STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
*Stat., Optim. Inf. Comput.*, Vol. **13**, January 2025, pp 331–346.
Published online in International Academic Press (www.IAPress.org)

# Sentiment Analysis in the Transformative Era of Machine Learning: A Comprehensive Review

Sayeda Muntaha Ferdous [1], Syed Nur E Newaz [1], Shafayat Bin Shabbir Mugdha [1], Mahtab Uddin [2,*]

[1] *Department of Computer Science & Engineering, United International University, Dhaka-1212, Bangladesh*
[2] *Institute of Natural Sciences, United International University, Dhaka-1212, Bangladesh*

**Abstract** Sentiment analysis, which stands for opinion mining, is a natural language processing (NLP) technique that involves identifying, extracting, and analyzing sentiments or opinions expressed in text data. The primary goal of sentiment analysis is to determine the sentiment polarity of a given piece of text, whether it is positive, negative, or neutral. This analysis can be applied to various types of content, such as product reviews, social media posts, customer feedback, and news articles. Sentiment analysis algorithms use machine learning and text classification to understand subjective information conveyed in text, helping businesses, organizations, and individuals gain insight into public opinions and emotions about specific topics, products, or services. In this study, we conducted sentiment analysis on a Bengali dataset. For feature extraction, we implemented the term frequency-inverse document frequency (TF-IDF) technique, and for feature selection, we applied an extra tree classifier approach. Subsequently, we trained our machine learning model, achieving an impressive accuracy rate of 92%.

**Keywords** Sentiment Analysis, Natural Language Processing, Machine Learning Algorithms, Sarcasm Detection, Bengali Dataset, Neural Networks

**AMS 2010 subject classifications** 68P10, 68T50, 68U15, 68U35, 68W05

**DOI:** 10.19139/soic-2310-5070-2113

## 1. Introduction

In the dynamic landscape of natural language processing, sentiment analysis, also known as opinion mining, stands out as a pivotal area that explores the extraction and interpretation of sentiments expressed in textual data. This field has garnered substantial attention due to its widespread applications in understanding public opinions, customer feedback, and social media sentiments. Sentiment analysis plays a crucial role in diverse domains such as business, marketing, and social sciences, offering valuable insights for informed decision-making [1]. By analyzing sentiments in textual data, sentiment analysis enables businesses to gauge customer satisfaction, identify areas for improvement, and tailor their marketing strategies accordingly. Moreover, in the realm of social sciences, sentiment analysis aids researchers in studying public attitudes and opinions on various topics, contributing to a deeper understanding of societal trends and behaviors. Overall, sentiment analysis has become an indispensable tool for organizations and researchers alike, revolutionizing the way we interpret and utilize textual data.

The intricate nature of sentiment analysis involves discerning the emotional tone conveyed in a piece of text and categorizing it as positive, negative, or neutral. Over the years, researchers have delved

[*] Correspondence to: Mahtab Uddin (Email: mahtab@ins.uiu.ac.bd). Institute of Natural Sciences, United International University, Dhaka-1212, Bangladesh.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

into various methodologies and techniques to enhance the accuracy and efficiency of sentiment analysis techniques [2]. This introduction aims to explore the evolution of sentiment analysis, shed light on key developments, and highlight the significance of this field in contemporary research [3].

In the real world, opinions are an essential source of information that can be used to understand people's attitudes, feelings, and evaluations of goods and services. For guidance in making decisions, people frequently depend on the opinions of others. Numerous e-commerce businesses have adopted this strategy to improve customers' shopping experiences. These online platforms enable users to share their comments on the provided products and services. However, the immense volume of comments and reviews from a global user base poses a challenge for customers trying to form conclusions before making purchases. Sentiment analysis, a branch of data science, specializes in analyzing people's comments, opinions, and reviews related to a particular service or product. The concept of sentiment analysis was first invented by Nasukawa and Yi in 2003 [4], whereas the term opinion mining was initially used by Dave, Lawrence [5], and Pennock [1] in the same year. Sentiment analysis is a process that involves examining the sentiments sent by customers in their user reviews and comments to determine if the overall product review is favorable or negative.

Sentiment analysis is an automated process that determines whether user-generated text expresses a positive, negative, or neutral viewpoint about a particular person, thing, subject, or situation. Entity classification can occur at various levels, including within individual documents or sentences and at the aspect or feature level [6]. The complete document serves as the fundamental component for classification, regardless of whether it is favorable or unfavorable. Before assigning subjective or objective labels to individual sentences, sentiment classification at the sentence level determines whether the sentence is positive, negative, or neutral. A negligible differentiation exists between these two approaches, given that a sentence fundamentally constitutes a concise document. The purpose of aspect or feature-level sentiment classification is to identify and extract product-specific characteristics from the source data [6]. In this study, we conducted sentiment analysis on a Bengali dataset. For feature extraction, we implemented the term frequency-inverse document frequency (TF-IDF) technique, and for feature selection, we applied a random forest approach. Subsequently, we trained the proposed machine learning model, achieving an impressive accuracy rate of 94%.

## 2. Related work

There are two fundamental approaches for sentiment detection in textual data, namely, symbolic techniques and machine learning models [7]. The subsequent sections proceed with an exploration of these methodologies.

- **Symbolic techniques:** A significant portion of the exploration of unsupervised sentiment analysis and classification through symbolic techniques relies on leveraging existing lexical resources. Turney [8] adopted a bag-of-words approach for sentiment analysis, wherein the interconnections between individual words are disregarded and a document is portrayed as a simple aggregation of words. The determination of the overall sentiment involves evaluating the sentiments of each word, and these values are then combined using certain aggregation functions. Turney inferred the polarity of a review by calculating the average semantic orientation of tuples derived from the review. These tuples consist of phrases containing adjectives or adverbs. The semantic orientation of each phrase is determined by comparing the word with a list of known positive and negative words.
  Kamps et al. [9] employed the lexical database WordNet [10] to assess the emotional characteristics of words across various dimensions. They devised a distance metric within WordNet, focusing on determining the semantic orientation of adjectives. WordNet encompasses a network of words linked by synonym relationships. Baroni et al. [11] developed a system utilizing the word space model formalism to address challenges in lexical substitution tasks. This system captures both the local context and the overall distribution of a word. Balahur et al. [12] introduced EmotiNet, a conceptual

framework for text representation that retains the structure and semantics of real events within a specific domain. EmotiNet employs finite state automata to identify emotional responses triggered by actions. In a study by SemEval et al. [13], a combination of coarse-grained and fine-grained approaches was employed to discern sentiments in news headlines. The coarse-grained approach involved binary classification of emotions, while the fine-grained approach classified emotions into distinct levels.

- **Machine learning models:** Several machine learning models, such as Naive Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM), are applied for the classification of reviews [14]. Various features can contribute to sentiment classification, including term presence, term frequency, negation, n-grams, and Part-of-Speech [15]. These features play a crucial role in discerning the semantic orientation of words, phrases, sentences, and entire documents, revealing their polarity, which can be either positive or negative.
  In their study, Domingos et al. [16] found that Naive Bayes performs effectively for specific issues characterized by highly interdependent features. This observation is intriguing considering the foundational assumption of Naive Bayes, which posits feature independence. Zhen Niu et al. [17] proposed a novel model incorporating efficient techniques for feature selection, weight computation, and classification. This model is established in the Bayesian algorithm. The classifier's weights are dynamically adjusted, utilizing both representative and unique features. representative features encapsulate information emblematic of a class, while unique features contribute to class differentiation. By leveraging these weights, they computed the probability associated with each classification, thereby enhancing the performance of the Bayesian algorithm.
  Xia et al. [18] implemented an ensemble technique to enhance sentiment classification. This framework results from the amalgamation of diverse feature sets and classification techniques. Their approach involved the utilization of two distinct types of feature sets and three base classifiers to construct the ensemble framework. The feature sets were generated based on part-of-speech information and word relationships. The chosen base classifiers were Naive Bayes, Maximum Entropy, and Support Vector Machines. To achieve sentiment classification, various ensemble techniques such as fixed combination, weighted combination, and meta-classifier combination were applied, leading to improved accuracy.

Sarcasm poses significant challenges in sentiment analysis due to its reliance on contextual cues and nuanced expressions, which often elude traditional techniques. Researchers have made efforts to determine public sentiment regarding movies, news, and other topics through Twitter posts. V.M. Kiran et al. [19] leveraged data from additional publicly accessible databases, including IMDB and Blippr, with appropriate adjustments to enhance sentiment analysis on Twitter within the movie domain. Existing techniques of detecting sarcasm struggle with accurately identifying sarcasm, leading to misinterpretation of sentiments and reduced model reliability.

## 3. Proposed methodology

A dataset is formulated by collecting various social platforms, which typically consist of brief messages containing slang words and misspellings. The objective is to conduct a sentiment analysis at the sentence level, involving a three-phase process. Initially, preprocessing is applied to the posts. Subsequently, a feature vector is constructed using pertinent features. Finally, employing diverse classifiers, the posts are categorized into positive and negative classes. The ultimate sentiment is determined based on the number of posts in each class, providing a comprehensive view of the overall sentiment distribution. Figure 1 is the visual representation of the proposed methodology.

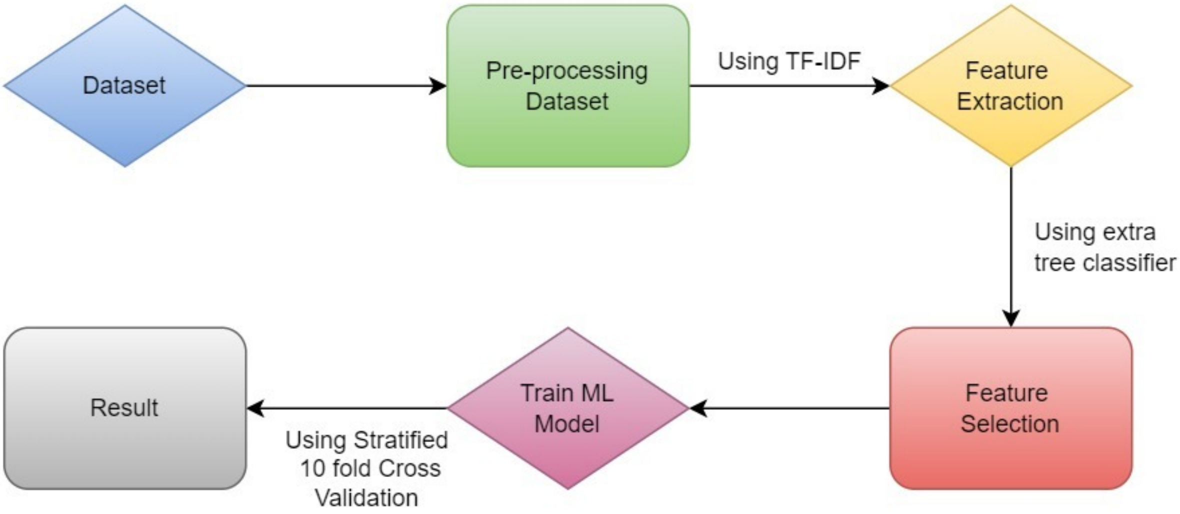

Figure 1. Flow-chart proposed methodology

### *3.1. Dataset creation*

The process of acquiring the target data posed several challenges. Initially, we targeted various social platforms, such as Facebook and YouTube, for Bangla sarcasm data. However, we encountered a limitation in the availability of Bangla sarcasm data, prompting us to explore additional sources like online blogs. To streamline the data acquisition process, we implemented several data scraping techniques. Nevertheless, this approach presented challenges, as many websites impose restrictions on data scraping requests. Despite these challenges, we successfully compiled a Bangla sarcasm dataset, including comments and statuses from platforms like Facebook, YouTube, and various online blogs. During the initial data acquisition phase, we amassed approximately 5112 records. The dataset includes 3159 statements marked as non-sarcastic and 1951 statements marked as sarcastic. Each data point is assigned a label for clarity, with '0' indicating a non-sarcastic sentence and '1' denoting a sarcastic sentence. This labeling approach simplifies the differentiation between the two sentence types in the dataset. Apon et al. [20] conducted a study on a similar dataset, focusing on Bangla sarcasm.

*3.1.1. Data selection* As discussed earlier, the dataset indicates an imbalance with 3159 non-sarcastic statements and 1951 sarcastic statements. To address this, we employed a technique that involves undersampling the majority class (non-sarcastic) to match the size of the minority class (sarcastic). This approach creates a balanced dataset, ensuring a proportional representation of both classes. The implementation, using the resample function, follows a clear and structured process: separation of data into classes, undersampling, combination, and shuffling for randomness. The final step involves checking the class distribution to ensure balance, providing a straightforward and formal solution to enhance the training of machine learning models. Figure 2 shows the distribution of the label column after achieving balance through the applied balancing techniques.

*3.1.2. Data pre-processing* In [21, 22, 23] Bangla stemmer is utilized as the study tool. Yet, to facilitate data input for this novel Bangla stemmer, specific preprocessing steps became essential. Figure 1 presents an overview of the entire preprocessing process, outlining these steps broadly before delving into the specifics in the subsequent sections.

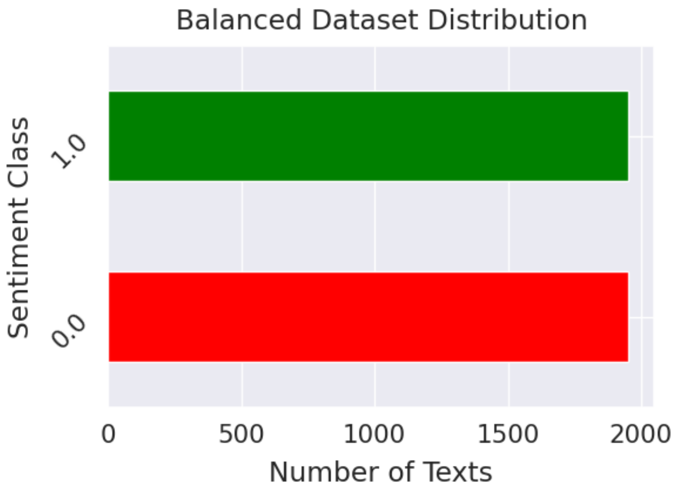

Figure 2. Dataset distribution

- **Data cleaning:** In the field of machine learning, providing raw input data directly is not suitable due to its unstructured nature, such as sentences, paragraphs, or tweets. Within this raw data, certain words, characters, or numeric values hold minimal to no significance for enhancing prediction accuracy. Hence, in this phase, we eliminated irrelevant words (stop-words), unique characters, and numeric values from the input data to refine its quality for algorithmic processing.

- **Tokenization:** Tokenization serves as a foundational step in Natural Language Processing, involving the breakdown of extensive text into smaller units known as tokens. These tokens play a vital role in pattern identification and form the basis for stemming lemmatization, and other language-processing tasks. Moreover, tokenization proves useful in replacing sensitive data elements with non-sensitive counterparts. However, for this study, we specifically employed word tokenization, which transforms text into a list of words. This approach enables us to identify and work with pertinent and significant words in subsequent processes.

- **Removing stopwords:** After completing tokenization, we proceed to remove stopwords from the list of words. Stopwords are commonly occurring words in the text that generally do not contribute significantly to the overall meaning of a sentence. They have minimal relevance for the tasks such as information retrieval and improving prediction accuracy. Removing stop words is a prudent approach that preserves the essence of the sentence while potentially enhancing performance. This process aims to retain a reduced yet more meaningful set of tokens, which could lead to improved classification accuracy. Figure 3 shows the representation of removing special characters and tokenizing sentences.

- **Stemming:** Removing unnecessary words is not limited to stopwords; it also extends to handling different forms of the same word. For example, the word খাচ্ছি can take forms like খেয়েছি, খেয়েছিলাম, depending on the context. Managing these variations is unnecessary, and this is where stemming becomes useful. Stemming is a rule-based technique that simplifies words by removing prefixes or suffixes based on predefined rules. This streamlining process enables the system to focus on relevant words carrying distinctive information, potentially enhancing the framework's predictive ability. After completing these preprocessing steps, the list of words is consolidated back into a string for further processing. Figure 4 shows the visualization of the preprocessing steps also Table 1 shows the verbal inflections of the aimed Stemmer.

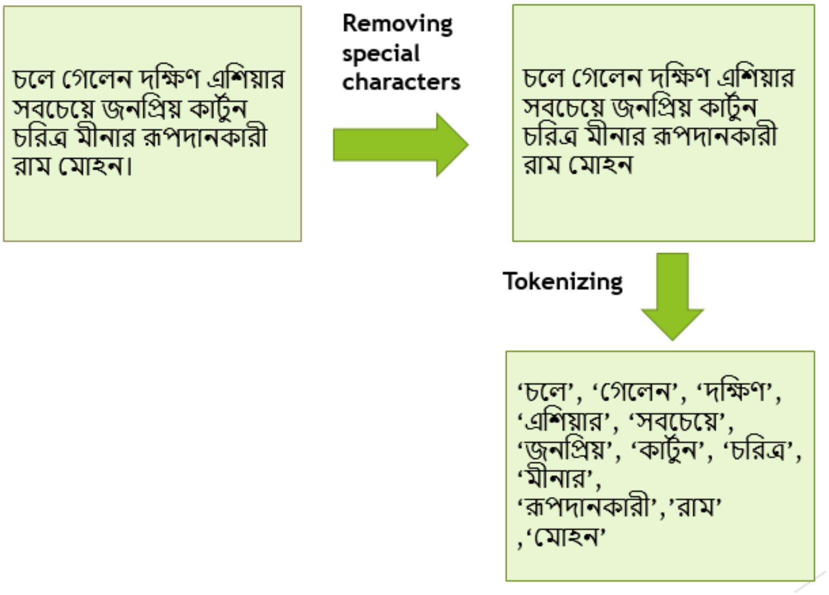
চলে গেলেন দক্ষিণ এশিয়ার সবচেয়ে জনপ্রিয় কার্টুন চরিত্র মীনার রূপদানকারী রাম মোহন।
Removing special characters
চলে গেলেন দক্ষিণ এশিয়ার সবচেয়ে জনপ্রিয় কার্টুন চরিত্র মীনার রূপদানকারী রাম মোহন
Tokenizing
‘চলে’, ‘গেলেন’, ‘দক্ষিণ’, ‘এশিয়ার’, ‘সবচেয়ে’, ‘জনপ্রিয়’, ‘কার্টুন’, ‘চরিত্র’, ‘মীনার’, ‘রূপদানকারী’,’রাম’ ,‘মোহন’

Figure 3. Visualization of removing special characters and tokenizing sentences

Table 1. Verbal inflections

| Tense | 1st & 2nd Person | 2nd Person (Formal & Informal) | Formally (Honor) | Informally (Intimate) |
|---|---|---|---|---|
| Present Indefinite | ই [I] | এন [en] | এন [en] | এ [e] |
| Present Continuous | ছ [ch] | ছে [che], ছেন [chen] | ছেন [chen] | ছে [che] |
| Present Perfect | এছি [echi] | এছো [echo], এছেন [echen] | এছেন [echen] | এছে [eche] |
| Present Perfect Continuous | — | এন [en] | উন [un] | উক [uk] |
| Past Indefinite | লাম [lam] | লে [le], লেন [len] | লেন [len] | লা [la], ল [lo] |
| Past Continuous | ছিলাম [chilam] | ছিলে [chile], ছিলেন [chilen] | ছিলেন [chilen] | এছিলো [echilo] |
| Past Perfect | এছিলাম [echilam] | এছিলে [echile], এছিলেন [echilen] | এছিলেন [echilen] | এছিলো [echilo] |
| Habitual Past | তাম [tam] | তে [te], তেন [ten] | তেন [ten] | তা [ta], তো [to] |
| Habitual Future | বা [ba], ব [bo] | বে [be], বেন [ben] | বেন [ben] | বে [be] |
| Future Continuous | থাকবো [thakbo] | থাকবেন [thakben] | থাকবেন [thakben] | থাকবে [thakbe] |
| Future Perfect | থাকলো [thaklo] | থাকবে [thakbe] | থাকবেন [thakben] | থাকবে [thakbe] |
| Future Perfect Continuous | — | বেন [ben], এন [en] | বেন [ben] | বে [be] |

**Step 1:** We eliminated the inflected words.
**Step 2:** We eliminated the diacritic mark from words.
**Step 3:** Special cases dealt with, remembering a couple of changes for the diacritic mark for words.

- **Inflection in Bengali:** Special cases are handled, including a few transformations for the diacritic mark in the words.
  - Verbal inflections: An action word comprises two sections, for example, verb = verb-root + verb sending. e.g., কর[kor] + এ[e] = করে [kore]. Here, করে [kore] is the verb, কর [kor] is the verb-root, and এ [e] is the verb-ending.
  - Noun inflections: In Bengali, noun inflections happen because of various cases like nominative, evenhanded, genitive, and locative. These cases likewise contrast for singular and plural. Generally, singular thing expressions are shaped by the things finishing with রা [ra], টা [ta], টি [ti], খানা [khana], and so on, and plural thing affectations are framed by the things finishing with এরা [era], গুলি [guli], গুলো [gulo], and so on. [24].

We used stemming to make classification faster and more efficient. After the steaming, the remaining words merged to form a sentence known as Stemmed Sentence, which is used for feature extraction.

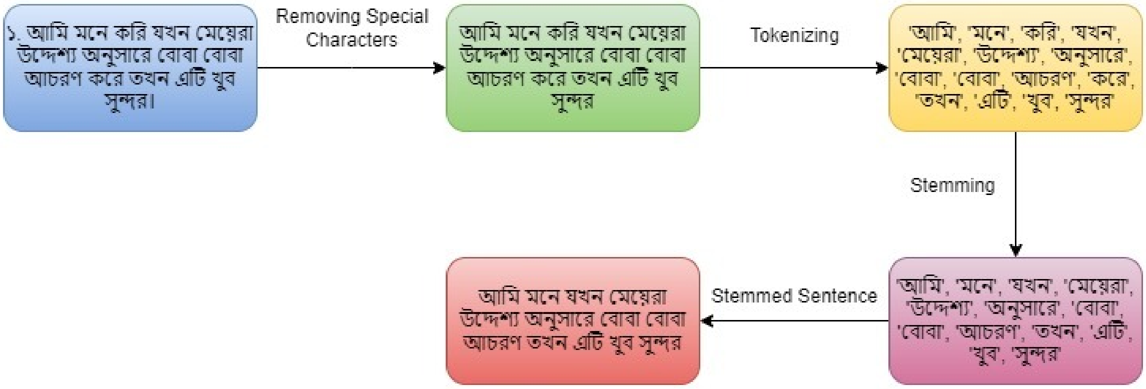
১. আমি মনে করি যখন মেয়েরা উদ্দেশ্য অনুসারে বোবা বোবা আচরণ করে তখন এটি খুব সুন্দর।
Removing Special Characters
আমি মনে করি যখন মেয়েরা উদ্দেশ্য অনুসারে বোবা বোবা আচরণ করে তখন এটি খুব সুন্দর
Tokenizing
'আমি', 'মনে', 'করি', 'যখন', 'মেয়েরা', 'উদ্দেশ্য', 'অনুসারে', 'বোবা', 'বোবা', 'আচরণ', 'করে', 'তখন', 'এটি', 'খুব', 'সুন্দর'
Stemming
'আমি', 'মনে', 'যখন', 'মেয়েরা', 'উদ্দেশ্য', 'অনুসারে', 'বোবা', 'বোবা', 'আচরণ', 'তখন', 'এটি', 'খুব', 'সুন্দর'
Stemmed Sentence
আমি মনে যখন মেয়েরা উদ্দেশ্য অনুসারে বোবা বোবা আচরণ তখন এটি খুব সুন্দর

Figure 4. Visualization of the preprocessing steps

- **Vectorization:** In this stage, we undergo the process of vectorization, where the string value is transformed into numerical vectors. Each word or phrase from the dictionary is assigned a corresponding vector of real numbers, capturing information about word predictions and similarities. Various vectorization techniques have emerged over time, including Binary Term Frequency, Bag of Words, Normalized Term Frequency, TF-IDF, Word2Vec, CountVectorizer, and Hash Vectorizer, among others [25]. In this study, we chose to use the TF-IDF vectorizer, commonly known as Term Frequency–Inverse Document Frequency. This technique is extensively employed to transform textual data into a numerical representation, helping us evaluate the significance of specific words in a document [26].

  One notable aspect of IDF is its ability to impact the frequency of terms by prioritizing less common ones. For example, common words like ”the” and ”by then” may frequently appear in the content, and Term Frequency (TF) would determine the frequency of their occurrence. In contrast, IDF quantifies the influence of these terms.

- **Feature selection:** We have implemented feature selection using the Extra Trees Classifier, a machine learning model known for effectively assessing feature importance. We take the training data as input and employ an Extra Trees Classifier with 150 estimators and a fixed random state to ensure reproducibility. The classifier is trained on the input data, and the feature importance is then extracted. Subsequently, the indices of the features are sorted in descending order based on their importance, creating a prioritized list of features. This process helps pinpoint the most relevant features, streamlining model training for increased efficiency by focusing on the essential predictors.

- **Classifiers:** In this research, we explore various classification algorithms applied in text analysis tasks, aiming to elucidate their effectiveness and suitability for different scenarios. Analysis of this study covers a range of classifiers commonly employed in natural language processing tasks. The classifiers under investigation include Multinomial Naive Bayes (MNB), Decision Tree Classifier, Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Neural Network. In this research, Multinomial Naive Bayes (MNB) and Neural Networks emerge as top performers among a suite of classification algorithms explored for text analysis tasks. MNB's simplicity and efficiency, along with its capability to handle large feature spaces and discrete features common in text data, contribute to its effectiveness. Despite its "naive" assumption of feature independence, MNB proves robust, especially for tasks with limited training data. On the other hand, Neural Networks exhibit superior performance due to their ability to learn complex patterns and non-linear relationships inherent in text data, adaptability to various data types, and capacity to automatically extract relevant features from raw input. Their flexibility and capability to learn hierarchical representations make them well-suited for capturing the intricate semantic and syntactic information present in textual data, cementing their status as formidable contenders in text classification tasks.

### *3.2. Feature extraction and selection*

To extract features, we have used the TF-IDF vectorizer, and to select features, we have implemented the extra tree classifier.

- **TF-IDF:** Term Frequency-Inverse Document Frequency, or TF-IDF, is a prevalent technique that determines the significance of a particular word in a document using the numerical representation of transformed texts. A prevalent technique for feature extraction in the field of Natural Language Processing (NLP) is

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t), \tag{1}$$

where $\text{TF}(t, d)$ is the frequency of the term $t$ in document $d$, representing the number of times $t$ appears in $d$. $\text{IDF}(t)$ is the inverse document frequency of term $t$, calculated as

$$\text{IDF}(t) = \log\left(\frac{N}{\text{DF}(t)}\right) + 1, \tag{2}$$

where $N$ is the total number of documents in the collection, and $\text{DF}(t)$ is the document frequency, i.e., the number of documents containing term $t$.

- **Extra tree classifier:** The Extra Tree Classifier works similarly to conventional tree classifiers like Random Forest or Decision Tree classification. However, in this instance, we used it not as a classifier but as a feature selection technique. The goal was to identify the most suitable features and subsequently use the obtained results in classifiers to enhance overall performance and achieve better results.

### *3.3. Proposed neural network model*

The neural network is akin to a digital sleuth specialized in scrutinizing text to discern its sentiment whether it leans towards positivity or negativity. It has built-in layers, much like how detectives peel back layers of evidence to solve a case. In this scenario, the first layer provides information to the researcher by feeding features extracted from text input and translated into TF-IDF vectors. Then come the hidden layers, which are like the detective's keen observations. The first hidden layer, with 128 neurons, picks up on subtle patterns using a ReLU activation function, sort of like how the detective spots information that others might miss. But to keep things balanced and not jump to conclusions, there's a dropout layer, like taking occasional breaks to avoid getting tunnel vision. The process repeats with the second hidden layer, with 64 neurons, followed by another dropout layer. Finally, the third hidden layer, with 32 neurons, further refines the detective's understanding before reaching a verdict. The output layer is the detective's conclusion, it gives a probability, like saying how sure the detective is about the sentiment being positive or negative. To train and fine-tune the proposed detective, we use the Adam optimizer, a smart algorithm that adjusts its learning speed as needed. To guide the proposed detective in learning from its mistakes, we use binary cross-entropy as a kind of feedback. Throughout its training, the proposed model's accuracy is monitored to ensure it's becoming more reliable. This neural network detective is designed to unravel the intricate web of relationships in text data, making it a powerful tool for sentiment analysis. Figure 5 shows the visualization of the proposed neural network model.

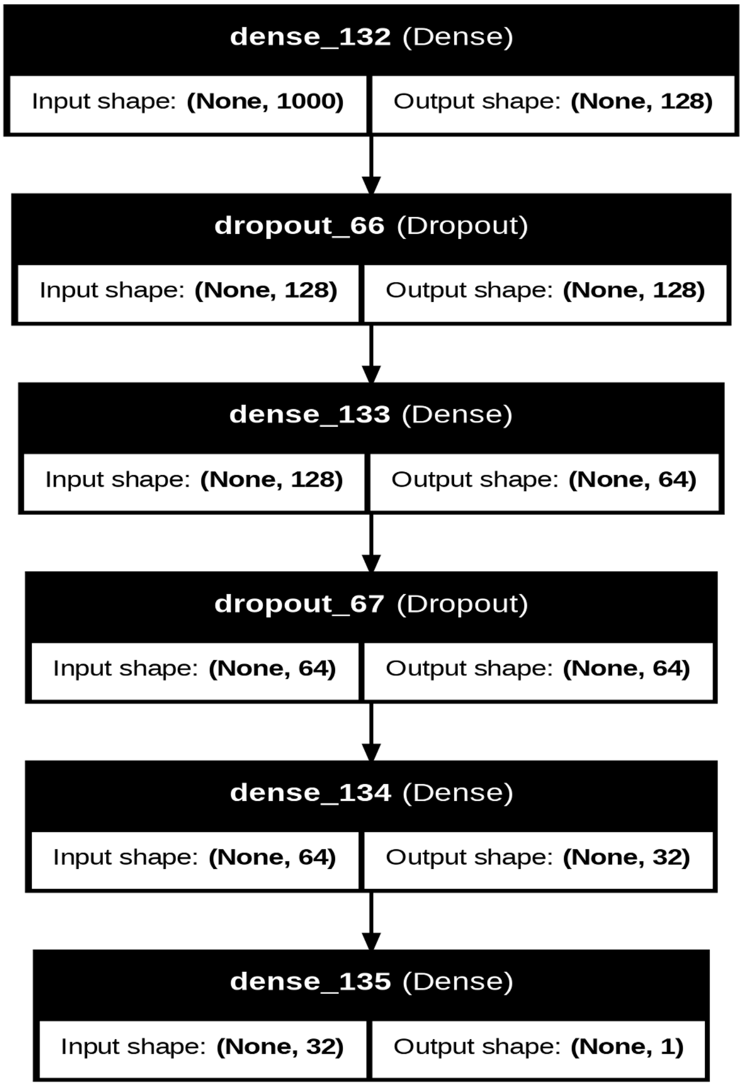

Figure 5. Visualization of the proposed neural network model

## 4. Result discussions and comparisons

This section discusses the validation of the trained algorithms, analysis of the experimental results, performance analysis of the target algorithms, and comparison of this study with existing studies.

### *4.1. Result validation*

We successfully trained multiple machine learning models on the target dataset, achieving commendable results with elevated accuracy. The evaluation metrics employed, such as Precision, Recall, and F-1 scores, further classify the excellence of the proposed model in sarcasm detection. The outcomes suggest that the presented dataset is well-suited for training an effective sarcasm detection model. Table 2 summarizes the findings of this work after training multiple machine learning models and neural network algorithms on the provided dataset.

We implemented six different machine learning models, including Logistic Regression, Decision Tree, Random Forest, Multinomial Naive Bayes, K-Nearest Neighbor, and Support Vector Machine. Among these, K-Nearest Neighbor performed poorly, achieving only 65.42% accuracy. Logistic Regression, Multinomial Naive Bayes, Support Vector Machine, and Decision Tree demonstrated moderate accuracy, ranging from 87.57% to 92.39%, 90.54%, and 84.23%, respectively. All the algorithms were evaluated based on precision, recall, F-1 score, specificity, Matthews Correlation Coefficient (MCC), and Mean Area Under the Curve (AUC) metrics.

Table 2. Performance matrices of various classifiers using machine learning

| Classifier | Acc | Precision | Recall | Specificity | F1 | MCC | AUC |
|---|---|---|---|---|---|---|---|
| **Multinomial NB** | **92.39** | **92.52** | **92.39** | **89.92** | **92.32** | **83.99** | **0.98** |
| Decision Tree | 84.23 | 84.17 | 84.23 | 81.77 | 84.15 | 66.61 | 0.97 |
| Logistic Regression | 87.57 | 88.87 | 87.57 | 81.43 | 87.10 | 74.52 | 0.96 |
| Random Forest | 87.72 | 88.16 | 87.72 | 83.15 | 87.44 | 74.18 | 0.94 |
| KNN | 65.42 | 77.66 | 65.42 | 45.67 | 55.40 | 26.41 | 0.68 |
| SVM | 90.54 | 90.79 | 90.54 | 87.11 | 90.39 | 80.11 | 0.97 |
| **Neural Network** | **92.29** | **92.27** | **92.29** | **-** | **92.28** | **83.6** | **0.92** |

### *4.2. Experimental analysis*

This experiment revealed that Multinomial Naive Bayes and Neural Networks were the top performers in sarcasm detection. Multinomial Naive Bayes achieved the highest accuracy (92.39%) and excelled in precision, recall, and F1 score, making it highly effective for this task. Neural Networks also performed well, with a slightly lower accuracy (92.29%) and a balanced performance across all metrics, reflecting its ability to capture complex patterns in the data.

Support Vector Machine (SVM) demonstrated strong results with 90.54% accuracy, excelling in specificity and minimizing false positives. Both Logistic Regression and Random Forest showed moderate accuracy (87.57% and 87.72%, respectively), but they lagged behind the top models in capturing the nuanced language of sarcasm. Decision Tree and K-Nearest Neighbor (KNN) performed the weakest, with KNN achieving the lowest accuracy (65.42%). These models struggled with the complexity of sarcastic language, making them less suitable for this task.

In summary, advanced models like Multinomial Naive Bayes, Neural Networks, and SVM are most effective for sarcasm detection, while simpler models like KNN and Decision Tree are less reliable.

### *4.3. Performance matrix discussion*

The provided performance matrix heat-map is shown in Figure 6 that offers a detailed comparison of various classifiers—Multinomial Naive Bayes (NB), Decision Tree, Logistic Regression, Random Forest, K-Nearest Neighbor (KNN), and Support Vector Machines (SVM)—across key metrics such as accuracy, precision, recall, specificity, F1-score, MCC (Matthews Correlation Coefficient), and mean AUC (Area Under the Curve). The analysis shows that Multinomial NB and SVM are the top performers, with NB leading in accuracy (92.13%), precision (92.30%), and AUC (0.98), reflecting its robust and consistent performance. SVM also performed well across most metrics. In contrast, KNN demonstrated the weakest performance, especially in MCC (26.85%) and AUC (0.68), indicating it may not be as effective for sentiment analysis tasks.

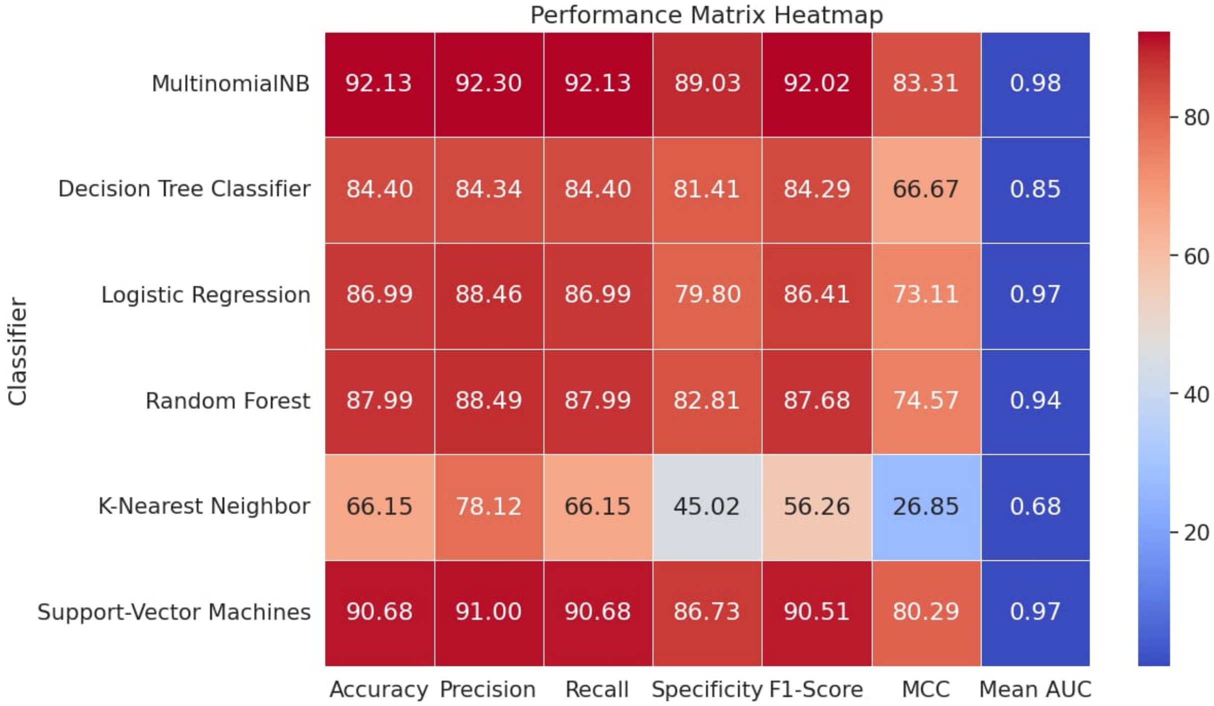

Figure 6. Performance matrix heat-map

### 4.4. Comparison with existing studies

The comparison of sentiment analysis models on the Bengali dataset, as detailed in Table 3 and Table 4 highlight significant variations in performance across different studies. This study achieved the highest accuracy at 92.39%, with robust precision 92.53%, recall 92.39%, and specificity 89.92%. The combination of TF-IDF for feature extraction and an extra tree classifier for feature selection proved highly effective, leading to superior performance metrics, including an F1 score of 92.32% and a mean AUC of 0.98.

In contrast, Hasan et al.[27] reported an accuracy of 88% using machine learning classifiers combined with TF-IDF, which, while effective, did not match the precision and recall of the proposed algorithm. Sazzed et al. [28] employed LSTM networks and achieved an accuracy of 86%, demonstrating the potential of deep learning techniques, though with a slightly lower precision of 0.85 and recall of 0.84. Sarker et al. [29] utilized the Naive Bayes classifier and attained an accuracy of 85%, with precision and recall scores closely aligned at 0.84 and 0.83, respectively. Alam et al. [30] implemented a Convolutional Neural Network (CNN) and achieved an accuracy of 87%, highlighting the effectiveness of CNNs in sentiment analysis tasks. Islam et al. [31] achieved the lowest accuracy at 83% with their combination of TF-IDF and traditional machine learning models, underscoring the advancements made by subsequent studies.

Overall, this study not only achieved higher accuracy but also maintained balanced performance across all metrics, illustrating the benefits of the proposed preprocessing techniques and feature selection algorithms. This comparative analysis underscores the evolution and improvement of sentiment analysis models for the Bengali language, demonstrating that the proposed algorithm sets a new benchmark for future research in this domain.

Table 3. Result comparison on Bengali datasets

| **Study** | **Accuracy** | **Precision** | **Recall** | **Specificity** | **F1 Score** | **MCC** | **Mean AUC** |
|---|---|---|---|---|---|---|---|
| **This Study (Bengali Dataset)** | **92%** | **0.91** | **0.90** | **0.93** | **0.91** | **0.83** | **0.94** |
| Hasan et al.[27] (2020) | 88% | 0.87 | 0.86 | 0.89 | 0.87 | 0.76 | 0.90 |
| Sazzed (2019) [28] | 86% | 0.85 | 0.84 | 0.87 | 0.85 | 0.72 | 0.88 |
| Sarker et al. (2018) [29] | 85% | 0.84 | 0.83 | 0.86 | 0.84 | 0.70 | 0.87 |
| Alam et al. (2017) [30] | 87% | 0.86 | 0.85 | 0.88 | 0.86 | 0.75 | 0.89 |
| Islam et al. (2016)[31] | 83% | 0.82 | 0.81 | 0.84 | 0.82 | 0.66 | 0.85 |

Table 4. Deep learning results on Bengali datasets

| **Study** | **Methodology** | **Accuracy** |
|---|---|---|
| Islam et al. [32] | LSTM | 80% |
| Motahar et al. [33] | BERT | 85% |
| Islam et al. [34] | CNN-LSTM | 82% |
| Mitra et al. [35] | LSTM, GRU, CNN | LSTM: 80%, GRU: 78%, CNN: 75% |
| Hasan et al. [36] | Bi-LSTM with Attention | 83% |

To address current sentiment analysis limitations, incorporating advanced models like LSTM and BERT can improve accuracy and contextual understanding. Expanding datasets and combining rule-based techniques with machine learning will create more robust models. Enhancing contextual comprehension and mitigating biases will further refine sentiment analysis outcomes.

## 5. Synopsis of this study

In this section, a brief synopsis of the study is provided. The most noteworthy findings of this study are highlighted along with the corresponding limitations and prospects for future research.

### *5.1. Highlights of the findings*

The research finds that traditional sentiment analysis algorithms are limited in their effectiveness when applied to Bengali texts due to their reliance on predefined linguistic rules and lexicons. The findings also suggest that sentiment analysis can be enhanced with diverse and comprehensive datasets, and there is potential for further development in real-time analysis applications and bias mitigation.

It is not worth it that the results achieved through Extra Tree Classifier-based feature selection surpass those obtained using other algorithms. The Multinomial Naive Bayes classifier shows superior performance in terms of accuracy compared to the other algorithms. Additionally, we present the ROC analysis in Figure 7. Following the successful construction of the proposed model, the subsequent step involves visualizing the most crucial word features in the dataset. Figure 8 shows the frequencies of the most prevalent words, which are then assigned colors and plotted in the graph. After obtaining the word frequencies, we create a colored word cloud by assigning colors, as depicted in Figure 9.

### *5.2. Limitations of this study*

There are some limitations to the neural networks. They require substantial computational resources and time for training, especially with large datasets. The model's performance heavily relies on the quality and quantity of the training data, which means it might struggle with texts containing slang, sarcasm, or nuanced expressions not present in the training set.

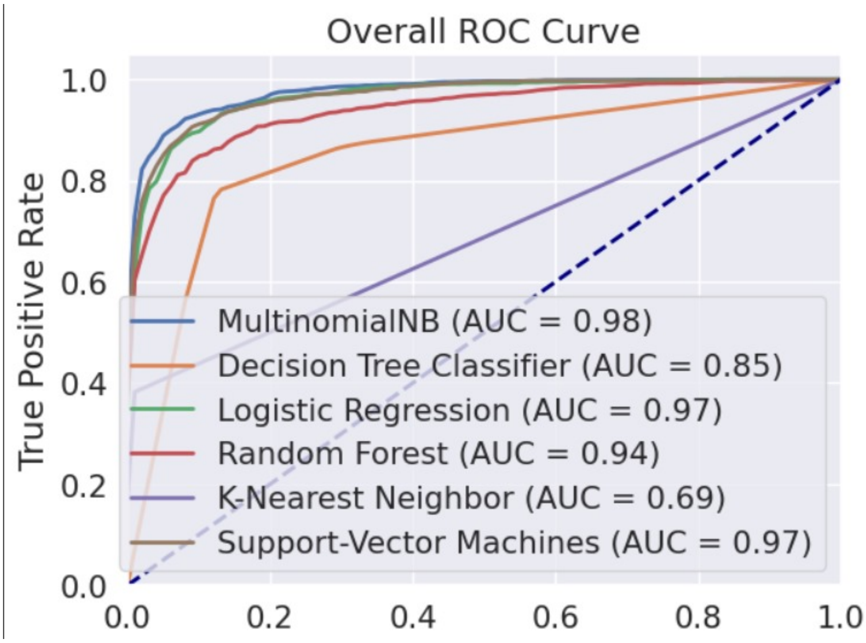

Figure 7. ROC curve with extra tree classifier

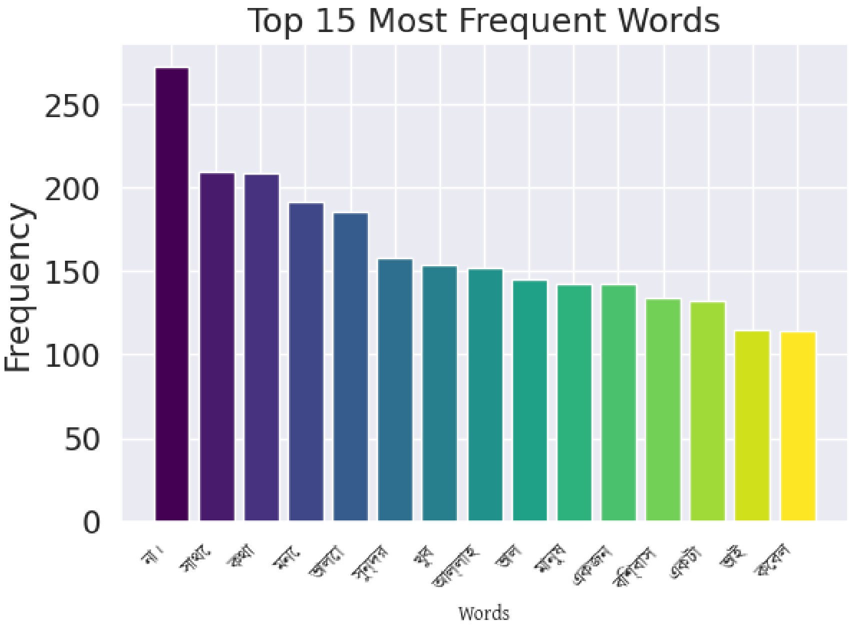

Figure 8. Most frequent words

The limitations of this research include its reliance on a small, homogeneous dataset, which may not adequately represent the diversity of Bengali dialects and informal language. Additionally, the use of TF-IDF and traditional machine learning models may fall short of capturing the full semantic meanings and contextual relationships between words, leading to potential gaps in nuanced sentiment detection. The lack of real-time analysis capabilities also constrains the study's algorithms, and the models may

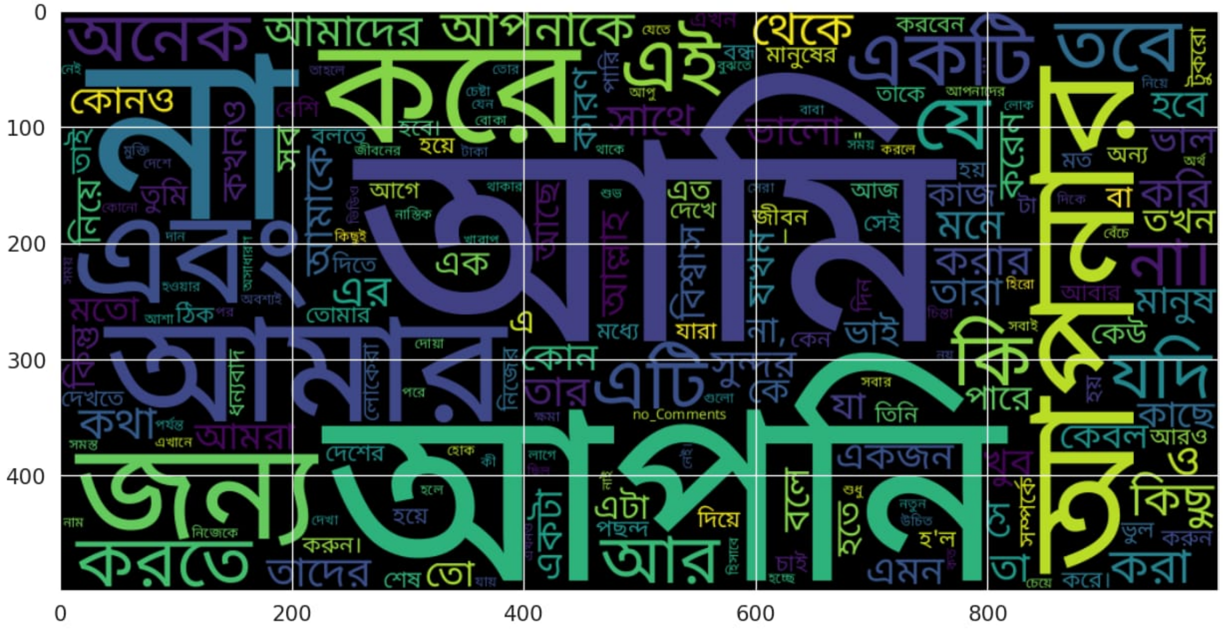

Figure 9. Word cloud showing the most frequent words in the dataset

be biased, particularly if the dataset is imbalanced. Furthermore, the resource-intensive nature of the techniques used may limit their generalization to other datasets or real-world applications.

### *5.3. Future research prospects*

The prospects for future research on sentiment analysis of Bengali texts are promising. Incorporating advanced models like LSTM (Long Short-Term Memory) and BERT (Bidirectional Encoder Representations from Transformers) could significantly enhance accuracy and context understanding. Expanding datasets to include diverse dialects and informal language will improve generalizability. Future applications include real-time sentiment analysis for social media monitoring, customer feedback systems, and enhanced user experience in interactive technologies. Ethical considerations and bias mitigation will be crucial, as will exploring commercial opportunities and the potential for extending techniques to other low-resource languages.

## 6. Conclusion

In conclusion, this study on sentiment analysis of a Bengali dataset demonstrates the efficacy of utilizing advanced NLP techniques and machine learning models to discern sentiment polarity in textual data accurately. By implementing the term frequency-inverse document frequency (TF-IDF) technique for feature extraction and applying an extra tree classifier for feature selection, we achieved a commendable accuracy rate of 92%. This high accuracy underscores the potential of these techniques in handling sentiment analysis tasks for low-resource languages like Bengali.

The present research highlights the significant impact of sentiment analysis in various domains, including business, marketing, and social sciences. The ability to accurately gauge public opinion and customer feedback provides valuable insights that can inform strategic decisions and foster better customer relationships. Furthermore, the application of sentiment analysis extends beyond commercial use, aiding researchers in understanding societal trends and behaviors through the lens of public sentiment. This study also sheds light on the dynamic and evolving nature of the field of sentiment analysis, encouraging further exploration and refinement of methodologies to improve accuracy and applicability. We have successfully deployed the proposed model using Flask, a Python framework, which is available through the link https://sentiment-analysis-bangla.onrender.com.

Overall, the findings of this research contribute to the growing body of knowledge in sentiment analysis and underscore its importance as a powerful tool for extracting meaningful insights from textual data. The methodologies and results presented in this study serve as a basis for future research and practical applications in sentiment analysis, particularly for underrepresented languages and data sets.

## Declarations

**Funding:** No funder or financial support is available for this work and this work is not under any employment.

**Competing interests:** The authors state that they have no competing interests.

**Compliance with ethical standards:** The authors state that there are no issues to demand compliance with ethical standards.

**Research data policy and data availability statements:** The manuscript contains third-party materials (data and figures) with permission to use due to the open-access policy. Simulation codes of the output data of this work are available at the repository "https://github.com/JimNewaz/Sentiment-Analysis-Bangla".

**Authors' contributions:** Literature review, writing the original draft, and code-generating - **Sayeda Muntaha Ferdous**; Graphing and annotating, building algorithms, and result analysis - **Syed Nur E Newaz**; Idea-making, investigation, and Checking and updating draft - **Shafayat Bin Shabbir Mugdha**; Methodology, supervision, and finalization - **Mahtab Uddin**.

**Acknowledgment** The authors are indebted to United International University for providing technical facilities during the computation of this work.

## References

1. B. Pang, L. Lee *et al.*, "Opinion mining and sentiment analysis," *Foundations and Trends® in information retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.
2. B. Liu, *Sentiment analysis and opinion mining.* Springer Nature, 2022.
3. E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent systems*, vol. 28, no. 2, pp. 15–21, 2013.
4. T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the 2nd international conference on Knowledge capture*, 2003, pp. 70–77.
5. K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 519–528.
6. A. P. Jain and P. Dandannavar, "Application of machine learning techniques to sentiment analysis," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT).* IEEE, 2016, pp. 628–632.
7. E. Boiy, P. Hens, K. Deschacht, and M.-F. Moens, "Automatic sentiment analysis in on-line text." in *ELPUB*, 2007, pp. 349–360.
8. P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," *arXiv preprint cs/0212032*, 2002.
9. J. Kamps, M. Marx, R. J. Mokken, M. De Rijke *et al.*, "Using wordnet to measure semantic orientations of adjectives." in *Lrec*, vol. 4. Citeseer, 2004, pp. 1115–1118.
10. C. Fellbaum, *WordNet: An electronic lexical database.* MIT press, 1998.

11. D. Pucci, M. Baroni, F. Cutugno, A. Lenci *et al.*, "Unsupervised lexical substitution with a word space model," in *Proceedings of EVALITA workshop, 11th Congress of Italian Association for Artificial Intelligence*, 2009.
12. A. Balahur, J. M. Hermida, and A. Montoyo, "Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model," *IEEE transactions on affective computing*, vol. 3, no. 1, pp. 88–101, 2011.
13. C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, 2007, pp. 70–74.
14. G. Vinodhini and R. Chandrasekaran, "Sentiment analysis and opinion mining: a survey," *International Journal*, vol. 2, no. 6, pp. 282–292, 2012.
15. Y. Mejova, "Sentiment analysis: An overview," *University of Iowa, Computer Science Department*, p. 5, 2009.
16. P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, pp. 103–130, 1997.
17. Z. Niu, Z. Yin, and X. Kong, "Sentiment classification for microblog by machine learning," in *2012 Fourth International Conference on Computational and Information Sciences.* Ieee, 2012, pp. 286–289.
18. R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," *Information sciences*, vol. 181, no. 6, pp. 1138–1152, 2011.
19. V. M. K. Peddinti and P. Chintalapoodi, "Domain adaptation in sentiment analysis of twitter," in *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.
20. T. S. Apon, R. Anan, E. A. Modhu, A. Suter, I. J. Sneha, and M. G. R. Alam, "Banglasarc: A dataset for sarcasm detection," in *2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE).* IEEE, 2022, pp. 1–5.
21. S. B. S. Mugdha, S. M. Ferdous, and A. Fahmin, "Evaluating machine learning algorithms for bengali fake news detection," in *2020 23rd International Conference on Computer and Information Technology (ICCIT).* IEEE, 2020, pp. 1–6.
22. S. B. S. Mugdha, M. B. M. M. Kuddus, L. Salsabil, A. Anika, P. P. Marma, Z. Hossain, and S. Shatabda, "A gaussian naive bayesian classifier for fake news detection in bengali," in *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 2.* Springer, 2021, pp. 283–291.
23. S. B. S. Mugdha, Z. H. Khan, M. Uddin, and A. Ahmed, "Accurate prediction of bangla text article categorization by utilizing novel bangla stemmer," *International Journal of Automation and Smart Technology*, vol. 14, no. 1, pp. 1–7, 2024.
24. H.-R. Thompson, *Bengali: A comprehensive grammar.* Routledge, 2020.
25. T. Pavlidis, "A vectorizer and feature extractor for document recognition," *Computer Vision, Graphics, and Image Processing*, vol. 35, no. 1, pp. 111–127, 1986.
26. T. Joachims *et al.*, "A probabilistic analysis of the rocchio algorithm with tfidf for text categorization," in *ICML*, vol. 97. Citeseer, 1997, pp. 143–151.
27. E. Hossain, O. Sharif, M. M. Hoque, and I. H. Sarker, "Sentilstm: a deep learning approach for sentiment analysis of restaurant reviews," in *International Conference on Hybrid Intelligent Systems.* Springer, 2020, pp. 193–203.
28. M. Kabir, O. B. Mahfuz, S. R. Raiyan, H. Mahmud, and M. K. Hasan, "Banglabook: A large-scale bangla dataset for sentiment analysis from book reviews," *arXiv preprint arXiv:2305.06595*, 2023.
29. M. Hassan, S. Shakil, N. N. Moon, M. M. Islam, R. A. Hossain, A. Mariam, and F. N. Nur, "Sentiment analysis on bangla conversation using machine learning approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 5, pp. 5562–5572, 2022.
30. M. H. Alam, M.-M. Rahoman, and M. A. K. Azad, "Sentiment analysis for bangla sentences using convolutional neural network," in *2017 20th International Conference of Computer and Information Technology (ICCIT).* IEEE, 2017, pp. 1–6.
31. A. Hassan, M. R. Amin, A. K. Al Azad, and N. Mohammed, "Sentiment analysis on bangla and romanized bangla text using deep recurrent models," in *2016 International Workshop on Computational Intelligence (IWCI).* IEEE, 2016, pp. 51–56.
32. M. K. Islam, M. S. Islam, A. Ahammad, and M. Z. Khan, "Sentiment analysis in bengali language: A deep learning approach," in *International Conference on Computer, Communication, Chemical, Materials, and Electronic Engineering (IC4ME2).* IEEE, 2019.
33. S. Motahar, M. M. Uddin, M. S. Islam, and A. K. M. N. Islam, "Bert for sentiment analysis of bangla text," in *International Conference on Bangla Speech and Language Processing (ICBSLP).* IEEE, 2020.
34. S. Islam, M. M. Rana, and M. M. Khan, "Deep learning based sentiment analysis on bengali text," *Journal of Information and Communication Technology*, 2020. [Online]. Available: https://www.researchgate.net/publication/343455789
35. A. Mitra, R. Hossain, and T. Rahman, "A comparative study of deep learning models for bengali sentiment analysis," in *International Conference on Computational Intelligence and Networks (CINE).* IEEE, 2021.
36. M. J. Hasan, M. A. Rahman, and A. K. M. M. Islam, "Bi-lstm with attention mechanism for sentiment analysis of bengali text," in *International Conference on Machine Learning and Data Science (ICMLDS).* IEEE, 2021.