International Journal of Computing and Digital Systems

TransNeuNet: Where Transformers Meet Neurons to Revolutionize Bangla News Classification

Ronjon Kar[1], Md. Touhidur Rahman Limon[1], Tanjila Akter[1], Tausif Ahmed[1] and Dr. Anisur Rahman[1]

[1]*Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh*

Abstract: Due to the rapid growth of news online every day, we urgently need an automated system to categorize this huge amount of news. There is a lot of works for categorizing English and other news, but very little work done for categorizing Bengali news. So, in our research, we proposed a hybrid deep learning model. We combined Bengali BERT and T5 embedding and classified the Bengali article using the ANN Model. We collected our dataset from two sources. Dataset 1 was collected from Kaggle, and we created Dataset 2 through web scraping. We trained our model with over 55,000 articles and 13 categories. The categories are Sports, Politics, Crime, Entertainment, Education, Jobs, Economy, Environment, Health, International, Religion, Lifestyle, Technology. We preprocessed our dataset by removing stop words, tokenization, label encoding, and date-time formatting. In addition to providing good accuracy, our model also understands changes in different news categories and contextual writing. Experimental results show that the proposed model performs much better than other trained models such as ANN, CNN, BiLSTM, Random Forest, Decision Tree, and SVM. ANN achieved only 88.89% accuracy after BERT embedding, but TransformNeuNet, which combines embeddings (BERT and T5), shows 93.53% accuracy. We also used LIME to interpret our predictions.

Keywords: TransNeuNet, NLP, Deep Learning, Transformer, Neural Network, Bangla News Classification

1. INTRODUCTION

With the rise in popularity of online news platforms and social media, the amount of digital news has also increased. Bangladesh is one of the largest populated countries in the world. A major portion of people in this country is very much interested to go through the Bengali news from the internet [1]. To organize this information properly and use it as needed, it is now necessary to use automated classification systems. This system makes it possible to quickly sort different types of news such as politics, sports, entertainment, technology and health related news. It not only helps readers find specific news but also helps applications such as search engines, recommendation systems and digital journalism analytics to find news.

Bengali is the seventh most spoken language in the world. Around 284 million people speak Bengali in Bangladesh, India and other countries [2]. Over 98% of Bangladeshis and a large section of the Indian population speak Bangla as their first language [3]. In NLP research, Bengali receives very little attention compared to English. There are several reasons for the low attention. One reason for this is the lack of high-quality datasets like English. Another reason could be the complex structure of Bengali in the language. It changes the form of words in various ways. It has complex words and different sentence structures, which makes tokenization and processing difficult. Important tools such as lemmatizers, stemmers and reliable trained models are not yet developed for the Bengali language. Bengali news often contains a mix of Bengali and English words. In addition, there are spelling variations and sometimes informal writing, which makes text cleaning and normalization more difficult.

Technological advances have solved many problems in Bengali text processing, which are not enough but very important. Models like Bangla BERT provide contextual embeddings [4], while Bangla T5 provides semantically rich representations with its text to text pretraining technique. By combining these two approaches, it is possible to create powerful feature representations for Bengali text. For this Bengali news classification system, we have combined Bangla BERT and Bangla T5 to create a powerful hybrid embedding system.

The contributions of our work are as follows:

- News was collected from various Bengali newspapers, and a large Bengali news dataset was created.
- Different preprocessing systems have been used at

E-mail address: 2022-1-60-091@std.ewubd.edu, 2022-1-60-272@std.ewubd.edu, 2022-1-60-078@std.ewubd.edu, 2022-1-60-315@std.ewubd.edu, anis@ewubd.edu

different stages for data processing.

- A powerful embedding system was created by combining Bangla BERT and Bangla T5.
- We have applied different deep learning and transformer-based models for better prediction.
- Performance was evaluated using precision, accuracy, recall, F1-score, confusion matrix, and ROC curve.
- We have done model explanation using LIME so that it is clear why the prediction was made.

Along with the classification system, we have created a user-friendly and easily accessible web application using the 'Streamlit' library. Streamlit is an open source Python framework that makes data manipulation very easy. It is a simple and powerful Python tool for building data-driven apps quickly. In this web app, when users write Bengali articles, it will show the respected predictions very quickly. This web application makes our system more effective and usable in real life. Overall, Our method works like a framework, it can be reused for sentiment analysis, fake news detection, recommendation systems, and many more.

This section presents the perspective of text classification systems. The next section is organized as follows. In the second section, we discuss various related works on text classification systems. In the third section, we describe our overall working methodology, and in the fourth section, we discuss the results and limitations of our work. Finally, in the fifth section, we conclude our work and talk about what we will do to achieve better results in the future.

2. Related Work

Nowadays, with the huge amount of news appearing in newspapers every day, regarding text classification methods, many great contributors have carried out remarkable work. They have discovered various methods of text classification. Below, we have highlighted some related work patterns of different authors. We tried to show it from our perspective, how they have sought to solve the problem. After that, we will discuss our work.
Hossain, M. R. et al. [5] used a balanced Bengali news collection of 3,000 articles and divided them into 12 different categories (crime, economy, education, entertainment, environment, international, opinion, politics, science, technology, sports, accidents, industry). Each category contains 250 articles. They then compared some machine learning and deep learning models for news classification. They achieved a maximum accuracy of 93.43% for CNN and 91% for Linear SVM and 92.60% for Bi-LSTM. Although their accuracy is good, their model is over-fitted. They do not get good results for new types of data. They also do not remove stop words that can create some noise in their analysis. In a research study, Hossain, A. et al. [6] worked only on titles, not on the entire content. They used over 100,000 titles across 8 categories. The classes in this dataset [7] are unbalanced. Since they analyzed only titles, it could not achieve better accuracy, and the model could not fully understand all the patterns in the text. They trained two deep learning models, GRU and LSTM. The accuracy of LSTM is 82.74%, while that of GRU is 87.48%. Although they trained only two models, they could not determine which one should be considered the better model for the study. They may not obtain better output for real-time data. Khushboo, S. A. et al. [8] also trained on headlines only. For their study, they used 8,602 Bengali news articles divided into 11 categories. The data is not enough for analysis. They trained many different kinds of machine learning models, such as Random Forest, Naive Bayes, SVM, and Logistic Regression. The models' accuracy was the lowest. Not one of them went over 43%. They also used a neural network-like deep learning model. Although achieving 90%, the neural network performs poor with unseen data. Amin, R., et al.'s [9] research only worked with news headlines. The model cannot fully represent the context of the news content . They collected 88,968 Bengali news headlines from 7 different categories. The dataset is unbalanced because some categories have more headlines than others. In comparison to earlier models like Naïve Bayes, SVM, and MLP, they introduced a parallel 1D CNN with word based data augmentation and achieved 93.47% accuracy. Also, they kept their dataset private, which limits future research. They suggested using GANs in the future to improve augmentation, classify entire articles, and improve accuracy and recall. Rahman, M. M. et al. [1] used 14,400 Bengali news articles from Prothom Alo which divided into 6 categories. They showed a classification model named Text-GCN. With an accuracy of 96.25%, the model performed better than models such as BiLSTM, GRU-LSTM, LSTM, Char-CNN, and BERT. However, their method only worked on a small dataset with few categories and used a lot of memory. In another research, Yeasmin, S et al. [10] collected 12,000 Bengali news articles from Prothom Alo divided into 5 categories. They compared neural network models, deep learning, and usual machine learning. Text-GCN performed better than CNN (92.1%) and BiLSTM (93.4%) with the highest accuracy of 94.8%. However, the model needs a lot of GPU power, and the dataset was collected from a single source with few categories. They suggested the use of transformer models, real-time applications, and multiple source datasets in the future. Chowdhury, P. et al. [11] used a CNN–LSTM hybrid model with GloVe embeddings to classify 14,000 Bengali news articles into 10 categories. The model performed better than SVM, LSTM, CNN, ANN, and BiLSTM, achieving an accuracy of 87% in the test. Rahman, S et al. [12] worked with 28, 666 Bengali news articles from *Daily Ittefaq* divided into 10 categories. They found that Random Forest combined with SMOTE achieved 95% accuracy and F1-score when they used TF-IDF with machine learning models. Balancing the dataset greatly improved performance compared to the imbalanced dataset, but they noticed problems with Unicode processing, limited generalization, and high computational

cost. Mouri, A. G. et al. [4] classified 136,811 Bengali news headlines into 6 categories using machine learning, deep learning, and transformer models. With an accuracy of 86.50%, XLM-RoBERTa performed better than Bangla-BERT as well as traditional ML and DL techniques. However, Performance was affected by the dataset's imbalance and some category overlap. Roy, A. et al. [13] evaluate machine learning and multilingual deep learning models using a 6-class dataset (Dataset-2) and a new 38-class Bengali news dataset (Dataset-1). FastText + SVC (92.61%) and XLM-RoBERTa-Large (95.12%) achieved the best results for Dataset-1 and Dataset-2 respectively. Using 50,000 articles, Ahmad, I. et al. [14] used ML models (LR, SVM, RF, NB) and DL models (LSTM, Bi-LSTM, CNN, GRU) to classify Bengali news into eight categories. GRU + FastText performed better than traditional machine learning models, achieving the best performance with an accuracy of 91.83%. A new 12-class complete Bengali news dataset was presented by Rashid, M. R. et al. [15], who also used deep generative models (VAE, RBM) to extract features. With an accuracy of 94.12%, VAE + SVM achieved the best results, compared to more conventional machine learning techniques like TF-IDF and Word2Vec. However, the method relied on large labeled datasets and was computationally costly. Sikder, M. F. et al. [16] modified the BanglaBERT-large model on a 400k *Prothom Alo* dataset split into 9 classes in order to take benefit from pre-trained transformers. The model showed important words using embedded Gradients for explainability and achieved around 92% accuracy on balanced data. The baseline RNN, Bi-GRU, and Bi-LSTM+CNN models did not perform well. Alam, S. et al. [17] evaluated CNN, ANN, Bi-LSTM, and hybrid CNN+Bi-LSTM models with word embeddings using a large open source Bengali news dataset (SUST, 12 categories). For 10 categories, their accuracy was 88.56%, and for 12 categories, it was 84.93%. Using LSTM, Bi-LSTM, and Bi-GRU models, Hasan, M. K. [18] classified 9,913 Bengali news headlines into six groups. With a validation accuracy of 77.91%, the Bi-LSTM model performed better than the others. However, the small dataset and the morphological complexity of Bengali caused overfitting. Chowdhury, O. et al. [19] used machine learning and deep learning models to classify 30,000 Bengali news headlines into 8 categories. With an accuracy of 84.01%, GRU performed better than the classical ML baselines (65%), Bi-LSTM (83.42%), and LSTM (82.74%). Mugdha, S. B. S. et al. [20] applied a novel rule-based Bangla stemmer with TF-IDF + Logistic Regression to a balanced 5-class subset of the BARD Bangla news dataset (75,000 articles). The stemmer performed better than Random Forest and Multinomial NB, achieving 95.3% test accuracy and improving feature quality. However, the study used single-source data, had few categories, and focused on classical machine learning. A hybrid stacking model (BiLSTM + SVM meta-classifier) called BanglaNewsClassifier was proposed by Hossain, T. et al.[21] and used on an 8-class Bengali news dataset from Kaggle. It performed better than CNN, LSTM, and ML models and achieving an accuracy of about 94%. Jakaria et al. [22] worked with an 8-class Kaggle dataset to test nine traditional machine learning models for Bangla news classification. The best performing models using TF-IDF features were SVM (92.76%), Bagging (92.64%), and Logistic Regression (92.26%). However, the study was based on single-source data, did not have transformer or deep models. Using the Kaggle dataset [23](400k articles, 9 categories), Rana, S. et al. [3] proposed NewsNet, a hybrid CNN + GRU + BiLSTM model for Bangla news classification. NewsNet performed better than all baseline machine learning techniques with an accuracy of 94.57% (best SVM: 88.42%).

In our study, we created a hybrid model named TransformNeuNet that achieved a 93.53% classification accuracy for 13 Bangla news categories. We used 150K news articles collected from our own web scraping efforts as well as to the 400K samples in the publicly available Kaggle Bengali news dataset for training and evaluation. TransformNeuNet performed better than deep learning (DL) architectures, traditional machine learning (ML) models, and even other Transformer-based techniques.

3. Methodology

In Figure 1, we present our proposed workflow. We start our work by collecting datasets from two different sources. Then pre-processing is performed to prepare the data for analysis. Next, the text is embedded for machine understanding. Finally, the model is trained and the results are displayed.

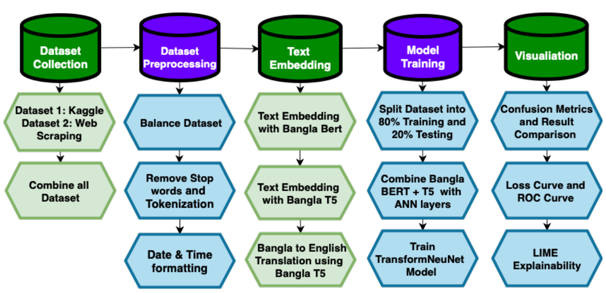

Figure 1. Methodology

A. *Data Collection and Description*

In our work, we used two different Bengali news datasets. By combining the two datasets, we were able to effectively evaluate our classification system and increase its number of categories, which gave us more diverse results. The two datasets are explained below.

1) *Dataset 1*

The Bangla Newspaper Dataset, published on Kaggle [23] by Zabir Al Nazi, is a well-structured and richly annotated collection of 400,000 Bangla news articles sourced from *Prothom Alo*, one of the most prominent newspapers in Bangladesh. This corpus provides a clearly defined categorization framework, consisting of 9 news categories such as Bangladesh, International, Sports, Economy, Opinion, Entertainment, Education, Technology, and Lifestyle. In Figure 2, we show the class distribution of dataset 1.

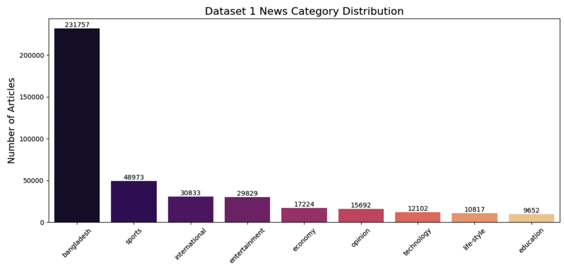

Figure 2. Dataset 1 Class Distribution

2) *Dataset 2*

There are not many classes in dataset-1. We need more data of different classes to classify more different articles. So We created a dataset of Bengali newspapers through web scraping and stored the data in a CSV file. We collected about 150,000 news items from four different newspapers: *Bangladesh Pratidin*, *Bangla Vision*, *Prothom Alo*, and *Dainik Inqilab*. There were about 20 categories, but most classes did not have enough data. So, we kept the information of only 13 classes. We used web scraping to get the headline, category, publication date, full article, and source of each news item. In Figure 2, we show the class distribution of dataset 2.

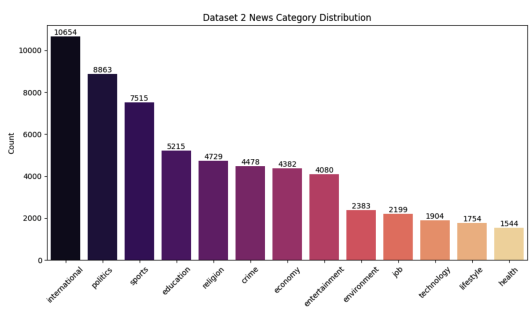

Figure 3. Dataset 2 Class Distribution

B. *Data Preprocessing*

After collecting all the data, we merged the datasets and aligned their schemas. After aligning, we removed some features that were not related to our research. Then we noticed a significant class imbalance, the Sports class had 160,000 records and the Religion class had only 5,000. To achieve balanced results, we applied undersampling. This saved 5,000 records from each class for further processing. Next, we removed common stopwords that appeared frequently in the overall dataset. Additionally, for specific classes we manually removed some stopwords that were not important to that class but might be important for others. After stopword removal, we generated word clouds and examined the top unigrams and bigrams. If any frequent words were still found we removed these. This process was repeated iteratively to ensure that the number of extra common words in the dataset was minimized at each step.

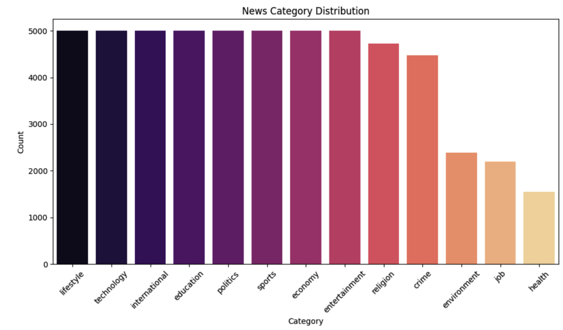

Figure 4. Balanced Class Distribution

1) ***Text Cleaning***

Before running a model, we always need to clean the dataset first. For our work, we cleaned the content by following these steps. First, we removed 800+ stop words from the overall dataset. Stop words are common words that do not help in classification and do not add much value to understanding the overall content. After removing stop words, we removed non-Bangla words, numbers, URLs, and some special characters from the text. This helps us find rare words related to a particular class from the text.

2) ***Tokenization***

Before feeding the model, we need to extract specific words related to a particular class. These words help identify patterns in the dataset for efficient classification. To find these specific words, we explore all the records and identify certain token words. Then, we tokenize each record and store it in an array using Unicode (text representation) and the Python library 're'. The total number of tokens in the overall dataset is more than 1 crore. Using these tokens we find patterns from each class and it will be easy to classify a particular category.

3) ***Label Encoding***

Machine learning models do not understand categorical labels. To understand this, we need to convert categorical labels to numeric labels. We use Scikit-learn's label encoder to convert categorical to numeric labels. This process is known as label encoding. It is an important step before training a model.

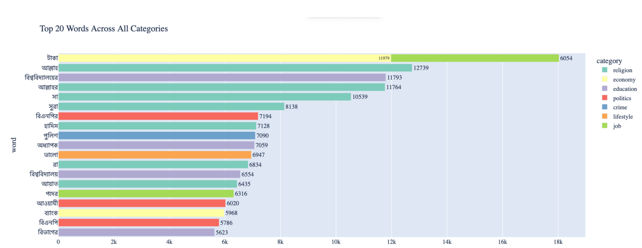

Figure 5. Most Frequent Words Across Categories

Figure 6. Word Cloud of Sports

Figure 7. Word Cloud of Crime

A word cloud is a visual that shows the most commonly used words in a dataset. Big words appear more often in word clouds than small words. In figure 6 and 7, we show just two categories. This indicates the most common words in a particular category.

C. Exploratory Data Analysis

To better understand news articles, we use a temporal analysis that shows how news changes over time. First, we convert the Bengali dates into date-time format so that the computer can interpret them correctly. We extracted some important features from there like articles by day of the week, month-year heatmap, articles by year by category, etc.

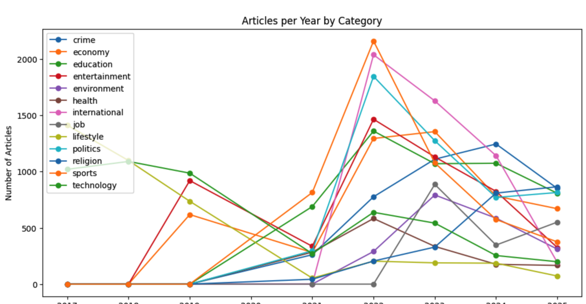

Figure 8. Amount of News Article Per Year

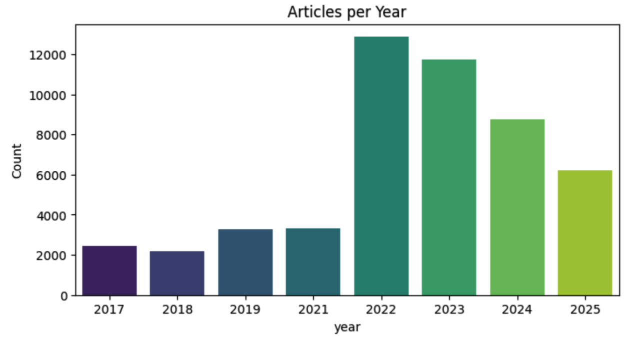

Figure 9. Article Per Year

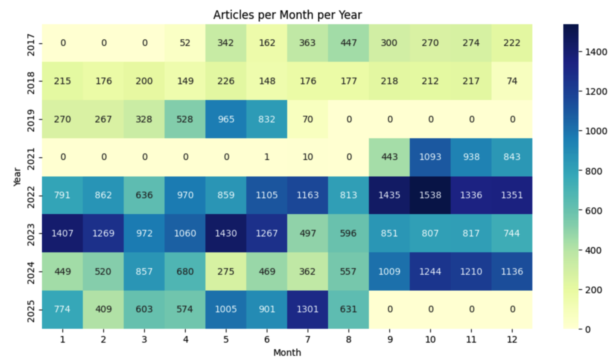

Figure 10. Article Distribution Heatmap per Year and Month

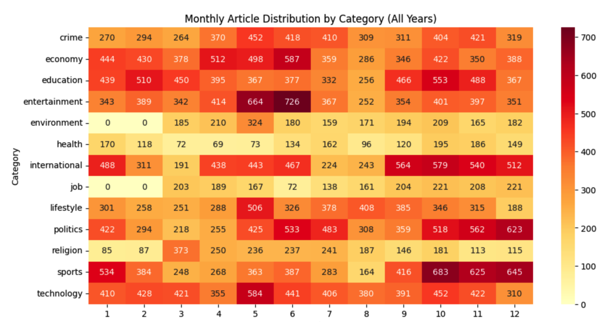

Figure 11. Monthly Article Distribution Heatmap

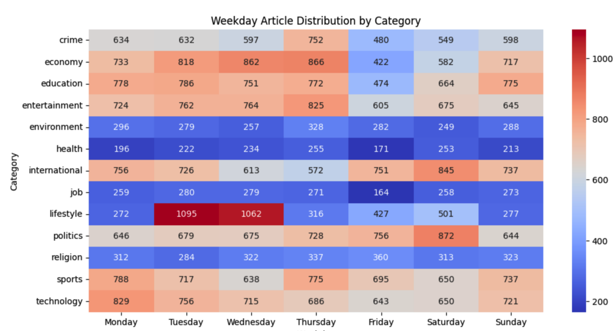

Figure 12. Weekday Article Distribution Heatmap

Figure 8 shows the number of news articles by category for each year, and Figure 9 shows the total number of articles per year. The number of articles published per month for different years is also shown in Figure 10. Finally, Figures 11 and 12 show the distribution of news articles by category on a monthly and weekly basis, respectively.

D. Dataset Split

We used 55,333 articles with 13 classes for training and testing. The training set contains 80% of the articles and the test set contains 20%. This 80% dataset is used to find patterns and 20% data is used to evaluate the performance of the model. Table I shows the number of samples in each set. We have not split the validation set separately here. The test set is to observe the performance of the model during training.

TABLE I. Dataset Split Information

Set	Number of Samples
Training (80%)	44,266
Testing (20%)	11,067

E. Model Description

Bangla Bert: The Bangla-BERT base model is likely constructed on the basis of the original BERT (Bidirectional Encoder Representations from Transformers) architecture, which utilizes a pillar from the transformer encoder layers. Figure 13 shows the architecture of the BERT model. It is generally known as a masked language model (MLM) that has been pre-trained on a vast collection of Bengali text. In a BERT-style encoder, there are L layers, each containing a multi-head self-attention process and a position-wise feed-forward network (FFN) in between. Residual connections and layer normalization are applied after each sub-layer.

Given an input sequence of tokens in Equation (1):

$$X = (x_1, x_2, \dots, x_N) \tag{1}$$

where x_1 is the special [CLS] token and x_i are the subsequent tokens, the BERT model computes a sequence of final hidden states. The corresponding equation shown in equation (2).

$$H = (h_1, h_2, \dots, h_N): \quad H = BERT(X) \tag{2}$$

where $h_i \in \mathbb{R}^D$ is the contextualized embedding D in dimensions for the token i-th, and D is the hidden size (typically 768 for a base model).

The code uses the hidden state corresponding to the first token, which is the [CLS] token, as the embedding of the sentence. This is the standard approach for classification and sentence representation tasks using BERT.

The final embedding of the sentence E_{BERT} is:

$$E_{BERT} = h_1 \tag{3}$$

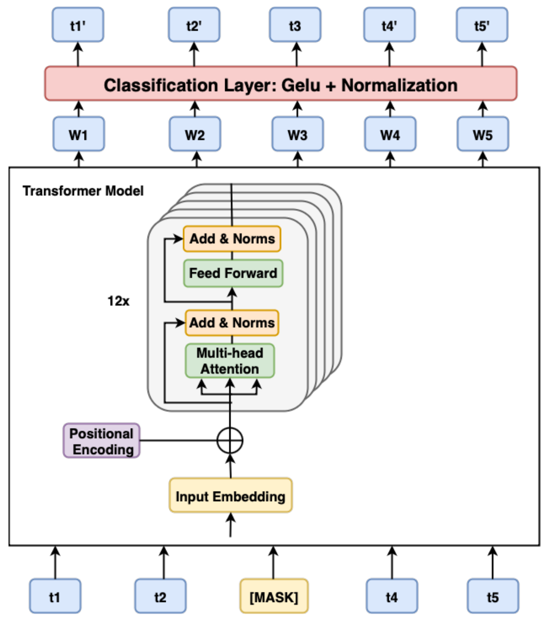

Figure 13. Bangla Bert Model Architecture

Bangla T5: The code uses the encoder of the mT5-base model via T5EncoderModel. Figure 14 shows the architecture of the T5 model. T5 (Text-to-Text Transfer Transformer) models are typically sequence-to-sequence architectures, but to generate embeddings, only the encoder is required. The T5 encoder is also a stack of transformer layers similar to BERT but uses a slightly different architectural design (e.g., pre-normalization instead of post-normalization, shared parameters). The mT5 encoder is pre-trained using a denoising objective (masked span prediction) on multilingual data, including Bengali.

Similar to BERT, given an input sequence of tokens in equation (4),

$$X = (x_1, x_2, \dots, x_N), \tag{4}$$

the T5 encoder computes a sequence of final hidden states shows in euation (5),

$$H = (h_1, h_2, \dots, h_N): \quad H = T5\_Encoder(X) \tag{5}$$

where $h_i \in \mathbb{R}^{D'}$ is the D'-dimensional contextualized embedding for the i-th token (typically 768 or 1024 for a base mT5 model).

The code applies mean pooling over all token hidden states to derive the sentence embedding. Crucially, it uses the attention_mask to ignore padding tokens in the average, a technique called masked mean pooling.

The hidden states H are multiplied element-wise with the attention mask M to zero out the hidden states of padding tokens. The corresponding equation shown in equation (6).

$$Sum\_Hidden = \sum_{i=1}^{N} (h_i \odot M_i) \tag{6}$$

where $\odot$ denotes element-wise multiplication, and M_i is the D'-dimensional attention mask value (0 for padding, 1 for real tokens) broadcast across the hidden dimension. In the code, this is

$$hidden\_states * attention\_mask.$$

The total number of non-padding tokens is computed by summing the attention mask which equation shown in equation (7).

$$Length = \sum_{i=1}^{N} M_i \tag{7}$$

In the code, this is

$$attention\_mask.sum(dim = 1) \tag{8}$$

(clamped to prevent division by zero).

The sentence embedding E_{T5} is the average of the non-padding hidden states. The corresponding equation shown in equation (9).

$$E_{T5} = \frac{Sum\_Hidden}{Length} \tag{9}$$

This embedding E_{T5} is the output of the get_t5_embeddings function.

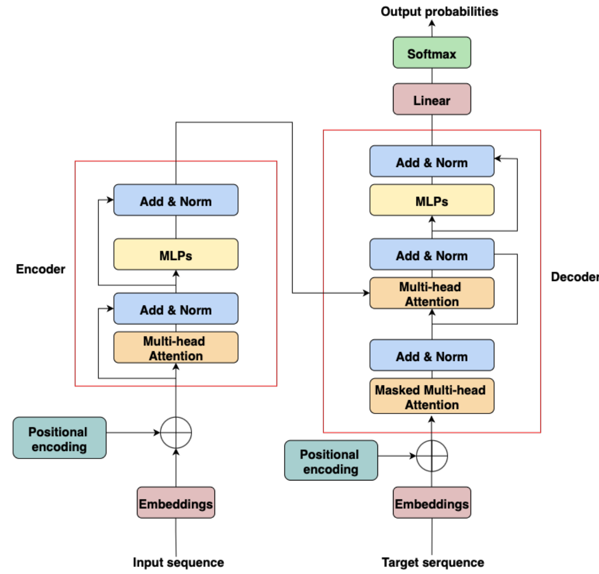

Figure 14. Bangla T5 Model Architecture

ANN: Artificial Neural Networks (ANNs) are computational models inspired by the structure and function of the human brain. They consist of interconnected layers of nodes (neurons) that process inputs through weighted connections and activation functions to learn complex patterns. ANNs are widely applied in both classification and regression tasks. The corresponding equation shown in equation (10).

$$y = f\left(\sum_i w_i x_i + b\right) \tag{10}$$

x_i = input features, w_i = corresponding weights, b = bias term, $f(\cdot)$ = activation function (e.g., sigmoid, ReLU), y = neuron's output.

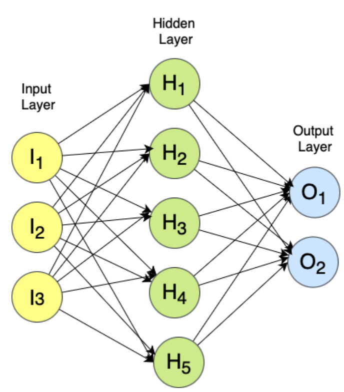

Figure 15. ANN Model Architecture

TransNeuNet: We presented TransNeuNet, a hybrid deep learning model in our work. The architecture of this model shown in Figure 16. Two modern language models, Bangla BERT and Bangla T5 have been worked into this model's architecture. Both models produce their respective representations when a sentence is given as input. While Bangla T5 generates embeddings by applying mean pooling over the encoder hidden states and Bangla BERT creates embeddings based on token representations.

The outputs of both models then pass through separate, dense and fully connected branches. Each branch reduces the embeddings to 256-dimensional feature vectors by applying a linear transformation, ReLU activation, and dropout. We get two 256-dimensional vectors from these embeddings. Finally, a single 512-dimensional combined vector is created by connecting the two feature vectors. Equation (11) shows the corresponding equation.

$$h = [h_B \,||\, h_T] \in \mathbb{R}^{512}. \tag{11}$$

This combined representation is passed through another fully connected block. First, it goes through a dense hidden layer (512 $\rightarrow$ 128) with ReLU activation and dropout, shown in equation (12).

$$z = ReLU(W_1 h + b_1), \tag{12}$$

where $W_1 \in \mathbb{R}^{512\times128}$.

Finally, the output probabilities for each class are generated by the last layer using a softmax function. Equation (13) shows the corresponding equation.

$$\hat{y} = Softmax(W_2 z + b_2), \tag{13}$$

where $W_2 \in \mathbb{R}^{128\times C}$ and $\hat{y} \in \mathbb{R}^C$ is the predicted class probability distribution.

The model is trained with a cross-entropy loss that compares the actual labels with the predicted class probabilities, Equation (14) shows the equation. The Adam optimizer helps to update the weights during training.

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{14}$$

where y is the one-hot encoded ground truth vector.

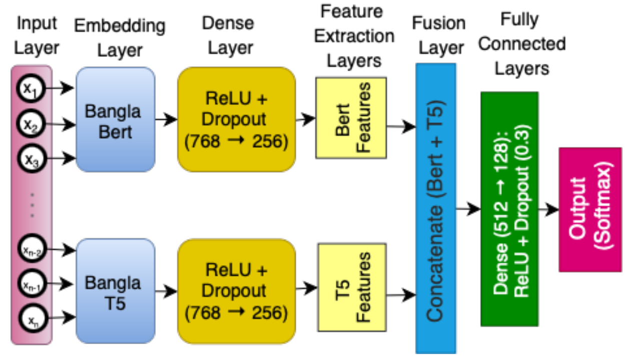

Figure 16. TransNeuNet Model Architecture

F. *Experimental Setup*

We used the Kaggle platform for our experiments. It provides a free online environment for machine learning projects. This platform provides GPU support for fast training and model testing. In our work, we used an NVIDIA Tesla P100 GPU with 16GB of RAM and about 13GB of GPU memory. We also used the Python language and many Python libraries such as PyTorch, TensorFlow, sckiti-learn, NumPy, pandas, etc. to complete our experiments.

4. Result and Discussion

In this section, we present the experimental results for the TransNeuNet model that we proposed. We use accuracy, precision, recall, and F1-score to compare the performance of TransNeuNet with other models. We analyze the loss curve and ROC curve of our proposed model and interpret the results using LIME. Finally, we discuss some limitations found in our study.

A. *Performance Metrics*

Accuracy: Accuracy measures the proportion of correctly classified instances (both positive and negative) among all instances in the dataset which is show in the equation (15).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

Where, TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives

Precision: Precision is the fraction of correctly predicted positive observations among all observations predicted as positive. The equation is shown in Equ-(16). Precision is often called the Positive Predictive Value.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

It answers the question: *"Of all instances predicted as positive, how many are actually positive?"*

Recall: Recall is the fraction of correctly predicted positive observations among all actual positive observations. It is often called Sensitivity or True Positive Rate. The equation is shown in Equ-(17).

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

It answers the question: *"Of all actual positives, how many did we correctly identify?"*

F1 Score : F1 Score is the average of Precision and Recall, using a method that gives more weight to smaller values. It helps balance Precision and Recall, especially when the dataset is Imbalance. The equation is shown in Equ-(18).

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{18}$$

B. Model Performance

Our model TransNewNet has shown good performance in news classification. The evaluation showed balanced reliable predictions in all categories, with an average precision, recall and F1-score of around 0.93 and an overall accuracy of 93.53%. We also train some traditional machine learning and deep learning models using Bert Embeddings. Random forest achieves 83.79% accuracy, Decision Tree achieves the lowest accuracy with 63.78%. SVM with 87.47% is better than all machine learning models but less than our TransNeuNet model. BiLSTM, a deep learning model shows better performance than all traditional machine learning models with 88.35% accuracy. Without combining bert and T5, ANN achieves 88.89% accuracy and CNN achieves 77.33%. But after combining Bert and T5, our transNeuNet model shows very good performance. TransNeuNet shows 93.53% accuracy which is higher than any other model. This shows how well transformer-based embedding and hybrid architectures work for accurate news classification. Overall, TransNeuNet ensures consistent performance and high accuracy across different types of news categories. Table II shows the comparison clearly.

TABLE II. Performance Comparison of Different Models

Model	Accuracy	Precision	Recall	F1 Score
CNN	0.7733	0.7781	0.7733	0.7740
BiLSTM	0.8835	0.8850	0.8835	0.8838
ANN	0.8889	0.8902	0.8889	0.8890
SVM	0.8747	0.8757	0.8747	0.8750
Decision Tree	0.6378	0.6383	0.6378	0.6379
Random Forest	0.8379	0.8429	0.8379	0.8391
TransNeuNet (Proposed)	0.9353	0.9359	0.9353	0.9354

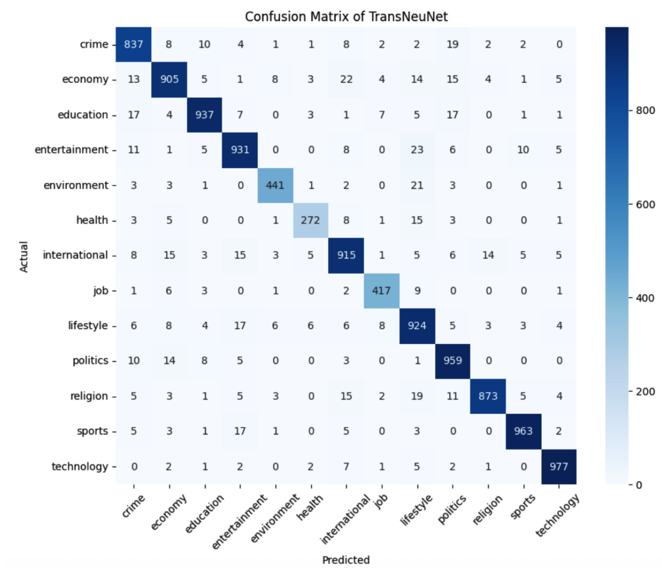

Figure 17. Confusion Matrix of TransNeuNet

In figure 17 confusion matrix shows the performance of the TransNeuNet model in 13 news categories. High dominance along the diagonal indicates that the model performed quite well. Some articles were not correctly identified. But overall the model is performing well.

TABLE III. Classification Report for **TransNeuNet**

Category	Precision	Recall	F1-Score	Support
Crime	0.91	0.93	0.92	896
Economy	0.93	0.91	0.92	1000
Education	0.96	0.94	0.95	1000
Entertainment	0.93	0.93	0.93	1000
Environment	0.95	0.93	0.94	476
Health	0.93	0.88	0.90	309
International	0.91	0.92	0.91	1000
Job	0.94	0.95	0.94	440
Lifestyle	0.88	0.92	0.90	1000
Politics	0.92	0.96	0.94	1000
Religion	0.97	0.92	0.95	946
Sports	0.97	0.96	0.97	1000
Technology	0.97	0.98	0.97	1000

According to TransNeuNet's classification report shown in Table III, F1-score, precision and recall are mostly above 0.90. The performance scores for technology, sports, and religion are the highest, but the scores for health and lifestyle are slightly lower. This indicates some difficulties in these areas. Overall, the model performance is excellent and works well for multi class classification.

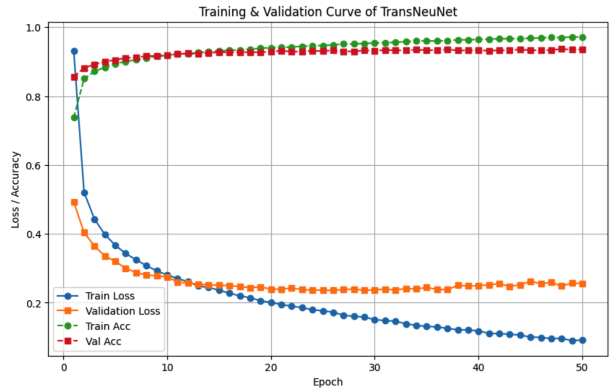

Figure 18. Loss Curve of TransNeuNet

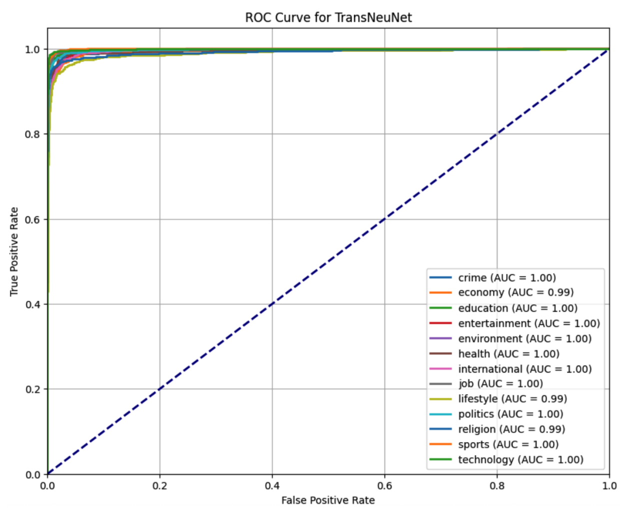

Figure 19. ROC Curve of TransformNeuNet

In figure 19, The ROC curve shows that the TransNeuNet model performs extremely well for news classification. Most classes have AUCs of 1.00 and only a few of 0.99. This means that the model can perfectly distinguish between categories with accuracy.

C. Model Interpretation

We use Local Interpretable Model-agnostic Explanations (LIME) to explain our results.

`Input Text:` জাতীয় নাগরিক পার্টির (এনসিপি) উত্তরাঞ্চলের মুখ্য সংগঠক সারজিস আলম বলেছেন, ‘আমরা স্পষ্ট করে বলব, বাং
লাদেশের মানুষ ওই একটা প্রশ্নে আপসহীন, শেখ হাসিনা খুনি–বাংলাদেশের দেয়ালে দেয়ালে এই লেখাটা রক্ত দিয়ে লেখা হয়ে গেছে। যাঁরা
এখন অন্তর্বর্তী সরকারের সুবিধাভোগী, তাঁরা তাঁদের জায়গা থেকে এইটা ভুলে যেতে পারেন, কিন্তু যে মা তাঁর ছেলের লাশ নিজের হাতে
ছুঁয়েছেন, ওই মা কিন্তু এটা ভুলতে পারবেন না। ওই মা যত দিন না ভুলবেন, আমরা যাঁরা এই অভ্যুত্থানে রাজপথে লড়াই করেছিলাম,
আমরা এটা ভুলব না, ভুলতে দেবও না।’ পঞ্চগড়ের আটোয়ারী উপজেলার বিভিন্ন পূজামণ্ডপ পরিদর্শনের সময় বুধবার বিকেলে মালিগাঁও
তেঁতুলতলা দুর্গামন্দিরে সাংবাদিকদের এ কথা বলেন সারজিস। তিনি বলেন, ‘আমরা স্পষ্ট করে বলি এই বাংলাদেশে আওয়ামী লীগের
আর কোনো রাজনীতি চলবে না।’ এনসিপির এই নেতা বলেন, ‘অভ্যুত্থান–পরবর্তী সময়ে যখন আমাদের আবার আওয়ামী লীগ নিষিদ্ধে
র দাবিতে রাজপথে নামতে হয়, এটাই হচ্ছে এই অন্তর্বর্তীকালীন সরকারের প্রথম ব্যর্থতা। দ্বিতীয়ত, যখন মানুষ আবার রাজপথে নামল
তাঁদের জায়গা থেকে এবং নামার পরে মানুষ রাজপথে দিনের পর দিন আন্দোলন করল এবং যমুনা ঘেরাওয়ের পর্যায় পর্যন্ত গেল, তখন
তারা এই রাজনৈতিক কার্যক্রম নিষিদ্ধের যে ধাপ সেই ধাপে গেল।’ প্রধান উপদেষ্টা অধ্যাপক মুহাম্মদ ইউনূসের উদ্দেশে সারজিস আলম ব
লেন, ‘তিনি যদি এখন বিভিন্ন কারণে আন্তর্জাতিক মহলে বাংলাদেশের মানুষের এত ত্যাগ–তিতিক্ষা সবকিছু ভুলে গিয়ে, বাংলাদেশের মানু
ষের এত রক্ত বিসর্জন–প্রাণ বিসর্জন ভুলে গিয়ে নিজের ইমেজের চেক অ্যান্ড ব্যালেন্স করার জন্য সুশীল সাজতে শুরু করেন এবং এই অ
ভ্যুত্থানের মতো একটা বড় ঘটনা ভুলে গিয়ে তিনি যদি এই স্টেটমেন্ট আসলে দিয়ে থাকেন যে আওয়ামী লীগের কার্যক্রম আবার সচল ক
রা হতে পারে, তাহলে আমরা শুধু একটা কথাই বলি, বাংলাদেশের মানুষ ছাড় দিতে পারে, কিন্তু ছেড়ে দেবে না।

Figure 20. Input Text

```
Predicted Class: politics
Class Probabilities:
crime: 0.0000
economy: 0.0000
education: 0.0000
entertainment: 0.0000
environment: 0.0000
health: 0.0000
international: 0.0000
job: 0.0000
lifestyle: 0.0000
politics: 0.9999
religion: 0.0000
sports: 0.0000
technology: 0.0000
```

Figure 21. Class Probabilities

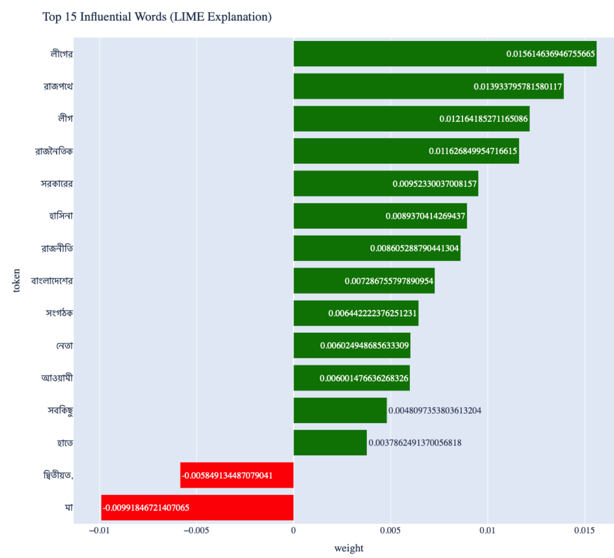

Figure 22. LIME Explanation Plot

Essentially, this tool helps us understand why the machine learning model made a particular prediction. In our system, we input a news article for analysis. The model then analyzes the article and predicts the results based on its training. After prediction, LIME is used to explain the decision by highlighting the most important words or features that influence the model. This process helps make the model's predictions more understandable and transparent. In Figure 20, we show an example input text. The predictions for this class are shown in Figure 21 and the interpretation of LIME is shown in Figure 22. Where, the green bar point to the strongest positive contribution and the red bar indicates the negative contribution.

D. Limitation

Although our model shows strong performance and we have done some great work, it has some limitations. We have not implemented any web application. Therefore, the user will not be able to use this model comfortably. We use two transformer model in our work, Bangla Bert and

Bangla T5. So, this model requires a lot of computational power and time. We used only 13 categories, which is not much compared to real-world scenarios. In real life, there are many more categories. We hoped to achieve more than 95% accuracy, but we failed to achieve this goal.

5. Conclusions and Future Work

In this study, news articles are divided into 13 categories. We developed our own dataset to support this work by using BeautifulSoup for web scraping, which allowed us to gather a lot of Bengali news articles and increase the number of categories. Our dataset is the most complete for Bengali news classification because many earlier studies did not include 12 or 13 categories. We build a transformer and neural network based model called TransNeuNet. We also apply different types of machine learning and deep learning models to compare the results of our built in model. Compared to all models, the experimental results show that combining Bangla BERT and Bangla T5 embeddings improves significantly the classification performance. Our model is more powerful for real-world applications because it ensures both high accuracy and interpretability. This model classifies articles and explains why a particular category was predicted using LIME explanation.

In future research, we plan to improve our work in several ways. First, we will add more data for different categories and make the model more diverse. Second, we plan to create a web application. By this users will be able to use our system easily and comfortably. Although we currently use LIME for interpretation, in the future we hope to include more advanced interpretation methods to help users understand the model results more clearly. Finally, we will focus on increasing the accuracy of our model. In terms of additional applications, we plan to apply our model to other related tasks such as headline generation, text summarization, and sentiment analysis. We also plan to add a spam detection feature to filter out spam news.

References

[1] M. A. Z. K. Md. Mahbubur Rahman, "Bangla news classification using graph convolutional networks," *Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–6, 2021.

[2] "Bengali language," *Wikipedia, The Free Encyclopedia*.

[3] M. I. H. Shakil Rana, "Newsnet: A comprehensive neural network hybrid model for efficient bangla news categorization," *Proceedings of the 15th International Conference on Computing Communication Networking Technology (ICCCNT)*, pp. 522–527, 2024.

[4] P. T. Aysha Gazi Mouri, "An empirical study on bengali news headline categorization leveraging different machine learning techniques," p. 312–317, 2022.

[5] S. S. Mohammad Rabib Hossain, "Different machine learning based approaches of baseline and deep learning models for bengali news categorization," *International Journal of Computer Applications*, vol. 176, pp. 10–16, 2020.

[6] N. C. Amran Hossain, "Bangla news headline categorization," *International Journal of Education and Management Engineering*, vol. 11, pp. 39–48, 2021.

[7] Amran0917, "Bangla-news-headlines-categorization," *GitHub repository*, 2019.

[8] A. K. M. M. Sharun Akter Khushbu, "Neural network based bengali news headline multi classification system: Selection of features describes comparative performance," *Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, 2020.

[9] S. S. Ruhul Amin Nabila, "Multiclass classification for bangla news tags with parallel cnn using word level data augmentation," *Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 174–177, 2020.

[10] R. K. Sharmin Yeasmin1, "Multi-category bangla news classification using machine learning classifiers and multi-layer dense neural network," *International Journal of Advanced Computer Science and Applications*, vol. 12, pp. 757–764, 2021.

[11] E. M. E. Pallab Chowdhury, "Bangla news classification using glove vectorization, lstm, and cnn," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 95, p. 723–731, 2021.

[12] S. K. M. Shagoto Rahman1, "An empirical study of machine learning-based bangla news classification methods," p. 1–6, 2021.

[13] K. S. Amartya Roy, "Bengali text classification: A new multi-class dataset and performance evaluation of machine learning and deep learning models," 2023.

[14] F. A. Istiak Ahmad, "Machine and deep learning methods with manual and automatic labelling for news classification in bangla language," 2022.

[15] S. A. Md. Rafi-Ur-Rashid, "Feature extraction using deep generative models for bangla text classification on a new comprehensive dataset," 2023.

[16] M. F. Md Fahim Sikder, "Explainable bengali multiclass news classification," *IEEE Access*, vol. 11, p. 12345–12356, 2023.

[17] M. A. U. H. Samrat Alam, "Bengali text categorization based on deep hybrid cnn–lstm network with word embedding," *IEEE Access*, vol. 11, p. 98765–98775, 2023.

[18] S. A. I. Mohammad Kamrul Hasan, "Classifying bengali newspaper headlines with advanced deep learning models: Lstm, bi-lstm, and bi-gru approaches," *Asian Journal of Research in Computer Science*, vol. 16, p. 372–388, 2023.

[19] M. A. Ovi Chowdhury, "Bengali news headline categorization: A comprehensive analysis of machine learning and deep learning approach," *BAIUST Academic Journal*, vol. 4(1), pp. 26–44, 2023.

[20] Z. H. K. Shafayat Bin Shabbir Mugdha, "Accurate prediction of bangla text article categorization by utilizing novel bangla stemmer," *International Journal of Automation and Smart Technology*, vol. 14(1), pp. 1–7, 2024.

[21] A. R. I. T. Hossain, "Banglanewsclassifier: A machine learning

approach for news classification in bangla newspapers using hybrid stacking classifiers," *PLoS One*, vol. 20(6), p. e0321291, 2025.

[22] A. J. M. J. R. R. Chowdhury, "A comparative study on different machine learning approaches for categorizing bangla documents," *International Journal of Computer Applications*, vol. 186(61), pp. 32–39, 2025.

[23] Z. Al Nazi Nabil, "Bangla newspaper dataset," 2020.