Sentiment Analysis in Bengali Text Data using NLP

Ankon sarker*, MD Sakhawat Hossen†, Avirup Howlader‡, Rezvi Ahmed§, Bakhtyear Fahim¶
*BRAC University, Bangladesh, iamankonsarkar@gmail.com
†Chittagong University of Engineering and Technology, Bangladesh, sakhawat3003@gmail.com
‡Bangladesh University of Business and Technology, Bangladesh, avik.4521@gmail.com
§BRAC University, Bangladesh, rezvi.ahmed222@gmail.com
¶Rajshahi University of Engineering and Technology, Bangladesh, bakhtyear.fahim@gmail.com

***Abstract*—In this study, we present BANEmo, a novel Bangla language dataset designed specifically for sentiment analysis tasks. BANEmo is derived from a diverse range of Bangla text/comments, harvested from both social media and newspapers, encompassing a total of 15,000 individual comments. Each comment in the dataset is meticulously labeled for various sentiments, such as happiness, sadness, disgust, anger, and more. In our investigation, we initially focus on a binary classification of sentiments, specifically happiness and sadness. Two classical machine learning algorithms, Support Vector Machine (SVM) and Decision Tree (DT), were employed to establish a baseline performance on the Banemo dataset. Notably, these conventional models achieved state-of-the-art accuracy levels for binary Bangla sentiment analysis. To further push the boundaries, we employed mBERT, a transformer-based pre-training method, for sentiment analysis. The results exceeded the benchmarks set by the traditional models, underlining the potential of transformer-based models in Bangla sentiment analysis. Our research makes a dual contribution to the field. Firstly, the Banemo dataset fills a vital gap in resources available for Bangla sentiment analysis, serving as a new benchmark for future research. Secondly, our findings suggest that mBERT can be effectively used for sentiment analysis in low-resource languages like Bangla, indicating a promising avenue for future research in this area. This study aims to stimulate further advancements in the field of Bangla natural language processing and sentiment analysis.**

***Index Terms*—NLP, BERT, mBERT, BanglaBERT, BoW, TF-IDF, Keras, Banemo**

I. Introduction

Sentiment analysis, a widely recognized subfield of natural language processing (NLP), aims to extract subjective information such as opinions or emotions from text. While substantial research has been conducted in this domain, the majority of work focuses on resource-rich languages such as English. This leaves many low-resource languages like Bangla, the seventh most spoken language worldwide, relatively underexplored. This lack of attention is primarily due to the scarcity of labeled datasets in these languages, which are indispensable for training and evaluating machine learning models. In this context, we introduce "Banemo", a comprehensive and diverse dataset comprising 15,000 Bangla text comments collected from social media and newspapers. The dataset has been annotated meticulously with labels such as happiness, sadness, anger, and disgust, among others. The diversity in data sources and the broad spectrum of sentiment labels provide an extensive ground for training and benchmarking machine learning models.

However, we've limited our initial explorations to a binary sentiment classification task, focusing specifically on happiness and sadness labels. To establish the foundational results on the Banemo dataset, we employed two classical machine learning models, the Support Vector Machine (SVM) and Decision Tree (DT). Remarkably, these models achieved state-of-the-art accuracy for Bangla binary sentiment classification.

To leverage the recent advancements in NLP, we applied a transformer-based model, mBERT, to the same task. The mBERT model outperformed the classical machine learning models, demonstrating the potential of transformer-based models in sentiment analysis tasks for low-resource languages.

The paper is organized as follows: after this introduction, we will describe the methods used for data collection and annotation in Section II. Section III presents the experimental setup, including the details of the machine learning models and the evaluation metrics used. Section IV discusses the experimental results, followed by a comparative analysis of the performances of the different models. We conclude the paper in Section V, summarizing our findings and suggesting potential directions for future research.

II. Background Study

Sentiment analysis, also referred to as opinion mining, has witnessed substantial growth in research interest over the past decade. The rise of social media platforms and user-generated content has further accelerated the need for effective sentiment analysis models [1]. While English language sentiment analysis has been extensively researched, there is a paucity of work on under-resourced languages such as Bangla.

Pak and Paroubek [2] provided one of the early directions in sentiment analysis using tweets. They employed a binary classification task distinguishing between positive

and negative sentiments, and this binary categorization has been a common approach in subsequent studies. In terms of machine learning techniques for sentiment analysis, Support Vector Machines (SVM) and Decision Trees (DT) have been popular choices. For instance, Dave, Lawrence, and Pennock [3] utilized SVM for sentiment polarity classification and found that it outperformed other machine learning methods such as Naive Bayes and Maximum Entropy.

With the advent of deep learning, more complex models like Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM) have been employed for sentiment analysis [4] [5]. These models demonstrated significantly higher performance than traditional machine learning algorithms, demonstrating the potential of deep learning in this field.

The transformer architecture, introduced by Vaswani et al [6], has further revolutionized NLP tasks, including sentiment analysis. Transformer-based models like BERT [7] and its multilingual variant mBERT have shown superior performance across multiple languages and tasks.

For Bangla sentiment analysis, existing research is limited. Hossain et al. [8] created a Bangla sentiment analysis dataset and utilized traditional machine learning models for classification. However, their dataset was relatively small, and they did not explore transformer-based models. In our work, we introduce a novel and larger Bangla sentiment analysis dataset, Banemo, and leverage both traditional machine learning and transformer-based models for sentiment classification, extending the existing research in the field.

III. Methodology

This section outlines the research methodology employed in our study. The first stage involved the construction of a novel dataset, comprising entirely of annotated Bengali comments. Data pre-processing techniques were then utilized to eliminate noise, ensuring the integrity of the dataset. Subsequent to this, text data was transformed using word embedding techniques to conform to the required input format. The processed data was subsequently utilized for the training and testing of machine learning models.

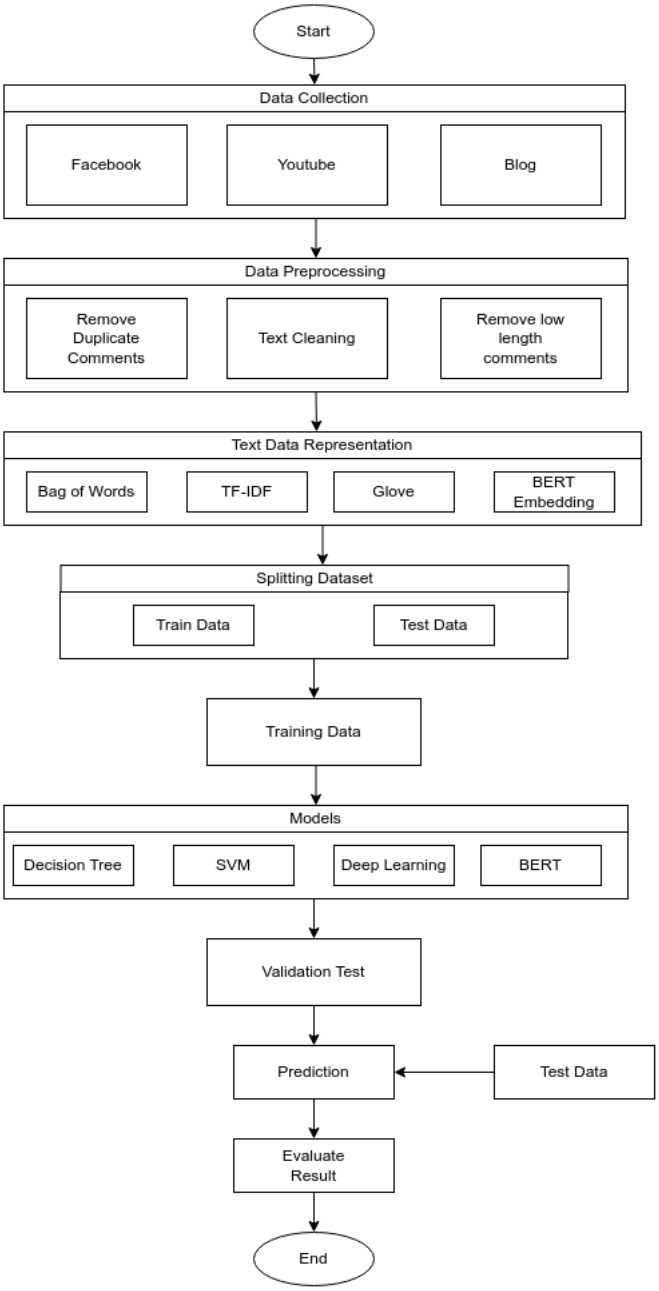

Figure 1: Methodology for Binary Sentiment Analysis

1) ***Description of the Dataset****:* The dataset, dubbed "BANEmo," is a novel, manually annotated compilation of Bengali text data, amounting to 14,999 entries. These entries were gathered from the comments sections of various social media platforms, including Facebook, YouTube, and several leading Bengali online news portals such as Prothom Alo, BBC Bangla, DW Bangla, among others.

The entries in the dataset have been carefully categorized into eight distinct classes. Six of these classes correspond to basic emotions: happiness, sadness, disgust, fear, surprise, and anger. The remaining two classes are designated for sarcasm and undefined sentiments. The annotation process was undertaken by three native Bengali speakers. To ensure the highest degree of accuracy, each entry was independently annotated by each individual. The most commonly identified sentiment across these independent annotations was then assigned as the official label for each entry. In instances where the three annotators assigned three distinct emotions to a single text entry, a Bengali-speaking meta-annotator was engaged to arbitrate and determine the final sentiment label.

For this study, only two classes labeled as “Happiness”

and “Sadness” has been chosen due to the prevalence of imbalance in the dataset.

Table I: Final Annotation based on Majority Vote

Text Data	**Label**
আলহামদুলিল্লাহ, অনেক দিন পর একটা ভাল সংবাদ পেলাম	Happy
নেহাত ষড়যন্ত্র নয়তো	Fear

Final Annotation based on Meta Annotator

	Text Data	**Label**
	আমাদের দেশের শিক্ষা প্রতিষ্ঠান রাজনীতিতে প্রথম হবে এটা নিশ্চিত করে বলা যায়	Disgust

Table II: Final Annotation based on Meta Annotator

The figure below illustrates the distribution of various emotion categories following the completion of the final labeling process.

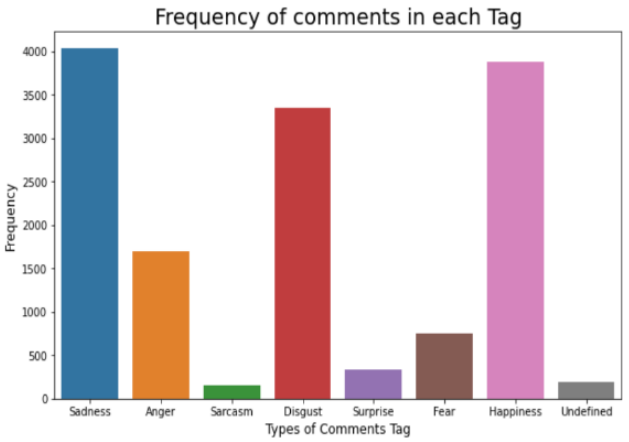

Figure 2: Frequency of the comments for each labeled category

2) ***Pre-processing of the Dataset:*** De-duplication and null value removal: The original dataset contained 95 duplicate entries and a single null value. These redundant and null values were eliminated from the dataset, leaving a total of 14,903 unique text entries.

Cleaning of the text data The text cleaning procedure entailed the removal of superfluous components such as emojis, punctuation, and user tags, thereby reducing noise in the dataset. Specifically, punctuation was removed from the comment text data to facilitate improved categorization. In addition, elements such as emojis, hashtags, and other extraneous entities were purged to enhance the cleanliness of the data. The removal of emoticons assisted the annotators in assigning unbiased labels to the text data. Further, entries that contained less than three words were discarded.
Following the implementation of these pre-processing steps, a total of 14,409 distinct comment entries remained available for subsequent analysis.

3) ***Representation of the Bangla Comments Data:*** For the purpose of this research, two widely-used techniques were employed for the representation of text data: the Bag of Words (BoW) model and Term Frequency-Inverse Document Frequency (TF-IDF).
Bag of Words model is a popular method in Natural Language Processing (NLP) for converting text into numerical features that can be understood by machine learning algorithms. In our research, the BoW model was utilized to represent Bengali text data in the ”BANEmo” dataset. The BoW model works by constructing a vocabulary of all unique words in the text data, then representing each comment by a vector, indicating the presence or absence of each word from the vocabulary in the comment. Despite its simplicity, the BoW model does not consider the order of words and it assigns equal importance to all words, which can be a limitation in capturing the context in the sentiment analysis task.
Term Frequency-Inverse Document Frequency (TF-IDF) has been utilized to counteract the limitations of the BoW model for text representation. TF-IDF is a statistical measure that reflects how important a word is to a document in a collection or corpus. It achieves this by balancing the frequency of the word in the document (Term Frequency) against the frequency of the word in the whole dataset (Inverse Document Frequency). This helps to give more weight to the words that are unique to a document, thereby potentially capturing more nuanced sentiment information than the BoW model. This numerical representation of the text data was then used as input for both classical machine learning and transformer-based models.
By utilizing these two models for text data representation, we were able to convert the raw Bengali comments in the ”BANEmo” dataset into a format that could be processed by our machine learning models, thereby facilitating the sentiment analysis task.

IV. Classification Methods

Our approach encompassed the use of diverse text representation methods and models in an effort to conduct a comprehensive sentiment analysis. Initial stages involved the deployment of Decision Trees and Support Vector Machines as foundational machine learning methods. Subsequently, we transitioned towards implementing deep learning and transformer-based models. The performance of these varied approaches was then critically analyzed and evaluated.

A. ***Decision Tree Model***

For our sentiment analysis task, we initially employed the Decision Tree (DT) model as one of the classification methods. A Decision Tree is a popular machine learning

algorithm that builds a model in the form of a tree structure, making it highly interpretable and easy to visualize [9].
In the context of our research, the decision tree was used to classify the comments in the "BANEmo" dataset into two sentiment categories, namely happiness and sadness. In a decision tree model, each internal node of the tree corresponds to an attribute or feature (in this case, the words or terms derived from the Bag of Words or TF-IDF methods), and each leaf node represents a class label (in this case, happiness or sadness).
The tree is built by selecting attributes that best split the data based on a certain criterion. This can be information gain, gain ratio, or Gini index, among others. The attribute with the highest value according to the selected criterion is placed at the root node, and the dataset is split based on this attribute. This process is recursively applied to each child node until the tree is fully built [10].
One of the advantages of decision trees is their simplicity and ease of interpretation. However, they can suffer from overfitting, especially when dealing with a large number of features, as can be the case when using Bag of Words or TF-IDF for text representation. In our research, we took measures to prevent overfitting, such as pruning the tree and limiting the maximum depth of the tree.
The decision tree model served as one of our baseline models, and the results were compared with other machine learning models and the more advanced transformer-based models.

B. *Support Vector Machine*

In addition to the Decision Tree, the Support Vector Machine (SVM) was also used as a classification method in this research. The SVM is a powerful and versatile machine learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection. It is particularly well-suited for the classification of complex but small or medium-sized datasets [11].
The SVM works by mapping the input data into a high-dimensional feature space and then finding the hyperplane that distinctly classifies the data points into separate classes, in our case, sentiments of happiness and sadness. The optimal hyperplane is determined by the maximum margin, which is the maximum distance between the data points of two classes. SVMs can handle linearly separable as well as non-linearly separable cases using what is known as the kernel trick, a function that takes low dimensional input space and transforms it into a higher dimensional space [12].
In this study, the SVM model was trained using the text data represented as feature vectors from the Bag of Words or TF-IDF methods. The trained model was then used to classify the sentiments in the "BANEmo" dataset. The performance of the SVM model was benchmarked against the Decision Tree model and also compared with the deep learning and transformer-based models to understand the relative strengths and weaknesses of these different approaches.

C. *Deep Learning Model*

For our research, we also incorporated deep learning models using Keras, a high-level neural networks API. Keras, being user-friendly and modular, allows for easy and fast prototyping of deep learning models [13].
In this context, we employed Keras to design and train our deep learning models for sentiment classification. Keras provides an assortment of pre-processed layers and project templates that help in building deep learning models. We could design both Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) like LSTM (Long Short Term Memory) for the task. These models have demonstrated remarkable success in NLP and sentiment analysis tasks by being able to capture the context in the text data [14] [15].
We used the Adam optimizer for training our models. Adam, short for Adaptive Moment Estimation, is an optimization algorithm that can handle sparse gradients on noisy problems. Adam combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp. It computes adaptive learning rates for different parameters and performs well in practice with little memory requirements. It is computationally efficient and has little memory requirement, making it suitable for our task [16].
The performance of the deep learning models was compared with the classical machine learning models (SVM and Decision Tree) and transformer-based models to identify the most effective model for Bangla sentiment analysis on the "BANEmo" dataset.

D. *BERT*

In our study, we extended the scope of our exploration to include transformer-based models, particularly mBERT (Multilingual BERT) and BanglaBERT. These models belong to the family of BERT models, which have revolutionized the field of Natural Language Processing (NLP).
BERT, which stands for Bidirectional Encoder Representations from Transformers, was introduced by Devlin et al [**?**]. Unlike previous models, BERT is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. This property allows the model to understand the context and ambiguity of words in text more accurately. For instance, the model can differentiate between the multiple meanings of a word based on the context in which it is used.
mBERT is a multilingual version of BERT, which is pre-trained on text from the top 104 languages with the largest Wikipedias. It is particularly useful when working with low-resource languages, as it can take advantage of transfer learning from other languages.
BanglaBERT, on the other hand, is a variant of BERT

specifically trained for the Bangla language introduced by Bhattacharjee [18]. By leveraging the power of the BERT model and the specificity of the Bangla language, BanglaBERT has the potential to capture nuanced sentiments in the Bangla text data.
In this research, we employed these transformer-based models for the sentiment classification task. Text data represented as feature vectors from the Bag of Words or TF-IDF methods were provided as input to these models. The performance of these models was then compared with the classical machine learning models (SVM and Decision Tree) and deep learning models to evaluate the effectiveness of these methods for sentiment analysis on the "BANEmo" dataset.

V. Performance Parameters

The performance of the sentiment analysis models in this research was evaluated using three main metrics: Sensitivity (or Recall), Positive Predictive Value (PPV, also known as Precision), and F-Measure (or F1 Score).

- Sensitivity or Recall: Sensitivity measures the proportion of actual positive cases that were correctly identified. In the context of our study, it refers to the proportion of comments from a certain sentiment that were correctly classified by the model. Mathematically, it can be expressed as:

$$\text{Recall} = \frac{\text{True Positive}}{\text{Truth Positive} + \text{False Negative}} \quad (1)$$

  where, True Positives are the correctly identified positive instances, and False Negatives are the positive instances that were incorrectly classified as negative.
- Positive Predictive Value (PPV) or Precision: PPV measures the proportion of predicted positive cases that were correctly real positives. In terms of our sentiment analysis task, it indicates the proportion of comments predicted to belong to a certain sentiment that truly belong to that sentiment. It can be calculated as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

  where False Positives are the negative instances that were incorrectly classified as positive.
- F-Measure or F1 Score: F-Measure is the harmonic mean of Precision and Recall and provides a single metric that balances both these considerations. It is especially useful when you want to compare two or more models or when there is an uneven class distribution, as it takes both false positives and false negatives into account. It is calculated as:

$$\text{Fi Score} = 2 * \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (3)$$

  These metrics offer a comprehensive view of the model's performance, considering both the model's ability to correctly identify a sentiment (Recall) and its ability to not misclassify other sentiments as the target sentiment (Precision). By using the F1 Score, we are able to consider both of these aspects in a single metric, which gives us a better overall understanding of the model's performance.

VI. Results and Discussion

As previously detailed, the initial stage of our research involved the classification and labeling of our dataset, "BANEmo," according to six basic emotion classes: "sadness," "happiness," "disgust," "surprise," "fear," and "anger." Moreover, some comments were assigned labels of "sarcasm," while those that did not align with any particular emotion were categorized as "undefined." However, due to the insufficient number of entries within the "surprise," "sarcasm," and "undefined" classes, we opted to merge them into a new class labeled "others."
In alignment with the findings of Jack et al [19]. which suggested human emotions could be effectively categorized into just five categories: happiness, sadness, fear, and a combined class of anger/disgust, we proceeded to further consolidate our dataset. This recommendation is based on the significant semantic similarities between expressions of anger and disgust. Consequently, we restructured our dataset into five classes, with the combined "anger and disgust" class being renamed as "exasperation" for more effective classification by our model. In this study, we have performed the analysis on the highly polarized comments labeled as "Happiness" and "Sadness".

Frequency of the texts after recombination

Types of Emotions	Total Data
Happiness	3886
Sadness	4035
Exasperation	5057
Fear	752
Others	679

Table III: Frequency of the texts after recombination

In order to effectively train and evaluate our models, we divided the "BANEmo" dataset into two distinct subsets: a training set and a test set. The training set is used to train the model, allowing it to learn and adjust its parameters to minimize the difference between its predictions and the actual labels. The test set, on the other hand, is used to evaluate the performance of the trained model on unseen data, providing an unbiased assessment of its generalization capabilities.
For our research, we allocated 60% of the dataset for training purposes. This subset provides a substantial amount of data for our models to learn from, thereby enabling them to capture the underlying patterns within the data that are relevant for sentiment classification.
The remaining 40% of the dataset was set aside as the test set. This allocation ensures that we have a sizable

amount of unseen data to gauge the performance of our models accurately.

Model	Precision	Recall	F1 Score	Accuracy
Decision Tree	72.13%	74.5%	73.3%	74%
Support Vector Machine	80.11%	77%	78.56%	79.1%

Table IV: Performance Metrics for Decision Tree and Support Vector Machine

A. ***Decision Tree***

The Decision Tree model was implemented as one of the baseline machine learning methods for our sentiment analysis task. On evaluating the performance of the Decision Tree model, the results showed a Recall (or Sensitivity) of 74.5%, a Precision (or Positive Predictive Value) of 72.13%, and an F1 Score (or F-Measure) of 73.3%.

The Recall of 74.5% suggests that the Decision Tree model was able to correctly identify 74.5% of the actual positive sentiment instances in the test data. In other words, the model captured a substantial majority of the comments belonging to the target sentiment classes.

The Precision of 72.13% indicates that of all the instances that the Decision Tree model predicted as positive, 72.13% were indeed true positive sentiments. This signifies that the model's predictions were reasonably accurate and not overly skewed by false positives. Lastly, the F1 Score of 73.3% provides a balance between the Precision and Recall scores. An F1 Score can be considered as a single metric that combines both precision and recall, thereby providing an overall measure of the model's performance. A score of 73.3% signifies that the Decision Tree model demonstrated a well-rounded performance, with a relatively balanced capacity for correctly identifying positive instances (recall), and for issuing correct positive predictions (precision).

B. ***Support Vector Machine***

Upon implementing and evaluating the Support Vector Machine (SVM) model for our sentiment analysis task, we obtained a Recall of 77%, a Precision of 80.11%, an F1 Score of 78.56%, and an Accuracy of 79.1%.

When compared to the Decision Tree model, the SVM model exhibited a higher performance across all metrics. The increase in Recall from 74.5% to 77% signifies that the SVM model was capable of correctly identifying a higher proportion of actual positive sentiment instances in the test data. This indicates an enhanced capability of the SVM model in recognizing and classifying the sentiments correctly.

The Precision also increased from 72.13% to 80.11%, indicating that a greater proportion of the instances that the SVM model predicted as positive were indeed true positive sentiments. This suggests that the SVM model provided more accurate predictions, with fewer false positives.

The F1 Score, which is a balance between Precision and Recall, also improved from 73.3% to 78.56%. This demonstrates that the SVM model achieved a better balance between correctly identifying positive instances and issuing correct positive predictions, thus offering better overall performance.

Lastly, the SVM model achieved an Accuracy of 79.1%. Accuracy is a measure of how many classifications the model got correct out of all classifications. This indicates that the SVM model correctly classified the sentiments of 79.1% of all comments in the test data, outperforming the Decision Tree model in the task of sentiment classification. These results suggest that, for this particular task and dataset, the SVM model may be a more effective choice for sentiment analysis compared to the Decision Tree model.

Model	Number of Epochs	Accuracy
Deep Learning (Keras)	2	83.31%
mBERT	2	82%
BanglaBERT	7	81.15%

Table V: Performance Matrices for Deep Learning and BERT Models

C. *Deep Learning*

In our exploration of deep learning models, we used the Keras framework with the Adam optimizer. After two epochs of training, the model achieved a validation accuracy of 83.31%.

This level of validation accuracy surpasses the accuracies attained by both the Decision Tree and Support Vector Machine models previously used. An epoch refers to one cycle through the full training dataset. During these two epochs, our deep learning model was able to learn and adjust its parameters to the data more effectively, leading to superior performance.

The validation accuracy of 83.31% signifies that the deep learning model correctly classified the sentiments of 83.31% of the comments in the validation set. This is a notable improvement compared to the accuracies of the Decision Tree and SVM models, which were 73.3% and 79.1% respectively.

This significant enhancement in performance underscores the potential of deep learning methods for sentiment analysis tasks. By learning complex patterns and representations in the data, these models can often outperform classical machine learning methods. The results also highlight the effectiveness of the Adam optimizer, which balances both the magnitude and direction of the gradients to optimize the learning process.

The comparison suggests that, for the task at hand and the "BANEmo" dataset, the Keras deep learning model with the Adam optimizer could be a more potent model for sentiment classification than the Decision Tree or Support Vector Machine models.

D. BERT

Further extending our research, we incorporated the transformer-based models mBERT (Multilingual BERT) and BanglaBERT. These models have been shown to perform excellently in various NLP tasks, including sentiment analysis.

Upon evaluation, the mBERT model demonstrated an accuracy of 82%, implying that it correctly predicted the sentiment of 82% of the comments in the test set. This performance is slightly lower compared to the 83.31% validation accuracy achieved by the Keras deep learning model. Despite this, the mBERT model still outperforms the Decision Tree and Support Vector Machine models, highlighting the potential of transformer-based models in sentiment analysis tasks.

The BanglaBERT model, specifically trained for the Bangla language, yielded an accuracy of 81.15%. This represents a marginal decrease in performance compared to mBERT. One possible explanation for this could be the variation in the training corpus for both models, as mBERT is trained on multilingual data while BanglaBERT is trained specifically on Bangla text data.

However, despite the slight drop in performance compared to mBERT and the Keras deep learning model, BanglaBERT still outperformed both the Decision Tree and SVM models, solidifying the effectiveness of transformer-based models for sentiment analysis tasks in the Bangla language.

Comparatively, the Keras deep learning model yielded state-of-the-art performance on the sentiment classification task in our study. Nevertheless, the mBERT and BanglaBERT models' performances indicate that transformer-based models offer a promising direction for future research in Bangla sentiment analysis.

VII. Conclusion and Future Work

In this research, we introduced "BANEmo," a novel Bengali text dataset, and employed various machine learning and deep learning models to perform sentiment analysis. From classical machine learning models such as Decision Trees and Support Vector Machines, to deep learning models using Keras with Adam optimizer, and further to transformer-based models like mBERT and BanglaBERT, we traversed a broad spectrum of methods for sentiment classification.

Our results indicate that deep learning models and transformer-based models perform notably better than classical machine learning models on our dataset. While the Keras model yielded the highest performance, the transformer-based models demonstrated their potential in this task, particularly for low-resource languages like Bangla.

The comprehensive exploration undertaken in this study provides significant insights for future research. However, as with all studies, ours also opens up new questions and opportunities for future work:

- One potential direction is to explore more advanced models like GPT-3 or RoBERTa for sentiment analysis in Bangla and compare their performance with the models used in this study.
- Another prospect is to extend the application of the "BANEmo" dataset to multi-class sentiment analysis rather than limiting it to binary classification. This could pave the way for a more nuanced understanding and prediction of sentiments in Bengali text.
- The annotation process for the dataset could be enhanced further by involving more annotators or using automated sentiment analysis tools to assist the annotators.
- It could be interesting to apply transfer learning techniques, where a model trained on a large dataset in one language is fine-tuned on a smaller dataset in another language.

This study demonstrates the potential of NLP research in under-resourced languages like Bangla and provides a foundation for future researchers to build upon.

References

[1] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1–135.

[2] Pak, A., & Paroubek, P. (2010, May). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In LREC (Vol. 10, pp. 1320-1326).

[3] Dave, K., Lawrence, S., & Pennock, D. M. (2003, August). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web (pp. 519-528).

[4] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[5] Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 1422-1432).

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[8] Hossain, M. S., Mojumder, N. J., Das, A., Sultana, S., & Chy, M. A. U. (2020). Bengali sentiment analysis using machine learning and deep learning models. Journal of Physics: Conference Series, 1529(2), 022027.

[9] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

[10] Rokach, L., & Maimon, O. (2014). Data mining with decision trees: theory and applications. World scientific.

[11] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[12] Scholkopf, B., & Smola, A. J. (2001). Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

[13] Chollet, F., et al. (2015). Keras. https://keras.io.

[14] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)

[15] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[16] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[17] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[18] Bhattacharjee, A., Hasan, T., Uddin, W. A., Mubasshir, K., Islam, M. S., Iqbal, A., ... & Shahriyar, R. (2022). Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. Findings of the North American Chapter of the Association for Computational Linguistics: NAACL.

[19] Jack, R. E., Garrod, O. G., & Schyns, P. G. (2014). Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time. Current biology, 24(2), 187-192.