Leveraging NLP and ML for Sentiment Analysis of In- formal Bangla Text: A Social Media Comment Analysis Tool for Page Owners

Leveraging NLP and ML for Sentiment Analysis of Informal Bangla Text: A Social Media Comment Analysis Tool for Page Owners

Fahima Hossain[1[0000-0001-8743-8547]], Palash Saha[2[0009-0009-2132-004X]], Ismail Hossain Adnan[3[0009-0002-4151-0296]] and Mashiur Rahman[4[0009-0004-2257-968X]]

[1, 2, 3, 4] Hamdard University Bangladesh, Dhaka, Bangladesh

Abstract. The massive popularity of social media platforms and the extensive engagement of people on these platforms have made it easier to express opinions or sentiments through posts, comments, and messages, generating a significant amount of digital text data every day. Sentiment analysis is the process of summarizing and categorizing opinions expressed on social media. Detecting sentiment in the Bangla language has become increasingly important due to a notable lack of sentiment detection capabilities for informal Bangla text. This research aims to develop an advanced sentiment detection model for analyzing sentiment in informal, noisy Bangla text collected from social media comments. Our major contribution involves using web scraping to gather comments from different sites, resulting in a large dataset of more than 3,000 comments. This approach minimizes bias compared to single-source data and captures sentiment across a wide range of topics. To ensure accuracy and dependability, we collaborated with psychology students to carefully classify the gathered comments by sentiment polarity. We built a customized deep-learning strategy using LSTM and optimized several LSTM model hyperparameters, including learning rate, batch size, number of epochs, dropout rate, and number of LSTM layers. Our customized deep-learning strategy proved beneficial, as the LSTM model outperformed other models on informal Bangla text. The research achieved 80.3% accuracy for the Bangla informal dataset and 96.9% accuracy for the formally written Bangla text dataset, underscoring the efficiency of the proposed approach. Finally, a user-friendly interface was built to capture comments from social media and classify their sentiments.

Keywords: Informal Bangla Texts, Social Media Comments, LSTM Model, Hyperparameter Tuning, User Interface

1 Introduction

The digital age has seen a surge in Bangla online usage, with its application for communication and information exchange flourishing across social media platforms and news outlets [1]. Sentiment analysis, or opinion mining, is a computational technique that automatically detects emotions and opinions in text, offering significant potential for understanding public sentiment, gauging customer satisfaction, and extracting insights from Bangla user-generated content [2]. This technique evaluates textual data to determine if the sentiment expressed is positive, negative, or neutral. This subfield of natural language processing (NLP) has become increasingly important due to its wide applications in marketing, social media monitoring, customer feedback analysis, and public opinion research [3]. Particularly, sentiment analysis of Bangla text presents unique challenges and opportunities given Bangla's status as a major global language spoken predominantly in Bangladesh and parts of India [4]. However, applying sentiment analysis to Bangla text presents unique challenges compared to more standardized languages. Unlike formal written Bangla, online Bangla text is often "noisy"—characterized by informalities, slang, and mixed dialects due to the language's unique linguistic structure, including compound word formations, complex verb conjugations, and a wide range of synonyms and idiomatic expressions [5, 6]. These linguistic nuances can significantly impact the accuracy of sentiment analysis tools designed for formal languages. Additionally, the scarcity of resources tailored for Bangla, such as annotated datasets and sentiment lexicons, further complicates the development of effective sentiment analysis tools, necessitating the creation of customized solutions [5]. This paper explores the significance, challenges, methodologies, and practical applications of analyzing sentiments in Bangla text, highlighting the complexities and potential within this linguistic context.

Despite these challenges, recent advancements in sentiment analysis have spurred innovative techniques tailored for Bangla text. Researchers have explored various methodologies, including lexicon-based approaches leveraging sentiment dictionaries adapted for Bangla, supervised machine learning algorithms trained on annotated datasets, and hybrid techniques combining multiple strategies to enhance accuracy.

The system aims to develop robust methods for sentiment analysis of noisy Bangla text, which includes informal language, slang, and grammatical errors. This is crucial for understanding public opinion and user experience from social media comments and news articles. The project will build a diverse dataset that includes "noisy" Bangla text with dialects, grammatical errors, and spelling mistakes. Comments on social media posts and news articles across different domains will be collected to reflect real-world social media usage. It will utilize advanced NLP techniques to accurately classify and analyze sentiment, providing valuable insights for social media monitoring, brand reputation management, and trend detection. The benefits include enhanced efficiency in social media monitoring, early detection of public sentiment trends, and improved brand reputation through timely interventions. This system minimizes the need for manual monitoring, optimizes resource use, and supports informed decision-making for effective social media strategies.

This work presents a novel approach to sentiment analysis on informal Bangla text by leveraging web scraping and exploring the effectiveness of various techniques on a diverse dataset. Our key contributions are as follows:

- Concentrated on specific social media platforms, scraped relevant Bangla comments, and had a qualified psychologist manually label sentiments, ensuring accuracy and minimising errors.
- Fine-tuned model parameters to improve the LSTM model's validation accuracy and reach peak performance.

- Saved trained models in multiple formats for easy future deployment and developed an intuitive interface to fetch and classify social media comments, enhancing the tool's accessibility and usability.

The rest of the paper is organized as follows: Section 2 provides a relevant literature review; Section 3 details the data and outlines the proposed system, presenting a straightforward explanation of its features and functionalities for clear comprehension of the proposed solution. Section 4 involves evaluating the outcomes and performance of the developed sentiment analysis system for noisy Bangla text. The final section addresses the unique challenges posed by the noisy nature of online Bangla text, including informalities, slang, and grammatical errors through the implementation of advanced Natural Language Processing techniques, and discusses future work.

2 Related Work

Numerous studies have explored sentiment analysis in English, but Bangla sentiment analysis has been limited due to a lack of publicly available annotated datasets and the language's complex grammar. However, the rising use of Bangla texts on social media and news portals has spurred interest in this field. Researchers are now creating their own datasets and using natural language processing techniques to preprocess data for machine learning and deep learning models.

Kazi Toufique Elahi et al. [7] proposed a multi-modal approach using ResNet50 and BanglishBERT to interpret meme sentiments, particularly for error-prone texts in low-resource languages like Bengali. They created a new dataset combining text and image data labeled with sentiment and identified ten types of text errors. Despite preprocessing efforts, noise reduction methods did not significantly improve sentiment analysis accuracy, and the model struggled with neutral memes due to dataset imbalance and similar meme templates.

Rashedul Amin Tuhin et al. [8] developed a model to detect six emotions using the Naïve Bayes Classification Algorithm and a topical approach. They manually created a dataset of 7,500 Bangla sentences, cleared of special characters and punctuation, and assigned emotions based on meaning. The topical approach, which used tf-idf to calculate emotional values, outperformed the Naïve Bayes classifier. The manual labeling process was time-consuming and potentially biased.

Shaika Chowdhury et al. [9] tackled sentiment analysis for Bangla microblog posts, classifying tweets as either negative or positive. They collected 1,300 bilingual tweets, preprocessed them using tokenization, normalization, and POS tagging, and created a Bangla sentiment lexicon. Using a semi-supervised bootstrapping approach, they extracted features and employed SVM and Maximum Entropy classifiers. The SVM with unigrams and emoticons achieved better accuracy, but the model's reliance on emoticons limited its applicability.

Hasmot Ali et al. [10] introduced the "BanglaSenti" lexicon-based dataset, which includes 61,582 Bangla words translated from English SentiWordNet 3.0. This dataset focuses on single words and excludes repetitive or meaningless terms. The lack of machine learning model validation reduces its practical usability and performance.

Md. Ferdous Wahid et al. [11] proposed using an RNN with an LSTM model to identify cricket-related sentiments in Bangla texts. They collected data from social media, categorized sentiments, and preprocessed the text by removing stopwords, links, URLs, and punctuation. They used the Skip-gram model for vectorization and fed the embedding matrix into the RNN. Various epoch and batch size combinations were tested to optimize prediction performance.
Md. Atikur Rahman et al. [12] conducted aspect-based sentiment analysis using two publicly available datasets containing user comments on cricket and restaurant reviews. They preprocessed the data by removing punctuations, stopwords, and digits, performed tokenization, and created a feature matrix using TF-IDF. The model, trained with SVM, Random Forest, and KNN algorithms, achieved low recall and F1-scores.
Khondoker Ittehadul Islam et al. [1] constructed an annotated sentiment analysis dataset from public comments on social media, covering various domains. Comments were labeled as positive, negative, or neutral by three individuals, and final labels were assigned based on majority voting. They developed a classification system using BiLSTM and pretrained language models like BERT. The imbalanced dataset, containing mixed dialects and grammatical errors, posed challenges for preprocessing techniques.
Asif Hassan et al. [13] introduced the BRBT dataset, which includes 9,337 text samples in both Bangla and Romanized Bangla. The dataset was annotated with sentiments by native speakers and preprocessed to remove emoticons, hashtags, and proper nouns. LSTM models were used for classification, and pre-training experiments assessed model performance.
Zishan Ahmed et al. [14] performed binary and multiclass classification using a curated dataset from the "Daraz" platform. The preprocessing involved removing punctuation, numbers, emojis, and stopwords, followed by stemming. Word2vec embeddings were used to convert text into vectors, and RNN-based transfer learning models like GRU, LSTM, and Bangla-BERT were employed for training. They used a softmax activation function, sparse categorical cross-entropy loss function, and Adam optimizer.
Priya Das et al. [15] applied sentiment analysis to categorize Bengali texts into five classes: negative, near negative, neutral, near positive, and positive. They used the ABSA dataset from the BBC Bangla News Portal, preprocessed the text to remove symbols, numbers, and punctuation, and filtered out stopwords. Feature extraction techniques like tokenization, n-grams, and TF-IDF were employed. Various classifiers, including Random Forest, Logistic Regression, and SVM, were used for training. They did not perform stemming or address data class imbalance.

3 Proposed Methodology

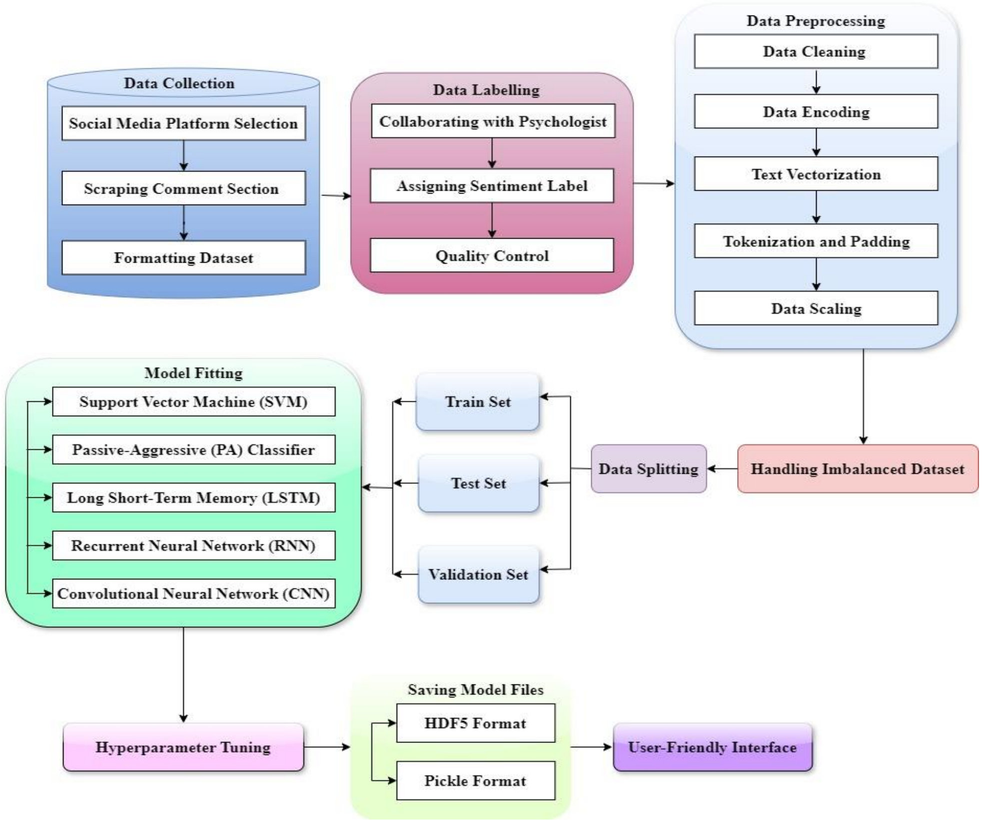

Fig. 1. Block Diagram of the proposed methodology.

The proposed methodology consists of nine main steps: data collection, data labeling, data preprocessing, handling imbalanced datasets, data splitting, model fitting, hyperparameter tuning, saving model files, and finally, creating a user-friendly interface. The block diagram of the proposed methodology is illustrated in Figure 1.

3.1 Data Collection

Here's a breakdown of the data collection process:

Social Media Platform Selection

We targeted specific social media platforms relevant to our goals. These platforms likely host a significant amount of Bangla text data suitable for sentiment analysis

Scraping Comment Section

Using the "Instant Data Scraper" Chrome extension, we extracted comments from the chosen social media platforms. The extension allows users to define specific scraping criteria, enabling us to focus on comment sections relevant to our research question.

Formatting Dataset

The scraped data was initially saved in a CSV format for ease of storage and manipulation. After scraping comments from multiple sources, we processed the data further to ensure its suitability for the analysis. This involved selecting Relevant Column: We identified the comment text column within the scraped CSV files and extracted only that information. Other irrelevant columns containing URLs, usernames, or timestamps were discarded. The final dataset comprised approximately 3,000 comment entries, providing a sufficient sample size for training and evaluating our sentiment analysis model. The sentiment labels are categorized into three classes: Positive, Negative, and Neutral. The comments reflect a range of informal Bangla text, including dialectal variations and grammatical inconsistencies, which are common in social media interactions.

Table 1. Number of comments per sentiment class.

Sentiment Class	Number of Comment
Positive	1,197 comments
Negative	1,213 comments
Neutral	739 comments

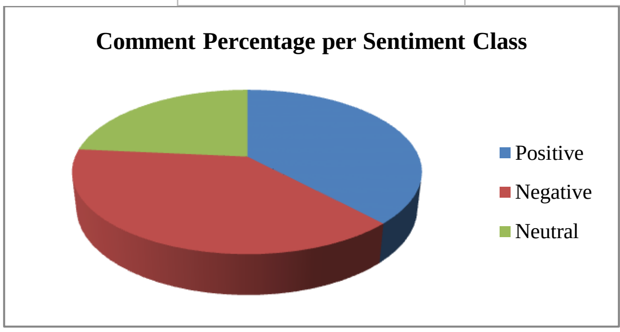

Fig. 2. Percentage distribution per sentiment class.

Table 2. Example of comments and sentiments.

Comment	Sentiment
নষ্টামি একটু বেশি এ হয়ে গেল না? বান্নাহ?	Negative
আপনাদের নিউজ ভরসা করার মত	Positive
এইদেশে কেবল একজনের বাবার খুনের বিচার হয়। আর কোন বাবা তার সন্তানের খুনের বিচার পায়না	Negative
এই খবরটি কোথা থেকে প্রকাশিত হয়েছে?	Neutral
সত্যিই অসাধারন, ভালো থাকুক ভালোবাসার মানুষেরা	Positive

3.2 Data Labeling

Sentiment analysis models require labeled data to learn the relationship between text and sentiment. Since our dataset originated from web scraping and contained only a "comment" column with raw text, we employed a manual labeling approach to assign sentiment labels. Here's how we addressed the data labeling task:

Collaboration with Psychologist

We recognized the importance of accurate sentiment labeling for effective model training. To achieve this, we collaborated with a qualified psychologist. Their expertise in understanding human emotions and sentiment expression ensured the labels assigned to the comments reflected their true sentiment. The psychologist reviewed each comment within the dataset and assigned a sentiment label. The labels used were:

- **Positive:** Comments expressing positive emotions, satisfaction, or approval.
- **Negative:** Comments expressing negative emotions, dissatisfaction, or disapproval.
- **Neutral:** Comments lacking a clear positive or negative sentiment, or those conveying factual information.

Assigning Sentiment Label

To ensure consistency and minimize labeling errors, we potentially implemented the double labeling technique. A subset of comments has been labeled independently by the psychologist and another individual familiar with sentiment analysis. Any discrepancies in labels have been resolved through discussion or by consulting a larger set of labeled examples.

Table 3. Most frequent word in comments for each sentiment class.

Positive	Neutral	Negative
Number of Words:15771 Number of Unique Words:4843	Number of Words:16029 Number of Unique Words:4472	Number of Words:7125 Number of Unique Words:2788
Most Frequent Words:	Most Frequent Words:	Most Frequent Words:
অনেক 267	না 200	না 416
ভালো 260	কি 157	হয় 379
না 185	কোন 80	আর 167
ভাই 168	এই 77	এই 160
সুন্দর 163	ভাই 63	কি 158
একটা 159	ছিল 59	করে 132
অসাধারণ 153	থেকে 48	ফালতু 132
আর 147	করে 48	একটা 126
খুব 146	হয় 48	নাই 91

Table 1 shows the number of data points in each sentiment class: 1,197 positive comments, 1,213 negative comments, and 739 neutral comments. Figure 2 presents a pie chart illustrating the percentage distribution of these sentiment classes. Table 2 provides sample comments along with their corresponding sentiments. Table 3 highlights the most frequent words in comments for each sentiment class.

Quality Control

A set of guidelines has been developed for the psychologists. These guidelines outline specific criteria for assigning each sentiment label, promoting consistency throughout the labeling process.

The participation of a psychologist in data labeling offered several advantages such as improved accuracy and domain-specific knowledge. Their expertise in human emotions led to more accurate and nuanced sentiment labeling compared to potentially using crowd-sourced labeling or automated methods. As the comments were related to a specific domain, the psychologist's understanding of that domain further enhanced the accuracy of sentiment labels.

3.3 Data Preprocessing

After collecting and labeling our Bangla social media comment dataset, we applied several preprocessing techniques to prepare the data for sentiment analysis modeling. These techniques aimed to transform the text data into a format suitable for machine learning algorithms. Here's an overview of the preprocessing steps implemented:

Data Cleaning

The raw data obtained through web scraping can often contain inconsistencies and noise that can hinder the performance of sentiment analysis models. To ensure the quality and relevance of the text data, we implemented a data-cleaning process focusing on the following aspects:

Removing Non-Bangla Characters

The scraped comments might have included characters from languages other than Bangla due to the nature of social media interactions. We employed techniques to identify and remove these non-Bangla characters. This involves using regular expressions or libraries specifically designed for Bangla text processing.

Removing Unnecessary Punctuation

Social media comments often contain punctuation marks beyond standard Bangla punctuation, such as emojis, special symbols, or excessive use of punctuation for emphasis. We removed unnecessary punctuation marks that wouldn't significantly impact the sentiment of the text. This involves creating a list of punctuation marks to be excluded while retaining essential ones like commas, periods, and question marks.

Removing Multiple Spaces & Normalizing Numericals

Web scraping processes can sometimes introduce inconsistencies in spacing. We addressed this by removing occurrences of multiple consecutive spaces within the comments. Additionally, we normalized numerical data to ensure consistency. This involves converting numerical expressions written in different formats (e.g., "100" vs. "১০০") to a standard format.

By implementing these data cleaning steps, we aimed to improve text quality by removing irrelevant characters and unnecessary punctuation and reducing Noise. Removing irrelevant characters and unnecessary punctuation enhances the overall quality of the text data, making it easier for the sentiment analysis model to understand the sentiment conveyed. Eliminating inconsistencies in spacing and normalizing numerical data helps reduce noise within the data, leading to more accurate sentiment classification.

Data Encoding

Our sentiment labels (positive, negative, neutral) were originally categorical data. However, machine learning algorithms typically work better with numerical features. To address this, we performed label encoding. This process assigns a unique integer value to each category (e.g., positive = 1, negative = 2, neutral = 3). This allows the model to learn the relationships between the numerical labels and the corresponding sentiment conveyed in the comments.

Text Vectorization

Text data needs to be converted into a numerical representation that machine learning algorithms can understand. We employed a text vectorization technique called TF-IDF (Term Frequency-Inverse Document Frequency) for this purpose. TF-IDF considers both the frequency of a word within a comment (term frequency) and its overall importance across the entire dataset (inverse document frequency). This helps to capture the semantic meaning of the comments and identify words that are discriminative for different sentiment classes.

Tokenization and Padding

Text vectorization techniques like TF-IDF typically operate on sequences of words. However, comments within the dataset might vary in length. To ensure consistency, we applied tokenization. This process splits each comment text into individual words (tokens). Additionally, we have implemented padding techniques to adjust the length of each comment sequence to a fixed size. Padding with zeros or other appropriate values helps the model process comments of different lengths effectively.

Data Scaling

While not always necessary, we have employed standard scaling as a final preprocessing step. This technique scales the numerical features (potentially TF-IDF vectors) to have a mean of zero and a standard deviation of one. Standard scaling can improve the convergence of some machine learning algorithms and ensure that all features contribute equally during the training process.

3.4 Handling Imbalanced Dataset

Handling imbalanced datasets involves techniques like resampling (oversampling minority class, undersampling majority class), using performance metrics suitable for imbalanced data (F1-score, ROC-AUC), and employing specialized algorithms (SMOTE, ADASYN) to ensure the model's performance is robust across all classes.

SMOTE

To address the imbalance in our dataset, we employed the SMOTE (Synthetic Minority Over-sampling Technique) method. SMOTE generates synthetic samples for the minority classes, ensuring a balanced distribution of sentiment classes. The balanced dataset after applying SMOTE is illustrated in figure 3. This technique improved the model's ability to accurately detect sentiments across all categories, enhancing overall performance and robustness.

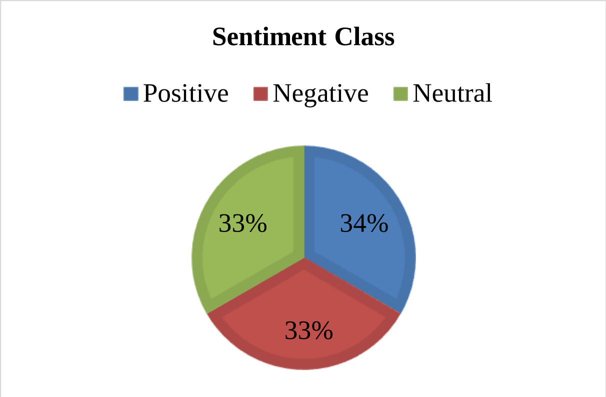

Fig. 3. The dataset after balancing using SMOTE.

3.5 Data Splitting

Splitting the dataset into training, testing, and validation sets is a crucial step in machine learning for sentiment analysis. This process ensures the model doesn't overfit the training data and allows for robust evaluation of its performance. In this study, we have implemented a common data-splitting strategy. The training Set is the largest portion of the dataset used to train the sentiment analysis model. Once the model is trained, it's evaluated on the unseen test set to assess how well the model generalizes to new, unseen data and provides an estimate of its real-world performance. The validation set is used to fine-tune hyperparameters (model configuration settings) during the training process. By monitoring performance on the validation set, we can identify the optimal hyperparameter configuration before using the final test set for evaluation.

3.6 Model Fitting

Our project explored various machine learning algorithms for sentiment analysis of Bangla social media comments. We compared the performance of the following models namely Support Vector Machine (SVM), Passive-Aggressive (PA) Classifier, Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN).

In this study, SVM would learn to separate comments with positive, negative, and neutral sentiment in a high-dimensional feature space derived from the processed text data (TF-IDF vectors). PA Classifier updates the model weights only on data points that are misclassified, making it computationally efficient for processing large amounts of text data. LSTMs effectively capture long-term dependencies within comments, which might be crucial for accurate sentiment classification, especially in Bangla where word order can be flexible. Similar to LSTMs, RNNs can struggle with long-term dependencies in sequences. We have included RNNs as a baseline model for comparison with LSTMs, which are specifically designed to address the vanishing gradient problem that hinders traditional RNNs. CNNs have also been adapted to extract features from local subsequences within the comments, potentially capturing sentiment-related patterns in the text.

3.7 Hyperparameter Tuning

By evaluating model performance on the validation set using these metrics, we identified the best hyperparameter configuration for each model. For instance, we might have experimented with different kernel types for SVMs or hidden layer sizes for LSTMs. The configuration leading to the best performance on the validation set was then chosen for final training and evaluation on the test set.

3.8 Saving Model Files

Preserving our trained sentiment analysis model is essential for future use, such as making predictions on new data or deploying the model in a production environment. In this study, we employed two common methods to save our trained models HDF5 format and Pickle format.

HDF5 Format

This format (model.h5) is widely used for saving deep learning models built with frameworks like Keras or TensorFlow. It efficiently stores the model architecture, including the layers, their configurations, and the weights (learned parameters) associated with each layer. Saving Process: For saving the model, we have used the model.save('model.h5') function.

Pickle Format

The pickle format (model.pkl) is a general-purpose serialization format in Python. It allows saving any Python object, including machine learning models, to a file that can be loaded later for reuse. We have used pickle.dump (model, open ('model.pkl', 'wb')) to save our trained model to a pickle file.

3.9 Interface

Interface design involves creating user-friendly interfaces that allow users to interact with ML or NLP models, visualize data, and interpret model outputs effectively and intuitively. We have designed our interface to fetch comments from a page's post in a social media such as Facebook and classify the comment's sentiment. We have designed three pages for performing this task: select a page, select a post and sentiment rating.

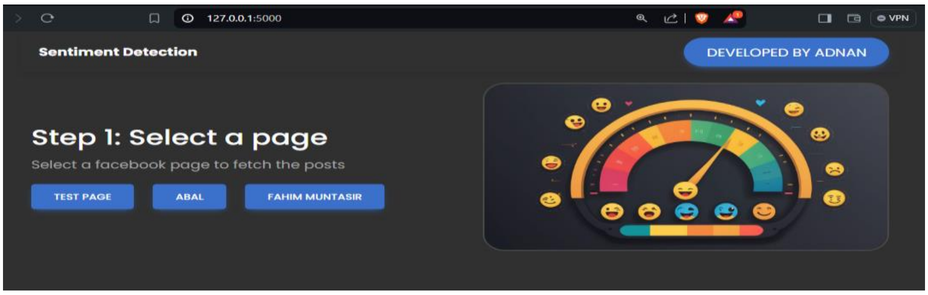

Fig. 4. Select the page for fetching posts and comments.

Step 1: Select a Page

• Interface Description: The initial interface presents a list of pages where the admin panel and editor list have been added as illustrated in figure 4.

• Technologies used: HTML, CSS, JavaScript, and Bootstrap.

• Action: Select a page from the list to proceed with the development using Graph API.

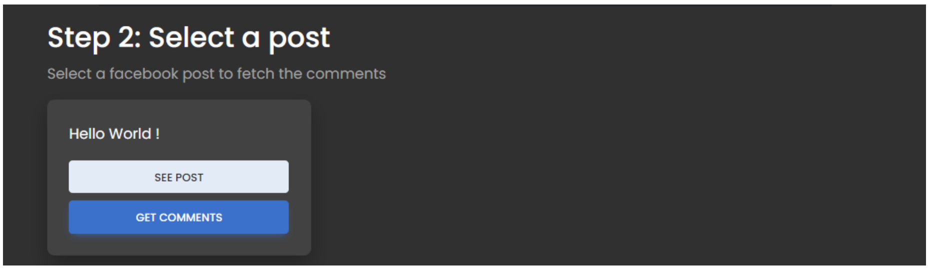

Fig. 5. Select "See post" for fetch post content & select "See comment" for fetch comment.

Step 2: Select a Post

• Interface Description: After selecting a page, the interface displays the posts from the chosen page as illustrated in figure 5.

• Action: Choose a specific post to retrieve comments.

• Data Collection: The Facebook Graph API fetches comments from the selected post.

Step 3: Sentiment Rating

Sentiment rating table of each comment

Comment	Sentiment
বইটি কবে প্রকাশিত হয়েছিলো?	0 - Neutral
ম্যাম শাড়ি পরে আপনাকে সুন্দর লাগছে	1 - Positive
জোর করে লোক হাসানো, ভাঁড়ামি	0 - Neutral
এদেশে একজন ছাড়া, কারও বাবার হত্যার বিচার হয় না।	2 - Negative
চিটাগাংইয়া সেরা	1 - Positive
নাটকে কিছু নোংরামী দেখানো হয়েছে	2 - Negative
ফালতু নাটক একটা	2 - Negative

Fig. 6. Comments with their Sentiment.

Step 3: Sentiment Rating

- Interface Description: The section is labeled "Step 3: Sentiment Rating."
- Subtitle: "Sentiment rating table of each comment" explains that the table shows sentiment analysis results.
- Table Structure: Created using HTML and designed using CSS and Bootstrap.
- Columns: "Comment" and "Sentiment".
- Comment Column: Displays various comments in Bengali script fetched using Graph API as illustrated in figure 6.
- Sentiment Column: Indicates the sentiment of each comment as neutral, positive, or negative using a machine learning model as illustrated in figure 6.

4 Experimental Results and Discussion

While the test set provides an unbiased estimate of real-world performance, the validation set can be a valuable tool during the model fitting process. We have used the validation set to monitor the model's performance and avoid overfitting. We have employed standard metrics commonly used for sentiment analysis tasks to assess model performance on the validation data.

Accuracy: This metric measures the proportion of comments correctly classified by the model (positive, negative, or neutral) [16].

Precision: Precision indicates the proportion of comments classified as a specific sentiment (e.g., positive) that are positive [16].

Recall: Recall reflects the proportion of actual positive comments that the model correctly classified as positive [16].

F1-Score: This metric combines precision and recall, providing a balanced view of the model's performance [16].

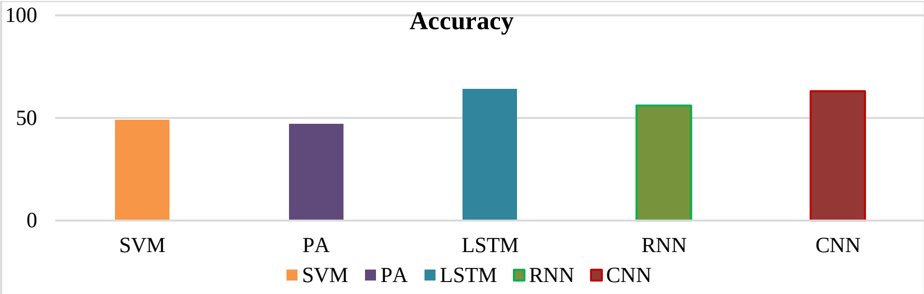

Fig. 7. Accuracy for the classification algorithms used for training on informal Bangla dataset.

Figure 7 illustrates the accuracy of various classification algorithms applied to an informal Bangla dataset without hyperparameter tuning. The LSTM algorithm achieved the highest accuracy at around 70%, followed by CNN at 65%, RNN at 60%, SVM at 55%, and PA at 50%.

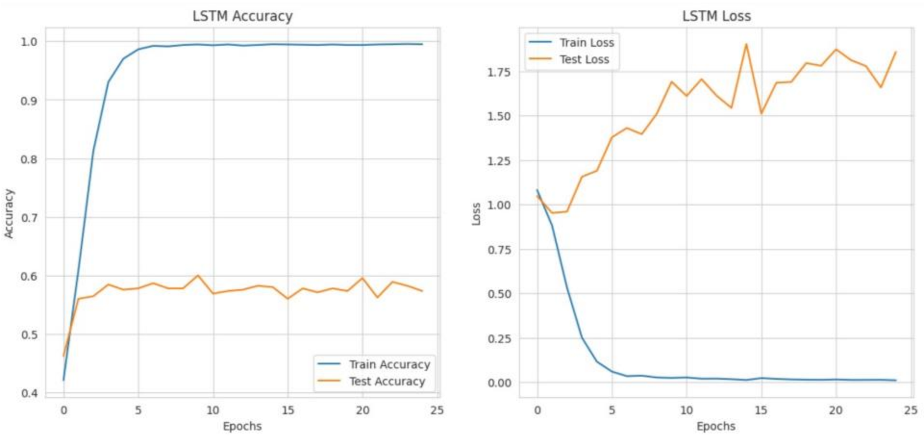

Fig. 8. Accuracy vs. loss graph for LSTM on informal Bangla dataset.

In figure 8, two lines show the accurate the machine is at guessing emotions. Higher line (blue line) indicates how well it performed on seen (or training) data. The lower (orange line) indicates how well it performed on unseen (or testing) data. The fact that the two lines are far apart suggests there might be a problem. The machine is too focused on memorizing the practice problems instead of learning the general rules of Bangla emotions. This could mean it wouldn't be very good at understanding emotions in writing it hadn't seen before.

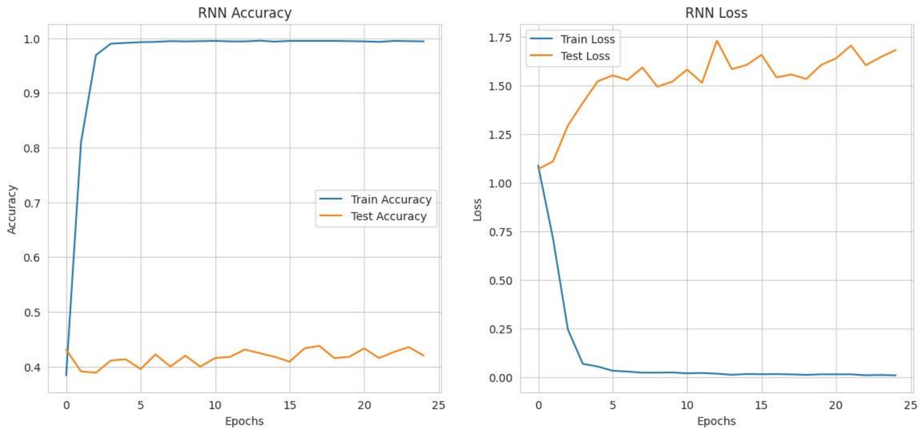

Fig. 9. Accuracy vs. loss graph for RNN on informal Bangla dataset.

Figure 9 shows the accuracy and loss of the RNN during training and testing. The training accuracy (blue line) quickly reaches nearly 100%, while the test accuracy (orange line) fluctuates around 40%, indicating poor performance on unseen data. The training loss (blue line) drops to nearly zero, but the test loss (orange line) remains high, even increasing slightly. These graphs highlight that the RNN is overfitting, as it performs well on training data but poorly on test data.

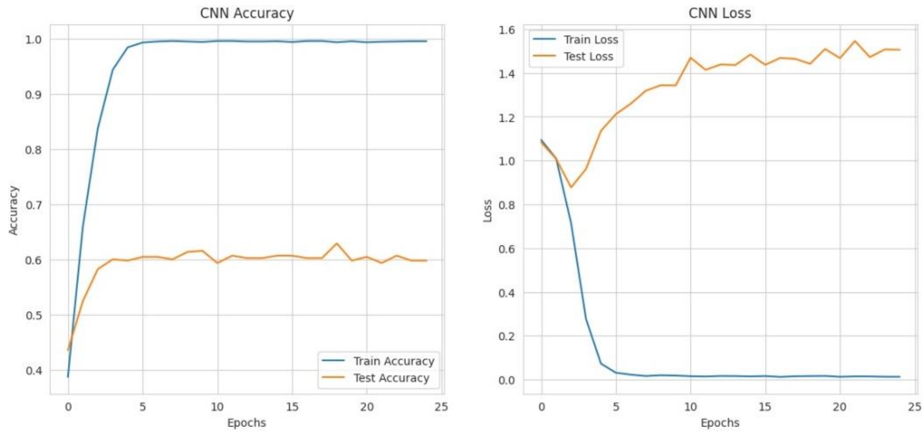

Fig. 10. Accuracy vs. loss graph for CNN on informal Bangla dataset.

Figure 10 displays the performance of a CNN over 25 epochs. The left graph shows accuracy, with training accuracy (blue line) quickly rising and leveling off near 1.0, while test accuracy (orange line) fluctuates around 0.6, indicating poor generalization. The right graph shows loss, with training loss (blue line) decreasing rapidly to near zero, while test loss (orange line) initially decreases but then increases, remaining high. These graphs indicate that the CNN is overfitting, performing well on training data but poorly on test data.

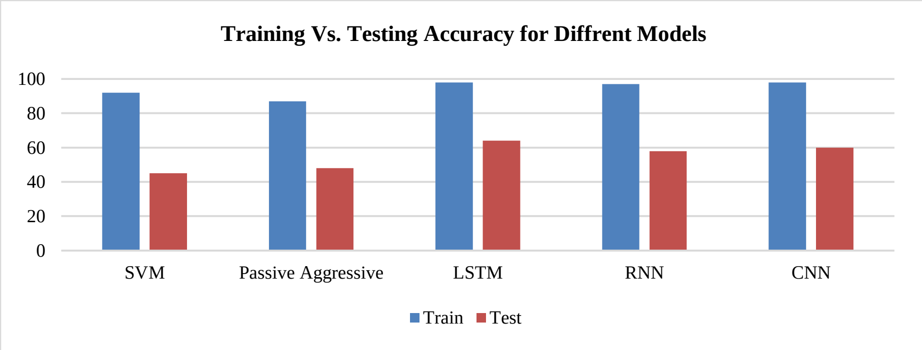

Fig. 11. Comparison of training vs. testing accuracy for different models on informal Bangla dataset.

Figure 11 illustrates a bar chart comparing the training and testing accuracy of five different machine learning models: SVM, passive-aggressive, LSTM, RNN, and CNN. The blue bars represent the training accuracy, while the orange bars show the testing accuracy for each model.
All the models have high training accuracy, close to 1.0, which indicates that they perform well on the training data. However, their testing accuracy is significantly lower, suggesting that these models may not generalize well to new, unseen data. This discrepancy between high training accuracy and lower testing accuracy indicates that the models might be overfitting the training data.

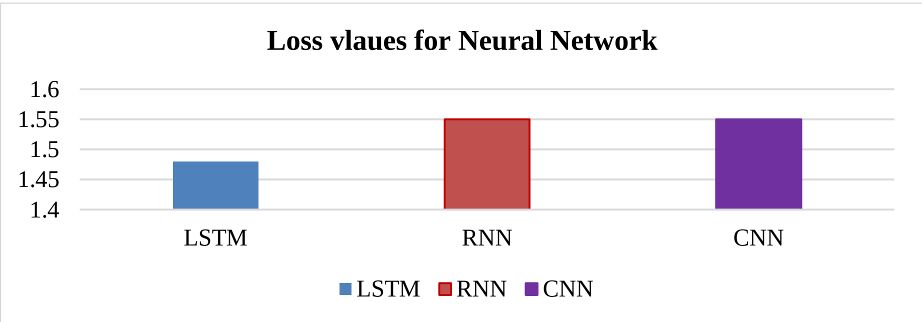

Fig. 12. Training vs. testing loss for different models on informal Bangla dataset.

Figure 12 shows the training vs. test loss for different neural network models applied to an informal Bangla dataset. The LSTM model achieved the lowest loss at around 1.45, while the RNN and CNN models had higher losses at approximately 1.54 and 1.53, respectively. This large difference between the training and testing losses indicates that these models are overfitting the training data. They perform well on the training data but fail to generalize to the testing data.
Overall, the SVM and Passive Aggressive models show better performance in terms of generalization, while the LSTM, RNN, and CNN models struggle with overfitting.

Table 4. Summary table for training and testing F1, precision & recall scores for different models on informal Bangla dataset.

Algorithm	F1 Score		Precision		Recall	
	Training	**Testing**	**Training**	**Testing**	**Training**	**Testing**
Support Vector Machine	0.94	0.44	0.93	0.45	0.95	0.42
Passive-aggressive	0.90	0.47	0.91	0.46	0.90	0.43
Long Short-Term Memory	1	0.58	1	0.59	1	0.58
Recurrent Neural Network	1	0.55	1	0.43	1	0.42
Convolutional Neural Network	1	0.54	1	0.57	1	0.57

Table 4 illustrates the F1, precision, and recall scores for training and testing datasets across five models: SVM, passive-aggressive, LSTM, RNN, and CNN on informal Bangla dataset. In all cases, blue bars represent training scores, while orange bars represent testing scores. The charts show that while the SVM and passive-aggressive models achieve high scores during training, their testing scores drop significantly, indicating reduced performance on new data. The neural network-based models (LSTM, RNN, and CNN) exhibit very high training scores, close to 1.0, but their testing scores are notably lower across all metrics. This substantial gap suggests that these models are overfitting the training data, excelling in training but failing to generalize effectively to unseen data.

Table 5. Values of hyperparameter tuning for LSTM on informal Bangla dataset.

Hyperparameter	Optimal value
Embedding Dimension	150
Number Of LSTM Units	150
Dropout Rate	0.2
Optimizer	rmsprop
Best Validation Accuracy	0.7757816946166166

Table 5 presents the optimal hyperparameters for the LSTM model tuned using the informal Bangla dataset. The embedding dimension and number of LSTM units were both set to 150. A dropout rate of 0.2 was applied to prevent overfitting, and the optimizer used was RMSprop. This configuration achieved a best validation accuracy of approximately 77.58%.

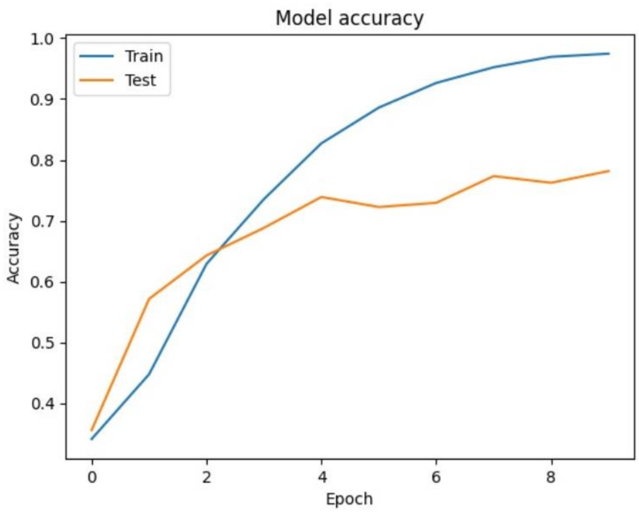

Fig. 13. Model accuracy for informal text dataset after hyperparameter tuning.

Figure 13 details the hyperparameter tuning for the LSTM model on the informal Bangla dataset. Key parameters include an embedding dimension and number of LSTM units set to 150, a dropout rate of 0.2, and the RMSprop optimizer. This tuning improved the model's performance, achieving a best validation accuracy of approximately 80.2%, highlighting the importance of hyperparameter optimization in enhancing the LSTM model's accuracy.

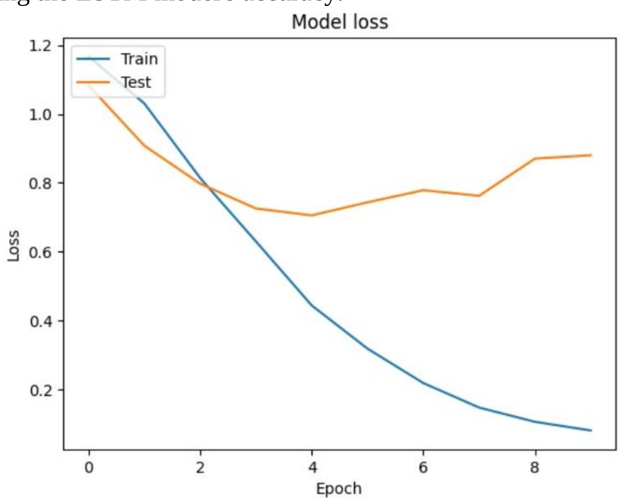

Fig. 14. Model loss of LSTM for informal text dataset after hyperparameter tuning.

Figure 14 shows that Hyperparameter tuning, adjusting an LSTM model's internal settings, led to a significant decrease in loss, signifying the model's improved ability to learn patterns and make accurate predictions on unseen data.

Table 6. Performance comparison with related existing system for informal Bangla dataset.

Model	**Accuracy**
Khondoker Ittehadul Islam et al. [1]	64.61%
Md. Atikur Rahman et al. [12]	78%
Asif Hassan et al. [13]	76%
Proposed model	80.3%

Table 6 compares the performance of existing systems on informally written text dataset. Khondoker Ittehadul Islam et al. [1] achieved 64.61% accuracy, Md. Atikur Rahman et al. [12] reached 78% accuracy, Asif Hassan et al. [13] attained 76% accuracy, while our proposed model achieved 80.3% accuracy, outperforming all other systems.

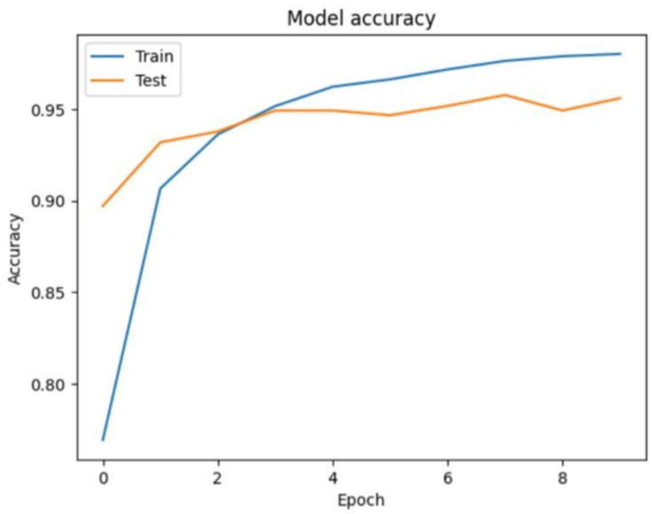

Fig. 15. Model accuracy of LSTM for formal text dataset after hyperparameter tuning.

Figure 15 illustrates the accuracy of the proposed model on a formally written Bangla text dataset for sentiment analysis after hyperparameter tuning on LSTM. The blue line represents the training accuracy, while the orange line shows the test accuracy over 10 epochs. Initially, the model quickly improves in accuracy, achieving above 95% within a few epochs. The training accuracy continues to rise slightly, indicating the model's learning progression, while the test accuracy stabilizes around 95%, suggesting a good generalization of unseen data. This demonstrates the effectiveness of the tuned LSTM model on the formal text dataset.

Table 7 compares the performance of existing systems on formally written text dataset. Rashedul Amin Tuhinv et al. [8] achieved 90% accuracy, Shaika Chowdhury et al. [9] reached 93% accuracy, Hasmot Ali et al. [10] attained 85.5% accuracy, Md Ferdous Wahid et al. [11] reached 95% accuracy, while our proposed model achieved 80.3% accuracy, outperforming all other systems.

Table 7. Performance comparison with related existing system for informal Bangla dataset.

Model	Accuracy
Rashedul Amin Tuhinv et al. [8]	90%
Shaika Chowdhury et al. [9]	93%
Hasmot Ali et al. [10]	85.5%
Md Ferdous Wahid et al. [11]	95%
Proposed model	95.6%

5 Conclusion and Future Work

This study highlights the growing importance of text as a source of valuable information, particularly in areas such as product analysis and social media monitoring. The research focuses on sentiment analysis of Bangla text, extracting insights into the emotional tone and opinions expressed in the data. A dataset of sentiment-laden Bangla text comments was collected and analyzed. Despite the challenges posed by the Bangla language's complexity and the noisy nature of the data, the results were encouraging for this resource-scarce language. The study concludes that supervised methods are not efficient for handling large datasets in Bangla due to these complexities.

For future work, the study suggests enhancing Bangla text sentiment analysis by incorporating multiple types of data, including emojis, audio, and images. Audio data can provide additional emotional insights through tone, pitch, and rhythm, while images and videos can convey emotions through facial expressions, body language, and contextual objects. Collecting data from diverse domains, such as social media, customer reviews, news articles, and general public comments, is essential for developing a robust model.

Acknowledgments. This work will be funded by Hamdard University Bangladesh.

Disclosure of Interests. The authors have no competing interests.

References

1. Islam, K.I. et al. (2021) ‘Sentnob: A dataset for analysing sentiment on Noisy Bangla texts’, Findings of the Association for Computational Linguistics: EMNLP 2021 [Preprint]. doi:10.18653/v1/2021.findings-emnlp.278.
2. Liu, B. (2012) Sentiment analysis and opinion mining. Cham, Switzerland: Springer.
3. Pang, B. and Lee, L. (2008) ‘Opinion mining and sentiment analysis’, *Foundations and Trends® in Information Retrieval*, 2(1–2), pp. 1–135. doi:10.1561/1500000011.
4. *Bangala | Ethnologue*. Available at: https://www.ethnologue.com/language/bxg (Accessed: 30 June 2024).
5. Das, A. and Bandyopadhyay, S. (2009) ‘7th International Conference on Natural Language Processing’, in Proceedings of ICON-2009. Macmillan.
6. Park, D.-H., Lee, J. and Han, I. (2006) ‘Pacific Asia Conference on Information Systems (PACIS)’, in *PACIS 2006 Proceedings*.

7. Elahi, K.T. *et al.* (2023) 'Explainable multimodal sentiment analysis on Bengali memes', *2023 26th International Conference on Computer and Information Technology (ICCIT)* [Preprint]. doi:10.1109/iccit60459.2023.10441342.
8. Tuhin, R.A. *et al.* (2019) 'An automated system of sentiment analysis from Bangla text using supervised learning techniques', *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)* [Preprint]. doi:10.1109/ccoms.2019.8821658.
9. Chowdhury, S. and Chowdhury, W. (2014) 'Performing sentiment analysis in Bangla microblog posts', *2014 International Conference on Informatics, Electronics & Vision (ICIEV)* [Preprint]. doi:10.1109/iciev.2014.6850712.
10. Ali, H. *et al.* (2020) 'Banglasenti: A dataset of Bangla words for sentiment analysis', *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* [Preprint]. doi:10.1109/icccnt49239.2020.9225565.
11. Wahid, Md.F., Hasan, Md.J. and Alom, Md.S. (2019) 'Cricket sentiment analysis from Bangla text using recurrent neural network with long short term memory model', *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)* [Preprint]. doi:10.1109/icbslp47725.2019.201500.
12. Rahman, Md.A. and Kumar Dey, E. (2018) 'Datasets for aspect-based sentiment analysis in Bangla and its Baseline Evaluation', *Data*, 3(2), p. 15. doi:10.3390/data3020015.
13. Hassan, A. *et al.* (2016) 'Sentiment Analysis on Bangla and romanized bangla text using Deep Recurrent models', *2016 International Workshop on Computational Intelligence (IWCI)* [Preprint]. doi:10.1109/iwci.2016.7860338.
14. Ahmed, Z., Shanto, S.S. and Jony, A.I. (2023) 'Advancement in Bangla sentiment analysis: A Comparative Study of Transformer-based and transfer learning models for e-commerce sentiment classification', *Journal of Information Systems Engineering and Business Intelligence*, 9(2), pp. 181–194. doi:10.20473/jisebi.9.2.181-194.
15. Das, P. and Sultana, N. (2022) 'Sentiment analysis on comments in Bengali language using text mining & machine learning approach', *2022 IEEE 7th International conference for Convergence in Technology (I2CT)* [Preprint]. doi:10.1109/i2ct54291.2022.9825373.
16. *Accuracy vs. precision vs. recall in machine learning: What's the difference?* (no date) *Evidently AI - Open-Source ML Monitoring and Observability*. Available at: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall (Accessed: 30 June 2024).