recall, and f1-score. Moreover, the model performance was further examined and analyzed through train-test-validation accuracy and loss curves, and XAI-LIME for visualization. The class imbalance was resolved by undersampling and oversampling methods in

Algorithm 3 HybridClassifier: BERT–CNN–BiLSTM with Attention

Require: Sentence S, tokenizer $TokBERT$, BERT encoder, max sequence length L_{max}, optional label y
Ensure: Prediction $\hat{y}$ or Loss L

tokens, mask $\leftarrow TokBERT(S)$
pad/truncate tokens to L_{max}
$E \leftarrow \text{BERT}(tokens, mask)$
$E' \leftarrow \text{Dropout}_{0.35}(E)$
$C_{feat} \leftarrow \text{CNNBranch}(E')$
$A \leftarrow \text{BiLSTMBranch}(E')$
$F \leftarrow \text{Concat}(C_{feat}, A)$
$F \leftarrow \text{BatchNorm}(F)$
$F \leftarrow \text{Dropout}_{0.35}(F)$
$F' \leftarrow \text{ReLU}(\text{Dense}_{192}(F))$
$F' \leftarrow \text{BatchNorm}(F')$
$F' \leftarrow \text{Dropout}_{0.35}(F')$
for each class $c \in \{1, \ldots, C\}$ **do**
    $z_{b,c} \leftarrow (W_o F'_b + b_o)_c$
end for
for each class $c \in \{1, \ldots, C\}$ **do**
    $\hat{y}_{b,c} \leftarrow \frac{\exp(z_{b,c})}{\sum_{j=1}^{C} \exp(z_{b,j})}$
end for
if training **then**
    for each class $c \in \{1, \ldots, C\}$ **do**
        $y_{b,c}^{smooth} \leftarrow (1 - \epsilon) \cdot y_{b,c} + \frac{\epsilon}{C}$
    end for
    $L \leftarrow -\frac{1}{B} \sum_{b=1}^{B} \sum_{c=1}^{C} y_{b,c}^{smooth} \cdot \log(\hat{y}_{b,c})$
    return L
else
    return $\hat{y}$
end if

both headline category and sentiment-based datasets, and the performance of balanced datasets was verified again using k-fold cross-validation. The results of the analysis are compared thoroughly, as discussed in the next section. It is worth noting that the BAN-ABSA dataset has not been used before; there is no existing prior implementation code available online. Therefore, we used this dataset for the first time, and the architecture developed based on extensive experimentation and insights available from the previous studies.

IV. Result Analysis

This section presents the experimental results for Bangla news headline classification and sentiment analysis using two different methods to address class imbalance. In Technique 2, the initial imbalanced dataset was first divided, and oversampling and undersampling were applied only to the training portion, leaving the validation and test sets to ensure fair evaluation. In Technique 1, oversampling and undersampling were applied directly to the entire dataset, and train–validation–test sets were then generated. To find out how resampling strategies affect model performance, both methods were tested on all dataset variants, including the original imbalanced set, the oversampled version, and the undersampled version. A thorough comparison across learning paradigms was made possible by the evaluation of several model categories, including Transformer-based, deep learning-based, and hybrid architectures. Usual criteria, such as F1-score, recall, accuracy, and precision, were used in the evaluation.

Table III presents the details results of headline and sentiment classification both undersampled and oversampled before spliting. Table IV presents the detailed results of k-fold cross-validation for headline and sentiment classification, which show the highest accuracy of technique 1.

A. Technique 1 - Headline Classification

- **Headline – Full Undersampled Dataset** Full undersampled configuration separated the headline dataset into train, validation, and test partitions after directly applying undersampling to the entire dataset. With a precision of 79.97%, recall of 77.22%, and F1-score of 77.86%, the BERT-CNN-BiLSTM model demonstrated 82.74% train accuracy, 78.16% validation accuracy, and 77.22% test accuracy. According to these findings, undersampling reduces the dataset size significantly and eliminates informative samples that are crucial for capturing semantic richness in Bangla news headlines, even though it lessens the impact of majority classes. As a result, the model's generalization ability is somewhat limited, as evidenced by the lower test metrics. Aggressive instance removal, according to the general trend, reduces contextual variety and impairs the model's capacity to capture sophisticated class patterns.
- **Headline – Full Oversampled Dataset** On the other hand, there was a noticeable improvement in performance when oversampling the entire dataset before the train–validation–test split. As evidenced by 82.21% precision, 78.57% recall, and an F1-score of 79.39%, the BERT-CNN-BiLSTM model trained on the oversampled variant obtained 84.12% train accuracy, 84.60% validation accuracy, and 78.57% test accuracy. The model was able to learn balanced decision boundaries across categories because of oversampling, which enhanced the minority classes with more representative samples. Because the model no longer biased against majority classes, it became more robust and equitable. The significant improvement over the undersampled version shows that oversampling data is a better way to preserve semantic diversity in headline classification.
- **Headline – K-fold on Full Oversampled Dataset** The oversampled dataset was evaluated using a 5-fold cross-validation setup in order to further evaluate robustness. With F1-scores ranging from 78.07% to 81.84%, the test accuracy was consistently high across all five folds, ranging from 75.77% to 81.04%. Additionally, precision values (84.57%–86.57%) remained consistent, indicating robust class-balance learning. The most prominent folds, FOLD-2 and FOLD-5, demonstrated dependable generalization across various data splits with test accuracy of 81.04% and 80.95%, respectively. The consistency between folds shows that oversampling preserves robustness when exposed to different subsets of the data in

Dataset	Train (%)	Val (%)	Test (%)	Precision (%)	Recall (%)	F1 (%)
Headline – Full OvS Dataset	**84.12**	**84.60**	**78.57**	**82.21**	**78.57**	**79.39**
Headline – Full UnS Dataset	82.74	78.16	77.22	79.97	77.22	77.86
Sentiment – Full OvS Dataset	**82.38**	**74.07**	**73.43**	**76.43**	**73.43**	**73.71**
Sentiment – Full UnS Dataset	78.58	65.42	66.94	72.44	66.94	66.72

TABLE III: Performance comparison of BERT–CNN–BiLSTM model using Technique 1

Fold	Headline – Full OvS Data						Sentiment – Full OvS Data					
	Train	Val	Test	P	R	F1	Train	Val	Test	P	R	F1
1	86.72	81.65	79.34	85.42	79.34	80.38	87.60	76.54	75.18	77.14	75.18	75.12
2	89.08	82.61	81.04	85.70	81.04	81.81	81.01	72.25	68.69	72.69	68.69	68.75
3	86.25	80.11	76.62	85.98	76.62	78.07	83.81	75.66	72.70	74.37	72.70	72.85
4	81.33	77.29	75.77	84.57	77.29	78.57	85.12	75.66	74.38	76.12	74.38	74.57
5	90.08	83.46	80.95	86.57	80.95	81.84	87.00	76.62	77.37	77.76	77.37	77.43
Avg	**86.69**	**80.62**	**78.34**	**85.65**	**78.34**	**80.13**	**84.31**	**75.75**	**73.66**	**75.62**	**73.66**	**73.75**

TABLE IV: K-fold cross-validation results of BERT-CNN-BiLSTM model for Headline and Sentiment Full OvS Data using Technique 1

Model	Headline Dataset						Sentiment Dataset					
	Train	Val	Test	P	R	F1	Train	Val	Test	P	R	F1
CNN	73.76	70.13	71.94	73.00	72.00	67.00	78.22	64.87	63.54	56.00	64.00	58.00
LSTM	58.20	55.83	57.14	44.00	57.00	48.00	59.91	55.59	55.26	71.00	55.00	48.00
BiLSTM	71.52	70.00	69.64	60.00	70.00	63.00	32.71	33.67	33.72	11.00	34.00	17.00
GRU	84.31	73.88	75.30	75.00	75.00	74.00	60.68	55.45	53.94	38.00	54.00	43.50
BiGRU	57.41	53.94	53.66	54.00	54.00	46.00	54.70	49.52	49.05	32.00	49.00	38.79
BERT–CNN–BiLSTM	**84.12**	**84.60**	**78.57**	**82.21**	**78.57**	**79.39**	**82.38**	**74.07**	**73.43**	**76.43**	**73.43**	**73.71**

TABLE V: Model comparison on Headline and Sentiment datasets using Technique 1

addition to improving performance on a single train–test configuration. This demonstrates how well oversampling produces a structurally balanced dataset that minimizes variances in performance while maintaining contextual richness.

This section assesses Technique-1 on headline, which divides the entire headline dataset into train, validation, and test sets after applying undersampling and oversampling. The findings demonstrate how model stability, generalization, and overall predictive performance are affected by dataset balancing. Additional K-fold cross-validation on the complete oversampled dataset supports the findings, which show distinct behavioral differences between the undersampled and oversampled versions.

B. Technique 1 - Sentiment Analysis

- **Sentiment – Full Undersampled Dataset** Due to the limitations of robust undersampling, the BERT-CNN-BiLSTM model performed moderately well in the full Undersampled sentiment dataset, achieving 78.58% training accuracy and 66.94% test accuracy. The model loses important contextual cues required for nuanced sentiment interpretation when majority-class data is reduced through undersampling, which lowers the recall (66.94%) and F1-score (66.72%). This result suggests that the model's capacity to successfully generalize sentiment signals across a variety of text samples reduces when the dataset is compressed, leading to decreased representational richness.
- **Sentiment – Full Oversampled Dataset** The same model achieved 82.38% training accuracy and 73.43% test accuracy, along with a higher F1-score of 73.71%, indicating noticeable improvements in the full oversampled sentiment dataset. By replicating minority-class samples, oversampling increases data diversity and enables the model to identify more evenly distributed sentiment patterns. Precision and recall are better aligned as a result of preventing majority-class bias. The gradual improvement in all metrics suggests that oversampling improves the model's capacity to more fairly and robustly identify both positive, negative and neutral sentiments.
- **Sentiment – K-fold on Full Oversampled Dataset** With test accuracies ranging from 68.69% to 77.37% across folds and F1-scores between 68.75% and 77.43%, K-fold evaluation on the Full OvS sentiment dataset further confirms its stability. These findings demonstrate that even when trained on distinct data partitions, oversampling preserves dependable performance. The benefits of balanced training data are emphasized by the folds with higher scores, and improved generalization and decreased variance are confirmed by the consistency of all folds. All things considered, the k-fold analysis shows that oversampling offers a strong basis for sentiment classification in Bangla text.

According to the sentiment analysis results, oversampling the entire dataset consistently outperforms undersampling, especially in terms of test accuracy and F1-score. Important contextual variations are lost in the undersampled dataset, whereas more balanced learning across sentiment classes is

made possible by oversampling. The stability of the model is further confirmed by K-fold evaluation on the complete oversampled dataset, which shows trustworthy generalization across folds.

C. Technique 1 - Model Comparisons with BERT-CNN-BiLSTM

- **Headline – Full Oversamling Dataset** In table V The BERT-CNN-BiLSTM model demonstrated its superior ability to extract contextual and semantic cues from the entire oversampled dataset, as evidenced by its 78.57% test accuracy and 79% F1-score for headline classification under Technique-1. Due to their inability to handle long-range dependency and class imbalance, traditional models like LSTM, BiLSTM, and BiGRU demonstrated noticeably worse results. In contrast, CNN and GRU performed moderately, with test accuracies ranging from 71 to 75% and F1-scores between 63 and 74%.
- **Sentiment – Full Ovesampling Dataset** In table V The BERT-CNN-BiLSTM once again demonstrated the best performance for sentiment classification, achieving 73.43% test accuracy and 73.71% F1-score, while baseline RNN-based models (LSTM, BiLSTM, GRU, and BiGRU) suffered greatly with F1-scores in the 35–48% range. CNN's performance was mediocre (63.54% accuracy, 58% F1), but it was still unable to match transformer-based performance. Overall, Technique-1 results demonstrate that compared to traditional deep-learning models, the hybrid BERT-CNN-BiLSTM architecture consistently captures richer contextual information and manages full-dataset oversampling more successfully.

Tables VI represent the details result of headline and sentiment classification over imbalanced, undersampled and oversampled dataset by using technique 2. Tables VII represent the stabilty of highest result of technique 2 both headline and sentiment by using k-fold cross-validation.

D. Technique 2 - Headline Classification

- **Headline – Imbalance Dataset** The BERT-CNN-BiLSTM model performed best when using the original imbalanced training set, achieving 81.37% test accuracy and 81.54% F1-score. This suggests that maintaining the full data distribution allowed the model to accurately and unbiased capture real class patterns. The model was able to generalize across headline categories because, in contrast to aggressive resampling, the imbalanced setup preserved richer contextual diversity. Despite the imbalance, the model managed minority classes fairly well, as further evidenced by the strong precision–recall balance.
- **Headline – Undersampled Dataset** Because training samples were significantly removed, the undersampled version performed worse (75.43% test accuracy, 75.90% F1-score). Large segments of the majority classes were removed, which reduced contextual variation and made it more difficult for the model to understand intricate semantic relationships in headlines. As a result, undersampling has a negative effect on headline classification in Technique 2, as the model showed lowered generalization strength and increased sensitivity to data availability.
- **Headline – Oversampled Dataset** Although the oversampled version outperformed the imbalanced dataset, it produced moderate gains over the undersampled version (75.31% test accuracy, 75.70% F1-score). The artificial repetition of minority samples created little semantic diversity, even though oversampling was successful in balancing class frequencies and minimizing bias toward majority classes. Only slight improvements over the undersampled data were obtained because of this redundancy, which limited the model's capacity to learn more complex linguistic cues. Consequently, oversampling did not achieve the stronger generalization and stability seen with the original imbalanced dataset, even though it improved fairness across categories.
- **Headline – K-fold on Imbalanced Dataset** The imbalanced dataset was further evaluated using 5-fold cross-validation, which showed strong consistency across folds, because it obtained the highest accuracy and F1. The model's consistent performance was confirmed by test accuracies, which varied from 76.23% to 78.06%, irrespective of data partitioning. The model successfully learned headline patterns under the natural distribution, confirming the effectiveness of training without resampling in Technique 2, as evidenced by the close alignment of precision, recall, and F1 across folds.

The results of Technique-2 demonstrate that the best and most reliable performance is obtained when training on the original imbalanced dataset. By eliminating crucial contextual information, undersampling significantly lowers accuracy. Although oversampling enhances class balance, the naturally distributed dataset is still superior.

E. Technique 2 - Sentiment Analysis

- **Sentiment – Imbalance Dataset** The baseline imbalanced sentiment dataset provided a moderate performance, with an F1-score of 64.19% and a test accuracy of 64.46%. Because of the unequal distribution of classes, the model was less able to accurately learn minority sentiment triggers and instead concentrated more on majority sentiment categories. This resulted in limited recall and F1-score, suggesting that when trained directly on an imbalanced dataset, the model had trouble predicting across all sentiment classes.
- **Sentiment – Undersampled Dataset** Performance was further lowered by undersampling, resulting in an F1-score of 60.92% and test accuracy of 59.77%. A significant amount of sentiment context was lost because undersampling eliminates significant portions of the majority class. The model's capacity to discern subtle emotional expressions was limited by this decrease in data diversity, which led to poorer generalization and the lowest scores of all dataset variations in Technique-2.

Dataset	Train (%)	Val (%)	Test (%)	Precision (%)	Recall (%)	F1 (%)
Headline – Imb	**85.00**	**77.92**	**81.37**	**82.49**	**81.37**	**81.54**
Headline – UnS	79.55	76.89	75.43	79.39	75.43	75.90
Headline – OvS	82.12	75.40	75.31	78.49	75.31	75.70
Sentiment – Imb	70.22	**67.62**	**64.46**	64.70	**64.46**	**64.19**
Sentiment – UnS	62.66	60.64	59.77	**66.60**	59.77	60.92
Sentiment – OvS	**77.13**	62.24	61.94	65.15	61.94	62.98

TABLE VI: Performance comparison of BERT–CNN–BiLSTM model using Technique 2

Fold	Headline – Imbalanced Dataset						Sentiment – Imbalanced Dataset					
	Train	Val	Test	P	R	F1	Train	Val	Test	P	R	F1
1	83.62	76.48	78.06	82.36	78.06	78.44	76.33	66.90	68.23	68.07	68.23	64.86
2	81.48	78.06	77.49	80.74	77.49	77.88	80.44	67.91	68.11	66.28	68.11	64.73
3	85.97	77.61	77.37	80.75	77.37	77.68	69.54	67.24	65.03	66.50	65.03	60.44
4	82.43	76.11	76.91	80.01	76.91	77.21	74.71	67.88	67.09	67.86	67.09	63.90
5	81.77	76.04	76.23	80.60	76.23	76.42	76.21	68.45	67.09	65.75	67.09	63.80
Avg	**83.05**	**76.86**	**77.61**	**80.89**	**77.61**	**77.93**	**75.85**	**67.68**	**67.11**	**66.89**	**67.11**	**63.55**

TABLE VII: K-fold cross-validation results of BERT–CNN–BiLSTM model on Headline and Sentiment Imbalanced datasets using Technique 2

Model	Headline Dataset (Technique 2)						Sentiment Dataset (Technique 2)					
	Train	Val	Test	P	R	F1	Train	Val	Test	P	R	F1
CNN	75.21	62.13	69.94	71.00	70.00	70.00	78.22	64.87	63.54	56.00	64.00	58.00
LSTM	69.84	59.47	62.40	63.00	62.00	62.00	52.59	52.52	52.57	0.28	0.53	0.36
BiLSTM	65.40	62.12	61.71	54.00	62.00	56.00	52.59	52.52	52.57	28.00	0.53	0.36
GRU	72.78	52.78	55.08	58.00	55.00	56.00	52.98	50.75	51.54	27.00	52.00	35.00
BiGRU	78.70	60.05	60.00	65.00	60.00	57.00	52.98	50.75	51.54	27.00	52.00	35.00
BERT–CNN–BiLSTM	**85.00**	**77.92**	**81.37**	**82.49**	**81.37**	**81.54**	**70.22**	**67.62**	**64.46**	**64.7**	**64.46**	**64.19**

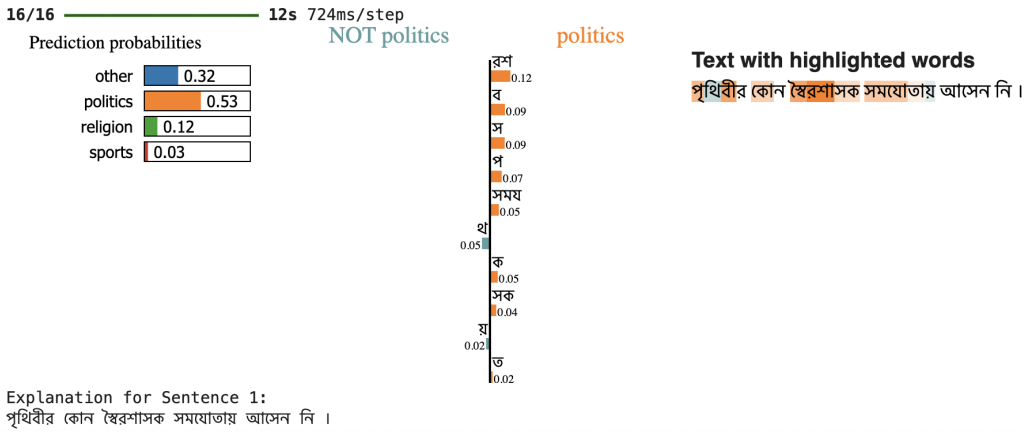
TABLE VIII: Model comparison on Headline and Sentiment datasets using Technique 2

- **Sentiment – Oversampled Dataset** The oversampled version achieved a 62.98% F1-score and 61.94% test accuracy, which was a slight improvement over the undersampled version. The advantages of oversampling for this task were limited because replicated or synthetic examples might not accurately capture the complex nature of actual sentiment expressions in Bangla text, even though it improved minority-class representation and somewhat balanced the dataset. Balanced class exposure is beneficial, as evidenced by the improvement over undersampling; however, the lack of richer, more varied training samples limited performance.
- **Sentiment – K-fold on Imbalanced Dataset** More stable behavior was found using K-fold evaluation on the imbalanced dataset; test accuracy varied between folds, ranging from 67.09% to 68.23%. The model's consistent performance in the face of class imbalance was confirmed by the average performance, which stayed near the initial single-split results. Nevertheless, the model continues to favor majority sentiment classes, as evidenced by the F1-scores across folds (60–65%), which show that class-wise prediction difficulty remains. K-fold validation demonstrates that dataset-level imbalance is still the main bottleneck for this task, even though it also validates robustness.

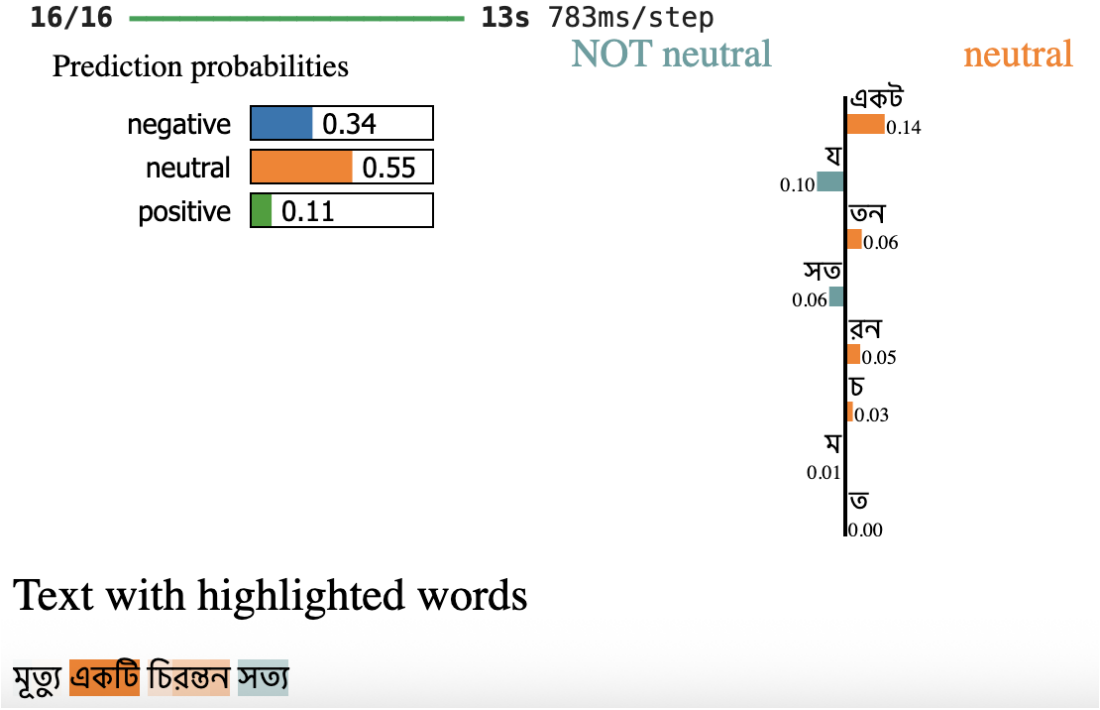
Undersampling decreased data diversity and performance (59.77% accuracy), while the imbalanced dataset favored majority classes, producing moderate accuracy (64.46%) and F1-score (64.19%). While minority-class learning was marginally enhanced by oversampling (61.94% accuracy), synthetic examples lacked depth. Stable but limited performance was confirmed by K-fold evaluation, with class imbalance continuing to be the primary obstacle.

F. Technique 2 - Model Comparisons with BERT-CNN-BiLSTM

- **Headline – Imbalanced Dataset** in table VIII we observed that BERT-CNN-BiLSTM performed better than all other models tested for headline classification, with an F1-score of 81.54% and an 81.37% test accuracy, indicating its superior capacity to capture contextual and semantic information. LSTM and BiLSTM performed marginally worse because of their inability to capture long-range dependencies in the imbalanced dataset, whereas CNN and GRU variants demonstrated moderate performance with test accuracies of roughly 55–70% and F1-scores of roughly 56–70%. Under Technique-2, using a hybrid transformer-CNN-BiLSTM architecture greatly increased recall and precision on imbalanced headline data.
- **Sentiment – Imbalanced Dataset** Compared to conventional RNN-based models, in table VIII we observed



(a) Headline Example 1



(b) Sentiment Example 1

Fig. 3: LIME-based explanations for Headline (top) and Sentiment (bottom) classification using the proposed BERT-CNN-BiLSTM model. The highlighted words show the most influential features contributing to the predicted class.

the BERT-CNN-BiLSTM model performed the best in sentiment classification, handling imbalanced classes with an F1-score of 64.19% and a test accuracy of 64.46%. In contrast to LSTM, BiLSTM, GRU, and BiGRU, which struggled with low precision, recall, and F1-scores (roughly 35–36%), CNN performed moderately (63.54% test accuracy, F1 58%), suggesting that it was difficult to capture minority-class sentiment cues. Overall, under Technique-2, the hybrid transformer-CNN-BiLSTM architecture produced the most robust and balanced results for imbalanced sentiment data.

Based on imbalanced datasets, the BERT-CNN-BiLSTM model achieved 81.37% test accuracy (F1 81.54%) for headlines and 64.46% test accuracy (F1 64.19%) for sentiment, outperforming all baseline architectures. Because of their limited capacity to manage long-range dependencies and class imbalance, traditional models such as CNN, LSTM, BiLSTM, and GRU demonstrated moderate to poor performance. Our approach combines local and global semantics in a transformer-based manner for headline classification by combining transformer-based contextual embedding (BERT), convolutional feature extraction (CNN), and sequential dependency modeling (BiLSTM). The same architecture uses CNN for local features, BERT for contextual representation, and BiLSTM for long-term dependencies to produce robust headline and sentiment classification results.

G. Model Interpretablilty - LIME

- **Headline Classification** The first LIME visualization shows in Figure 3 (top) shows a Bangla sentence pre-

dicted as politics with a probability of 0.53, while the next closest category is other at 0.32. The highlighted words strongly contributed to the politics class, shown in orange bars on the right. Meanwhile, some other words had weaker contributions or even neutral/contradictory signals, slightly supporting "not politics." This indicates that the model is able to identify domain-specific political terms and use them as decisive features in classification.

- **Sentiment Analysis** The second LIME visualization shows in Figure 3 (bottom) shows a Bangla sentence predicted as neutral with a probability of 0.55, while negative (0.34) and positive (0.11) received lower scores. The highlighted tokens made the strongest contributions toward the neutral class, as indicated by the orange bars on the right. In contrast, words like (truth) provided weaker opposing contributions that slightly favored a non-neutral sentiment, reflected in blue. This outcome indicates that the model tends to rely on descriptive or abstract terms to infer neutrality, whereas conceptually heavier words exert a smaller pull toward more emotional classes. The result highlights a moderate confidence prediction where neutrality is supported by structural cues rather than overt emotional signals.

Dataset	Train	Val	Test	P	R	F1
Potrika Dataset	85.42	84.00	84.34	84.62	84.34	84.30
Advanced Sentiment	77.56	71.73	71.52	74.72	71.52	71.52

TABLE IX: Performance of BERT–CNN–BiLSTM (Technique 2) on External Datasets

H. External Dataset Validation - Technique 2

- **Potrika Datase - Headline Classification** Using an external eight-class news dataset containing 665K articles, we sampled 24,000 headlines (3000 from each class) to create a balanced evaluation set [33]. In table IX The hybrid model achieved 84.34% accuracy, precision, recall, and F1-score, proving robustness across larger and more balanced collections. This result shows that the model is not overfitted to the primary dataset but generalises well to new large-scale corpora. Furthermore, the consistent performance across all classes highlights the ability of the hybrid architecture to handle diverse topics effectively.
- **Advanced Sentiment Dataset - Sentiment Analysis** We used an imbalanced dataset with 8738 negative, 7328 strongly negative, 7697 neutral, 6246 positive, and 4803 strongly positive samples for sentiment classification [40]. The BERT-CNN-BiLSTM model demonstrated strong performance despite class imbalance, achieving 73.05% test accuracy without the use of oversampling in Table IX. Its precision, recall, and F1-score were 76.28%, 73.05%, and 73.39%, respectively. These findings show that our hybrid model is robust even when there are unequal class distributions and successfully captures both local and global sentiment patterns across several classes.

V. Conclusion

In this paper, we presented a unified model for Bangla text classification that jointly learns headline categorization and sentiment analysis, which were studied in isolation previously. The work used a hybrid BERT-CNN-BiLSTM model using two experimental strategies and investigated several versions of the dataset (imbalanced, undersampled, and oversampled before and after splitting dataset) by performing 5-fold cross-validation with different evaluation metrics like Accuracy, Precision, Recall and F1-score. LIME-XAI is used to understand predictions and interpret the model for more transparency. The experimental results proved the proposed framework achieved higher performance gains than those methods compared in baseline studies, including ML, DL and hybrid works and brought additional contributions from joint representation of headlines and sentiment. Indeed, this work is an important contribution to the Bangla NLP community as it establishes a strong baseline for processing a low-resource language and opening new directions for future work in multilingual and domain-specific applications.

References

[1] Mustafa, A. A. & Abdulazeez, A. M. A review of text classification based on ML & data mining algorithms. *The Indonesian Journal of Computer Science*. **13**(3) (2024)

[2] Palanivinayagam, A., El-Bayeh, C. Z. & Damaševičius, R. Twenty years of machine-learning-based text classification: A systematic review. *Algorithms*. **16**(5), 236 (2023)

[3] Hassan, S. U., Ahamed, J. & Ahmad, K. Analytics of machine learning-based algorithms for text classification. *Sustainable Operations And Computers*. **3**, 238–248 (2022)

[4] Garrido-Merchan, E. C., Gozalo-Brizuela, R. & Gonzalez-Carvajal, S. Comparing BERT against traditional machine learning models in text classification. *Journal of Computational and Cognitive Engineering*. **2**(4), 352–356 (2023)

[5] Roy, A., Sarkar, K. & Mandal, C. K. Bengali text classification: A new multi-class dataset and performance evaluation of machine learning and deep learning models. (2023)

[6] Salehin, K., Alam, M. K., Nabi, M. A., Ahmed, F. & Ashraf, F. B. A comparative study of different text classification approaches for Bangla news classification. In *2021 24th International Conference On Computer And Information Technology (ICCIT)* pp. 1–6. IEEE (2021)

[7] Alam, T., Khan, A. & Alam, F. Bangla text classification using transformers. *arXiv Preprint arXiv:2011.04446*. (2020)

[8] Emon, E. A., Rahman, S., Banarjee, J., Das, A. K. & Mittra, T. A deep learning approach to detect abusive Bengali text. In *2019 7th International Conference On Smart Computing & Communications (ICSCC)* pp. 1–5. IEEE (2019)

[9] Saha, U., Mahmud, M. S., Chakrobortty, A., Akter, M. T., Islam, M. R. & Al Marouf, A. Sentiment classification in Bengali news comments using a hybrid approach with Glove. In *2022 6th International Conference On Trends In Electronics And Informatics (ICOEI)* pp. 01–08. IEEE (2022)

[10] Alam, S., Haque, M. A. U. & Rahman, A. Bengali text categorization based on deep hybrid CNN–LSTM network with word embedding. In *2022 International Conference On Innovations In Science, Engineering And Technology (ICISET)* pp. 577–582. IEEE (2022)

[11] Mahmud, H., Mahmud, H. & Rashid, M. R. A. Enhancing sentiment analysis in Bengali texts: A hybrid approach using lexicon-based algorithm and pretrained language model Bangla-BERT. *arXiv Preprint arXiv:2411.19584*. (2024)

[12] Hassan, S. B., Noor, M. A., Forhad, S., Tasnim, J. & Siddique, A. H. Analyzing TF-IDF and BERT approach for Bangla text classification using transformer-based embedding for newspaper sentiment classification. In *International Conference On Computational Intelligence In Pattern Recognition* pp. 375–390. Springer (2024)

[13] Saha, P. & Sultana, N. Sentiment analysis from Bangla text review using feedback recurrent neural network model. In *Communication and Intelligent Systems: Proceedings of ICCIS 2020*, pp. 423–434. Springer (2021)

[14] Hoq, M., Haque, P. & Uddin, M. N. Sentiment analysis of Bangla language using deep learning approaches. In *International Conference on Computing Science, Communication and Security*, pp. 140–151. Springer (2021)

[15] Karmakar, D. R., Mukta, S. A., Jahan, B. & Karmakar, J. Sentiment analysis of customers' review in Bangla using machine learning approaches. In *Innovations in Computer Science and Engineering: Proceedings of the Ninth ICICSE, 2021*, pp. 373–384. Springer (2022)

[16] Bhowmik, N. R., Arifuzzaman, M. & Mondal, M. R. H. Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. *Array*. **13**, 100123. Elsevier (2022)

[17] Haque, R., Islam, N. & Tasneem, M. & Das, A. K. Multi-class sentiment classification on Bengali social media comments using machine learning. *International Journal of Cognitive Computing in Engineering*. **4**, 21–35. Elsevier (2023)

[18] Samin, T., Mouri, N. A., Haque, F. & Islam, M. S. Sentiment analysis of Bangladeshi digital newspaper by using machine learning and natural language processing. In *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pp. 149–153. IEEE (2024)

[19] Mahmud, A., Antu, M. A. M. & Nayeem, G. M. Transformers vs. traditional models: Analyzing media bias in Bangladeshi news through sentiment analysis. In *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–5. IEEE (2025)

[20] Bhuiyan, M. R., Keya, M., Masum, A. K. M., Hossain, S. A. & Abujar, S. An approach for Bengali news headline classification using LSTM. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 1*, pp. 299–308. Springer (2021)

[21] Hossain, T., Islam, A. R., Mehedi, M. H. K., Rasel, A. A., Abdullah-AL-Wadud, M. & Uddin, J. BanglaNewsClassifier: A machine learning approach for news classification in Bangla newspapers using hybrid stacking classifiers. *PLoS One*. **20**(6), e0321291. Public Library of Science (2025)

[22] Khushbu, S. A., Masum, A. K. M., Abujar, S. & Hossain, S. A. Neural network based Bengali news headline multi-classification system: Selection of features describes comparative performance. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6. IEEE (2020)

[23] Chowdhury, O., Ahmed, M., Ara, M. T., Reno, S. & Alam, A. Bengali news headline categorization: A comprehensive analysis of machine learning and deep learning approach. (n.d.)

[24] Chowdhury, P., Eumi, E. M., Sarkar, O. & Ahamed, M. F. Bangla news classification using GloVe vectorization, LSTM, and CNN. In *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021*, pp. 723–731. Springer (2021)

[25] Rana, S., Haque, M., Sultana, N., Amid, A., Hosen, M. & Islam, S. Newsnet: A comprehensive neural network hybrid model for efficient Bangla news categorization. *2024 15th International Conference On Computing Communication And Networking Technologies (ICCCNT)*. pp. 1–7 (2024)

[26] Sultana, Z., Tasnia, R., Rahman, N., Prapti, A. & Das, O. Attention Based Bengali News Classification System Using Hybrid Deep Learning Model. *2025 International Conference On Electrical, Computer And Communication Engineering (ECCE)*. pp. 1–6 (2025)

[27] Ahmed, M. BAN-ABSA: An Aspect-Based Sentiment Analysis Dataset for Bengali. Kaggle. (2020), https://www.kaggle.com/datasets/mahfuzahmed/banabsa, Accessed: 2025-09-17

[28] Jarrahi, A., Mousa, R. & Safari, L. SLCNN: Sentence-level convolutional neural network for text classification. *arXiv Preprint arXiv:2301.11696* (2023)

[29] Yao, L. & Guan, Y. An improved LSTM structure for natural language processing. *2018 IEEE International Conference Of Safety Produce Informatization (IICSPI)*. pp. 565–569 (2018)

[30] Elfaik, H. Deep bidirectional LSTM network learning-based sentiment analysis for Arabic text. *Journal Of Intelligent Systems*. **30**(1), 395–412 (2021)

[31] Khudhair, I., Majeed, S., Ahmed, A., Alsaeedi, M. & Aswad, F. An Improved Hybrid GRU and CNN Models for News Text Classification. *JOIV: International Journal On Informatics Visualization*. **9**(1), 303–313 (2025)

[32] Wang, X. & Wang, R. Text sentiment classification based on Vit-BiGRU-attention model. *International Conference On Artificial Intelligence And Intelligent Information Processing (AIIIP 2022)*. **12456**, 338–343 (2022)

[33] Ahmad, I., AlQurashi, F. & Mehmood, R. Potrika: Raw and balanced newspaper datasets in the Bangla language with eight topics and five attributes. *arXiv Preprint arXiv:2210.09389* (2022)

[34] Bhowmik, N., Arifuzzaman, M. & Mondal, M. Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. *Array*. **13**, 100123 (2021)

[35] Devarajan, G., Nagarajan, S., Amanullah, S., Mary, S., Bashir, A. AI-Assisted Deep NLP-Based Approach for Prediction of Fake News From Social Media Users. *IEEE Transactions On Computational Social Systems*. **11**(4), 4975–4985 (2024)

[36] Aurna, N., Yousuf, M., Taher, K., Azad, A. & Moni, M. A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models. *Computers In Biology And Medicine*. **146** pp. 105539 (2022)

[37] Kim, M., Park, J. & Kim, Y. IM-BERT: Enhancing Robustness of BERT through the Implicit Euler Method. *arXiv Preprint arXiv:2505.06889* (2025)

[38] Aurpa, T., Fariha, K., Hossain, K., Jeba, S., Ahmed, M., Adib, M., Islam, F. & Akter, F. Deep transformer-based architecture for the recognition of mathematical equations from real-world math problems. *Heliyon*. **10**(20) (2024)

[39] Han, D., Liu, Q. & Fan, W. A new image classification method using CNN transfer learning and web data augmentation. *Expert Systems With Applications*. **95** pp. 43–56 (2018)

[40] Tripto, N. I. & Ali, M. E. Detecting multilabel sentiment and emotions from Bangla YouTube comments. *2018 International Conference On Bangla Speech And Language Processing (ICBSLP)*. pp. 1–6 (2018)