Natural Language Processing Journal 7 (2024) 100069

Contents lists available at ScienceDirect

Natural Language Processing Journal

journal homepage: www.elsevier.com/locate/nlp

Sentiment analysis of Bangla language using a new comprehensive dataset BangDSA and the novel feature metric skipBangla-BERT

Md. Shymon Islam *, Kazi Masudul Alam

Khulna University, Khulna 9208, Bangladesh

ARTICLE INFO

Keywords:
Sentiment analysis
Bangla dataset
Bangla-BERT
Skipgram
CNN
Bi-LSTM

ABSTRACT

In this modern technologically advanced world, Sentiment Analysis (SA) is a very important topic in every language due to its various trendy applications. But SA in Bangla language is still in a dearth level. This work focuses on examining different hybrid feature extraction techniques and learning algorithms on **Bangl**a **D**ocument level **S**entiment **A**nalysis using a new comprehensive dataset (BangDSA) of 203,493 comments collected from various microblogging sites. The proposed BangDSA dataset approximately follows the Zipf's law, covering 32.84% function words with a vocabulary growth rate of 0.053, tagged both on 15 and 3 categories. In this study, we have implemented 21 different hybrid feature extraction methods including Bag of Words (BOW), N-gram, TF-IDF, TF-IDF-ICF, Word2Vec, FastText, GloVe, Bangla-BERT etc with CBOW and Skipgram mechanisms. The proposed novel method (Bangla-BERT+Skipgram), skipBangla-BERT outperforms all other feature extraction techniques in machine leaning (ML), ensemble learning (EL) and deep learning (DL) approaches. Among the built models from ML, EL and DL domains the hybrid method CNN-BiLSTM surpasses the others. The best acquired accuracy for the CNN-BiLSTM model is 90.24% in 15 categories and 95.71% in 3 categories. Friedman test has been performed on the obtained results to observe the statistical significance. For both real 15 and 3 categories, the results of the statistical test are significant.

1. Introduction

Sentiment analysis (SA) which is also known as opinion mining is a process of determining a person's views on a particular topic (Kabir et al., 2023; Islam and Alam, 2023a). SA is the mining study of human opinion that analyzes people's opinions, feelings, evaluations and judgment towards social entities such as services, products, people, events, organizations etc (Kabir et al., 2023; Islam and Alam, 2023b). Sentiments can vary across cultures and languages (Nafisa et al., 2023). It classifies the polarity of a document as whether the opinion being communicated through blog, review, tweet, news, comment etc. is positive, negative, or neutral. Traditional approach that is generally implied in SA is to classify the human comment, review, blog, news as positive, negative or neutral class. Human opinions, feelings, evaluations and judgments towards social entities cannot be limited to only positive, negative and neutral categories. In a broader sense, positive class itself carries several different sentimental forms like happy, love, joy, enthusiasm, fun, care, excited, surprise, relief, bliss, satisfaction etc and so on. Similarly the negative category itself carries various emotional forms such as sad, angry, boring, disgust, fear, hate, worry, troll, sexual, bully etc. Some comments cannot be defined in any of the predefined categories that are referred to as neutral categories. The task of correctly identifying the polarity of a comment to some predefined categories such as positive, negative or neutral (traditional approach) classes, or in a broader sense to happy, sad, angry, love, fun, enthusiasm, boring, disgust, fear, hate, worry, troll, sexual, bully, neutral etc categories is basically termed as sentiment analysis (Bitto et al., 2023).

Why sentiment analysis is important? Let us search for the answer of this asking. People believe in human review on a topic more than traditional advertising (Prottasha et al., 2022). Nowadays anybody go for purchasing a product or service firstly seek the reviews of the previous buyers of that similar product or service. So, public opinion towards a product or service is an important issue for the buyers. Buyers always consider the thing in mind that what the common people are saying about that product or service. Nowadays almost every organization maintains their own website from where buyers can buy their necessary products or services from online. After the period of COVID-19, this is the most common form of product or service dealings (Bhowmik et al., 2022). Also the organizations need to know the opinion of the customers towards their product to stay up to the mark at the market place. To stay alive in the market place, organizations always analyze the customer reviews towards their products and the products from rival parties. By analyzing customer sentiments companies try to keep their reputation. So, opinion mining which is also known as sentiment analysis is a very important issue.

* Corresponding author.
E-mail addresses: shymum1702@cseku.ac.bd (Md.S. Islam), kazi@cseku.ac.bd (K.M. Alam).

https://doi.org/10.1016/j.nlp.2024.100069
Received 30 July 2023; Received in revised form 4 March 2024; Accepted 30 March 2024

Nowadays internet has become the most valuable part of our life. In this century, one cannot imagine a day without using the internet, browsing different social media accounts, posting different types of content on his profile. Everyone is maintaining a huge network on the internet with which he interacts daily (Sumit et al., 2018). So, millions of posts, blogs, comments, reviews, opinions are gathered on the internet everyday (Habibullah et al., 2023). People express their thoughts, feelings, opinions, evaluations on a particular topic generally in the form of text on the Internet in different languages and platforms. Numerous research studies have been done on sentiment analysis in English, Chinese, Hindi, Japanese, Arabic and Urdu languages while sentiment analysis in Bengali language is still in a dearth level (Nafisa et al., 2023; Bitto et al., 2023; Junaid et al., 2022). Few research works have been conducted in Bengali language on sentiment analysis due to lack of resources, datasets/corpus and complexity of Bengali language (Habibullah et al., 2023; Bitto et al., 2023; Amin et al., 2019). Bangla (Bengali), an ancient Indo-European language, the seventh most spoken language and is used daily by more than 250 million people in the world, it is the primary language of Bangladesh and secondary language of India (Habibullah et al., 2023; Bhowmik et al., 2022). Its use is becoming more prevalent with the recent growth of online microblogging sites (Azmin and Dhar, 2019). Bangladeshis are increasingly involved in online activities such as connecting with friends and family through social media, expressing their opinions and thoughts on popular micro-blogging and social networking sites, sharing opinions and thoughts through comments on online news portals, online market places and so on Hassan et al. (2022). This brings about a large amount of user-generated information on various sites, which can be used for many applications. Sites need to examine the millions of messages posted daily, extract all relevant posts for that product or service, analyze various types of user feedback, and finally outline user feedback to gain useful information. This task can be done manually by humans but it is very time consuming and tedious. This is why the concept of creating automated systems for sentiment analysis has become so important.

Sentiment analysis is also termed as opinion mining, opinion extraction, sentiment extraction, sentiment mining, subjectivity analysis, emotion analysis, review mining, polarity analysis, emotional AI etc (Hassan et al., 2022; Prottasha et al., 2022). Sentiment analysis has a lot of empirical and practical applications such as product analysis, social media monitoring, market analysis (Nafisa et al., 2023), product review analysis (Bitto et al., 2023), market trend analysis (Bhowmik et al., 2022), customer interest analysis, movie review analysis (Hassan et al., 2022), political review analysis (Tabassum and Khan, 2019) etc. Sentiment analysis is very important for business industries, NGOs, Governments and other organizations (Hassan et al., 2022). Sentiment analysis can be performed in three different levels. They are document level, sentence level and aspect level (Prottasha et al., 2022). The document level considers that a document has an opinion on an entity, and the task is to classify whether an entire document expresses a positive or negative sentiment (traditional SA). The task at the sentence level are with sentences and from a sentence it can be decided whether the sentence is positive, negative or neutral (traditional SA). The aspect level broadly known as aspect-based sentiment analysis performs a fine-grained analysis that recognizes aspects of a provided document or sentence and the sentiment expressed towards each aspect (Rahman and Dey, 2018). For example, cricket is an aspect where sentiment analysis can be performed, all the comments or reviews of people are related to that aspect (cricket). Similarly restaurant, election, football, world cup, fifa, movie, cinema, drama, viral person can be some examples of different aspects. In each different aspect SA can be applied.

Sentiment analysis is a well known application of natural language processing (NLP) (Bitto et al., 2023). SA is widely implemented using machine learning in different areas (Nafisa et al., 2023). Sentiment of texts can be impact fully analyzed with the help of machine learning methods (Shafin et al., 2020). If machine learning systems are trained with benchmark instances of different sentiments/emotions, machines can automatically learn how to detect sentiment without the help of human interaction (Shafin et al., 2020). Supervised machine learning (ML) algorithms such as Naive Bayes (NB), Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (K-NN), Support Vector Machine (SVM) etc. ensemble learning (EL) algorithms such as Random Forest (RF), Gradient Boost (GB), XGboost (XGB), LightGBM etc. deep learning (DL) algorithms such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (BiLSTM) etc are greatly applicable for sentiment analysis (Nafisa et al., 2023; Prottasha et al., 2022).

In this work, we have proposed a method for sentiment analysis on Bangla language based on a new comprehensive document level dataset and machine learning and deep learning approaches. The dataset contain a total of 203,463 Bangla comments collected from various microblogging sites. We have examined various hybrid feature metrics and various ML, EL and DL algorithms. The followings are the contributions of the proposed work:

1. A newly created comprehensive Bangla sentiment corpus of 203,463 comments from 5 microblogging sites (Facebook, YouTube, Instagram, TikTok, Likee), manually tagging them into 15 categories containing 204,6150 tokens and 165,319 unique tokens.
2. Validate the dataset by 40 native Bangla speakers with a validation accuracy of 94.67%.
3. Examining various feature extraction techniques such as BOW, N-gram, TF-IDF, TF-IDF-ICF, Word2Vec, FastText, GloVe, Bangla-BERT etc and groping them to form hybrid feature metrics and make a comparative study among them on the created Bangla sentiment dataset to extract important features and make a comparative study among the techniques applied.
4. The proposed novel hybrid feature extraction method (Bangla-BERT+Skipgram), skipBangla-BERT, outperforms all other techniques.
5. Applying ML, EL and DL algorithms that generates better performance in different metrics compared to state-of-the-art techniques. The hybrid CNN-BiLSTM model surpasses the existing state-of-the-art methods.

In this chapter, we briefly describe the basic concepts of sentiment analysis or opinion mining and its present perspective on Bangla natural language processing in Section 1, related works in Section 2, the proposed methodology for sentiment analysis on Bangla language in Section 3, experimental results analysis and discussion in Section 4 and Section 5 concludes the work with some future remarks.

2. Related works

In this modern technically advanced world, sentiment analysis is a topic of great importance in every language. Some of the recent studies of Bangla SA are discussed and summarized here.

An extensive dataset for sentiment classification from Bangladesh e-commerce reviews (Daraz and Pickaboo) was conducted by Rashid et al. (2024), this study proposed a dataset consisting of 78,130 reviews including 67,268 positive and 10,862 negative reviews from a wide range of products. Verse-based emotion analysis of Bangla music from lyrics was proposed by Mia et al. (2024), the authors developed a new dataset comprising of 6500 verses from 2152 Bangla songs and manually annotated them into 3 emotion classes namely Love, Sad and Idealistic, furthermore used several machine learning and neural network based classifiers. Among the models BERT outperformed others with an accuracy of 65%. A machine learning based method for the sentiment analysis of Bangla food reviews was proposed by Islam and Alam (2023b) creating a dataset of 44,491 reviews from different Facebook

pages and groups. The authors implemented several machine learning algorithms such as the MNB, SVM, KNN, LR and RF; and various deep learning algorithms namely CNN, LSTM, GRU, Bi-LSTM, Bi-GRU, CNN-LSTM, CNN-BiLSTM, CNN-GRU and CNN-BiGRU. Among these models, RF and CNN-BiGRU outperformed others from ML and DL domains respectively with an accuracy of 88.73% and 90.96%, furthermore, they also considered the Friedman statistical test and explainable NLP to explain the performance of the best fitted model. BanglaBook which is a new large-scale Bangla dataset collected from book reviews was (Kabir et al., 2023) proposed that consists of 158,065 samples. The authors used BOW and N-grams as feature extraction methods, afterwards implemented ML and DL classifiers and obtained highest 0.9331 f1-score with BERT. Another work from Nafisa et al. (2023) compiled a method for bipolar SA of online news comments, implemented six ML models along with BOW and TF-IDF transformers and a DL approach LSTM along with Word2Vec metric. They obtained highest 80% accuracy with RF and 83% with LSTM. Another new study was performed (Bitto et al., 2023) for the user reviews collected from food delivery startups. They collected 1400 reviews from 4 food delivery Facebook pages and applied bipolar SA. Applying ML and DL algorithms, they obtained highest accuracy of 89.64% using XGB and 91.07% from LSTM. An extended lexicon dictionary based method was (Bhowmik et al., 2022) proposed where the authors utilized DL algorithms and deployed in two aspect based datasets collected from Kabir et al. (2023). They obtained highest 84.18% accuracy using a hybrid model BERT-LSTM. At Hassan et al. (2022), the authors proposed a method for Bangla conversation reviews, they collected 1141 data from Bangla movies and short film scripts and implemented seven ML algorithms and recorded highest 85.59% accuracy with SVM. Another study carried by Prottasha et al. (2022) focused on transfer learning strategy of BERT based supervised fine tuning. They examined 6 different publicly available datasets and obtained highest results with the hybrid model CNN-BiLSTM along with BERT based fine tuning. They proved by experiments that BERT outperform Word2Vec, FastText, GloVe feature extractor techniques. Another DL based study was performed in Alvi et al. (2022) using LSTM, GRU and BLSTM classifiers along with 10-fold cross validation and achieved highest 78.41% accuracy score.

A method for Bangla text sentiment analysis using supervised machine learning with extended lexicon dictionary was proposed by Bhowmik et al. (2022). The authors collected datasets from Rahman and Dey (2018), there were two aspect based datasets, one is the Cricket dataset with a total of 2059 comments and the other is the Restaurant dataset with 2979 comments both covering three sentiment categories positive, negative and neutral. They created a weighted categorical lexicon data dictionary (LDD) for extracting sentiments from Bangla texts. There were a total of 1056 and 1115 active sentimental tokens as well as 970 and 2190 contradictory tokens in Cricket and Restaurant datasets respectively. They also developed the weighted list of dictionaries for Bangla conjunction, adjective and adverb quantifier. They developed a rule based BTSC algorithm of 30 steps that can classify the polarity of a Bangla document or sentence. The BTSC algorithm basically works on the basis of LDD and POS tagging and produces an SCS value (score of sentence) that is either less than 0 (belonging to the negative category), or greater than 0 (belonging to the positive category) or equal to 0 (belongs to neutral category).

For the product review sentiment analysis a method was proposed by Shafin et al. (2020) using NLP and machine learning in Bangla language. Online marketing become very popular after the period of COVID-19 and Bikroy, Daraz, Evaly, Chaldal.com are some popular e-commerce sites in Bangladesh (Shafin et al., 2020). The authors collected 1020 reviews from Bangla e-commerce sites. They preprocessed their data and used TF-IDF to extract important features from the dataset before go for ML models. Their dataset contained 52.2% positive reviews and 47.8% negative reviews. They implemented five supervised ML algorithms, they are — SVM, Random Forest, KNN, Logistic Regression and Decision Tree. They examined 30%, 40%, 50%, 60% and 70% data as test dataset. Using 30% data as test dataset they obtained accuracy 88.81% in SVM, 85.92% in Random Forest, 80.14% in KNN, 88.09% in Logistic Regression, 83.03% in Decision Tree. In the test dataset the real reviews were distributed as 62.5% positive reviews and 37.5% negative reviews while the model predicted 58.3% positive reviews and 41.7% negative reviews.

An automated system for sentiment analysis was proposed by Tuhin et al. (2019) from Bangla text using supervised learning techniques. The authors implemented sentence and document level of sentiment analysis. They created a sentiment dataset consisting of 7500 sentences which was tagged manually by them into six basic emotion categories namely happy, sad, excited, angry, tender and scared. From the six emotion categories, they mapped the happy, excited and tender categories to the higher emotion level positive category and rest of the three emotion categories (sad, angry and scared) to negative class. They split their built corpus into training dataset and test dataset using holdout method and took 7400 sentences for training and only 100 sentences for testing. They implemented two classification algorithms: Naive Bayes and Topical approach and compared their proposed work with two other papers on sentiment analysis from Bengali texts. Topical approach acquired highest accuracy above 90% but only for 100 test sentences.

Two new datasets for aspect-based sentiment analysis (ABSA) in Bangla language were introduced by Rahman and Dey (2018). Sentiment analysis can be performed in three different levels. They are document level, sentence level and aspect level (Rahman and Dey, 2018). They created two new datasets (name: Cricket dataset and Restaurant dataset) for ABSA. A total of 2900 comments from the Cricket domain covering 5 aspect categories make up the Cricket dataset and 2600 reviews from Restaurants make up the Restaurant dataset. The authors collected comments from different Facebook pages, BBC Bangla, Daily Prothom Alo etc. for Cricket dataset. The Cricket dataset was annotated by six native graduate students from second year, one faculty member, one MS student and two employees from Institute of Information Technology, University of Dhaka, into batting, bowling, team, team management and other aspects. They validated the cricket dataset based on zipf's law and measured an intraclass correlation value of 0.71 to validate the annotation method. To build the Restaurant dataset, they directly took assistance from the English standard Restaurant dataset. Abstract translations of all comments were done with their proper annotations into Bangla. A total of 2800 comments were contained in the main English dataset. The Restaurant dataset was based on five aspect categories food, price, service, ambiance and miscellaneous. Both the datasets were labeled into three target sentiment labels positive, negative and neutral. They applied TF-IDF to extract features from texts. They have implemented three supervised machine learning algorithms: SVM, Random Forest and KNN. On the Cricket dataset, they obtained the highest f1-score of 0.37 in the Random Forest classifier while the highest f1-score obtained for the Restaurant dataset was 0.42 from KNN.

A method for sentiment analysis on Bangla and Romanized Bangla text was proposed by Hassan et al. (2016) using deep recurrent models. The authors considered standard Bangla, Banglish (mixing of Bangla words with English words) and Romanized Bangla in this research work. They created a dataset with 9337 samples (BRBT dataset), of which 6698 samples are Bangla and 2639 samples are Romanized Bangla (RB dataset). They collected data from five different sources: 4621 samples from Facebook, 2610 samples from Twitter, 801 from YouTube, 1255 from online news portals, and 50 samples from product review pages and tagged them into three emotion categories — positive, negative and ambiguous. They utilized two native (Bangla) human experts to annotate all the test samples for a total of two validations one annotator knew nothing about the other's decisions. To show the validity of the performed tagging procedure, they showed a confusion matrix about all the annotated test samples and from this it was proved that the annotators agreed on 75% of the test samples. They mainly used Recurrent Neural Network (RNN), if we want to be more specific

Table 1
Summary of the recent works for the sentiment analysis from Bangla text

Name	Year	Dataset used	No. of classes	Dataset ownership	Dataset publicly available?	Feature metric	Best model (ML)	Best model (DL)
Mia et al. (2024)	2024	6500	3	Self	No	TF-IDF GloVe	SGD	BERT
Rashid et al. (2024)	2024	78,130	2	Self	Yes	N/A	N/A	N/A
Islam and Alam (2023b)	2023	44,491	3	Self	No	TF-IDF	RF	CNN-BiGRU
Kabir et al. (2023)	2023	1,58,065	3	Self	No	BOW N-Gram	RF	Bangla-BERT
Bitto et al. (2023)	2023	1400	2	Self	No	Word2Vec	XGB	LSTM
Hassan et al. (2022)	2022	1141	3	Self	No	N/A	N/A	Bangla-BERT
Junaid et al. (2022)	2022	1040	2	Self	No	BOW N-Gram TF-IDF Word2Vec GloVe	LR	LSTM
Prottasha et al. (2022)	2022	2900, 2600	3	Collected	Yes	Word2Vec FastText GloVe BERT	SVM	CNN-BiLSTM
Bhowmik et al. (2022)	2022	2900, 2600	3	Collected	Yes	Word2Vec	N/A	BERT-LSTM
Alvi et al. (2022)	2022	7000	3	Collected	Yes	Word2Vec	N/A	GRU

we will say they used LSTM based neural networks which contained three layers namely the embedding layer, LSTM layer and a fully connected layer containing three nodes. They obtained a maximum accuracy of 78% with 2 categories, and 70% with 3 categories on the Bangla dataset, 55% accuracy on the BRBT dataset with 2 categories, and 22% accuracy on 3 categories using categorical cross entropy loss.

A method was proposed by Chowdhury and Chowdhury (2014) for performing sentiment analysis in Bangla microblog posts. The authors collected 1300 Bangla tweets using the Twitter API and created their dataset. Instead of manual annotation they applied a semi-supervised bootstrapping method (constructing a lexicon dictionary of 737 single words) to annotate tweets into positive or negative sentiment categories. They split their dataset into training and test sets using the holdout method, leaving 1000 instances for training and 300 for testing. Two state-of-the-art supervised learning models, namely SVM and MaxEnt, were used in this study. Thirteen different features were used for both classifiers and f-scores were measured for both positive and negative categories. They achieved highest f-score 0.93 and accuracy 93% using SVM for both categories with (unigram+emoticon) feature.

2.1. Observations from literature review

We have several observations from the literature being studied. The summary of the recent works for the sentiment analysis from Bangla text is given in Table 1. The observed findings are summarized in the subsequent sections.

2.1.1. Small dataset size

An annotated high quality dataset is the pre-requisite of any NLP based classification task (Rahman, 2019). The datasets developed by Kabir et al. (2023) contain 158,065 samples, dataset of Bitto et al. (2023) contain only 1400 reviews from Facebook pages, dataset of Hassan et al. (2022) consists of 1141 data samples from Bangla movies and short film scripts, dataset of Hassan et al. (2016) have only 9337 samples (BRBT dataset), dataset of Shafin et al. (2020) contain 1020 reviews from Bangla e-commerce sites, dataset of Tuhin et al. (2019) containing 7500 sentences and dataset of Chowdhury and Chowdhury (2014) contains 1300 Bangla tweets. Two aspect based datasets on Cricket (2900 samples) and Restaurant (2600 samples) were proposed by Rahman and Dey (2018), and later the authors of Bhowmik et al. (2022), Prottasha et al. (2022) used those datasets. From the above study, it is clear that no benchmark dataset is still available for sentiment analysis on Bangla language. So, there is a scope to build one or more benchmark datasets for Bangla sentiment analysis.

2.1.2. Fewer number of categories (polarity labels)

Traditional approach that is generally implied in SA is to classify the human review as positive, negative or neutral class. The authors of Kabir et al. (2023), Bitto et al. (2023), Hassan et al. (2022, 2016), Shafin et al. (2020), Chowdhury and Chowdhury (2014), Rahman and Dey (2018), Bhowmik et al. (2022), Prottasha et al. (2022) used this traditional approach. Another work from Tuhin et al. (2019) used six basic emotion categories namely happy, sad, excited, angry, tender and scared. So, there is a scope to consider more polarity labels in Bangla SA instead of using only the traditional classes which can reflect more actual feelings of human being.

2.1.3. The use of traditional feature extraction methods

For text mining related works, TF-IDF, N-gram and BOW is the most common traditionally used feature extraction metrics for ML models. The authors of Kabir et al. (2023), Nafisa et al. (2023), Shafin et al. (2020), Tuhin et al. (2019) used these metrics. A new feature extractor named lexicon data dictionary (LDD) is used by Bhowmik et al. (2022). The authors of Nafisa et al. (2023), Prottasha et al. (2022) used word embedding model of Word2Vec and BERT respectively. But the authors (Chowdhury and Chowdhury, 2014) used 13 different hybrid feature extractors. Therefore, it is noticed that most of the works mainly used the traditional feature extraction methods. So, there is a scope to use different hybrid feature extraction methods for the sentiment analysis on Bangla language.

2.1.4. Domination of deep learning models over fundamental machine learning models

Most of the recent methods on Bangla sentiment analysis focuses on both the ML and DL models, but the obtained results of DL models surpass the traditional ML models. The methods proposed by Kabir et al. (2023), Bitto et al. (2023), Bhowmik et al. (2022), Prottasha et al. (2022) achieved better results with DL models and so does for others also. So, there is a clear domination of deep learning models over the fundamental machine learning models.

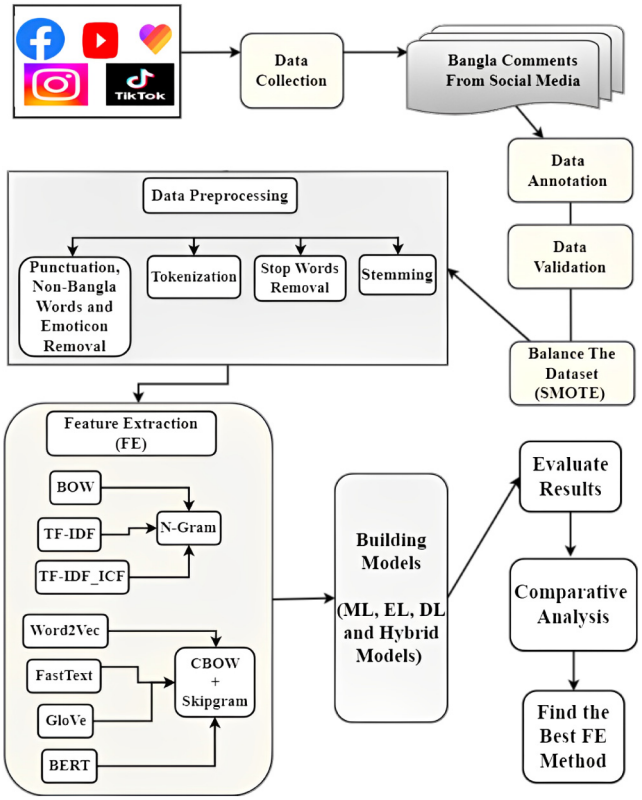

Fig. 1. Workflow of the proposed Bangla SA system.

3. Proposed methodology for sentiment analysis on bangla language

Sentiment analysis (SA) is the mining study of human opinion that analyzes people's opinions, feelings, evaluations and judgment towards social entities such as services, products, people, events, organizations etc (Nafisa et al., 2023). An annotated high quality dataset is the pre-requisite of any NLP based classification task (Rahman, 2019). In this work (SA_Bangla), we have proposed a method for Bangla SA using a new novel comprehensive dataset and applying various hybrid feature extraction techniques. Our work starts with data collection, then gradually we adopt several more steps such as data preprocessing, data visualization, split the dataset into training and test sets, feature extraction, building models and evaluate results etc. Fig. 1 illustrates the workflow of the proposed method.

3.1. Data collection

Currently micro-blogging sites are being used by a large number of Bangla speakers (Chowdhury and Chowdhury, 2014), millions of people are commenting and texting in Bangla language using various social media platforms such as Facebook, YouTube, Instagram, TikTok, Likee and so on. We have collected Bangla comments by using our self-developed crawlers from 5 micro-blogging sites, a total of 203,463 Bangla comments were collected and saved in an excel file containing 5 columns: *comment*, *basic sentiment category*, *higher sentiment category*, *reaction number* and *data source*. Six people were involved in data collection process (4 males and 2 females) and 5 were involved in data annotation process (3 males and 2 females) and their details information are given in Table 2. The overall summary of the domain based data collection is presented in Table 3. We have collected more data from Facebook and YouTube because Bangla comments are more available there. Nowadays Bangladeshis are very interested in using new social sites like TikTok and Likee (small video community) so we have also considered these two new sites. Bangla comments are rare on Instagram, so we could not collect more comments from Instagram. We have collected data using our self-developed crawler and named it as Sentiment Analysis Dataset Crawler (SAD_crawler). The pseudocode of the developed crawler is shown in Algorithm 1 where *X, Y, P* and *Q* are all dynamic classes; getElementsByClassName(), replace(), trim(), include() are several used methods and *copied_content* is the output variable to store Bangla comments.

Algorithm 1 SAD_Crawler

```
Take a content d to collect Bangla comments
Take an empty variable alldata to store comments
alldata ← NULL
englishWordPattern (EWP) ← [a-zA-Z]
comments ← d.getElementsByClassName("X")[0]
c ← comments
length ← c.getElementsByClassName("Y").length
for i in length do
    rc ← getElementsByClassName("P")[0]
    rc ← rc.innerText.replace(EWP, " ")
    co ← c.getElementsByClassName("Y")[i].rc
    if co.trim() == NULL then
        continue
    end if
    alldata ← alldata + comment + "\t"
    try
        r ← d.getElementsByClassName("Y")
        s ← r[i]
        t ← s.getElementsByClassName("Q")
        likes ← t[0].innerText
        if likes.include(}likese) then
            alldata ← alldata + likes + "\n"
        else
            alldata ← alldata + " " + "\n"
        end if
    catch Expected exception
        alldata ← alldata + " " + "\n"
    end try
end for
copied_content ← copy(alldata)
Output: In copied_content variable, Bangla comment and its visible reaction number (likes) are copied.
Paste the copied content to an excel file to save the data
```

3.2. Data annotation and validation

We have annotated the collected 203,463 Bangla comments manually by five human experts (4 graduate students and 1 MS student) into 15 basic sentiment categories such as happy, sad, angry, enthusiasm, fun, love, sexual, boring, disgust, surprise, fear, worry, hate, relief and neutral which takes around six month. Among the 15 categories, the base emotion classes are subsequently mapped to 3 higher sentiment categories such as happy, enthusiasm, fun, love, surprise and relief belong to positive sentiment category and rest of the others belong to negative sentiment category and the unidentified or mixed comments belong to neutral category. To test the validity of the annotation process, we have conducted an analysis with 40 native Bangla speakers who are graduate students of North Western University, Bangladesh, each student was given 100 different comments and asked to annotate them into 15 predefined sentiment categories. So after running an audit on a total of 4000 comments (with 800 samples each from Facebook, YouTube, Instagram, TikTok and Likee) we found that it provides 94.67% accuracy. The confusion matrix of the annotation process is shown in Fig. 2 and the complete description of the dataset is given in Table 4, where we have shown category based measurements including no. of total comments, tokens, unique tokens, top 2 topic keywords etc.

Table 2
Information of data collectors and annotators.

ID of participant	Gender	Profession	Role	Collection amount	Annotation amount
p1	Male	MS student/author	Data collection and annotation	83,652	94,197
p2	Male	Faculty/author	Data collection	10,545	N/A
p3	Male	Graduate student	Data collection and annotation	35,760	35,760
p4	Male	Graduate student	Data collection and annotation	40,647	40,647
p5	Female	Graduate student	Data collection and annotation	17,350	17,350
p6	Female	Graduate student	Data collection and annotation	15,509	15,509

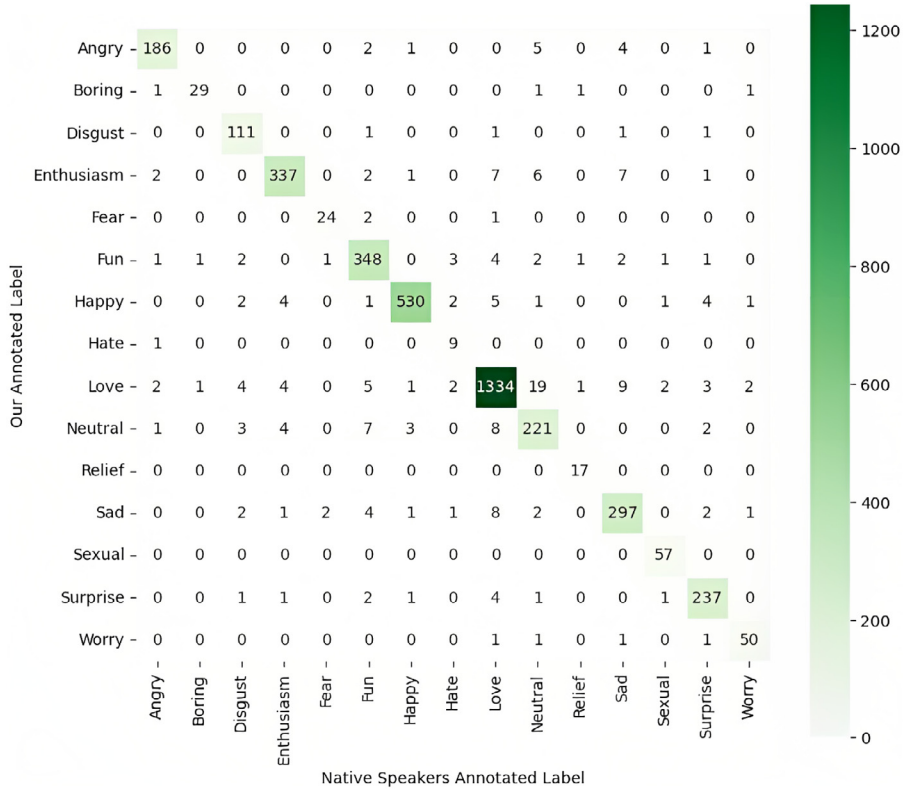

Fig. 2. Confusion matrix of annotation process.

Table 3
Summary of domain based data collection.

Data source	No. of comments	Collection period
Facebook	71,429	2022–2023
YouTube	42,884	2022–2023
Instagram	12,764	2022–2023
TikTok	52,367	2022–2023
Likee	24,019	2022–2023

3.3. Problem definition

Now all we have an annotated sentiment dataset, we need to know what our problem definition actually is or what we actually want to do with the dataset. The task of correctly identifying the polarity of a document to some predefined categories such as positive, negative or neutral (traditional approach) classes, or in a broader sense to happy, sad, angry, love, fun, enthusiasm, boring, disgust, fear, hate, worry, troll, sexual, bully, neutral etc categories is basically termed as sentiment detection (Bitto et al., 2023). The problem is to classify sentiments correctly from a labeled dataset. consider a document in the dataset contains many sentences with a total word count of N, which is denoted by D_{sent} where P_{comt} is a vocabulary of K words. S_{categ} of the D_{sent} in P_{comt} represents the category of different sentiments, where r is the total sentiment category labels. E_x is the required output sentiment label for the test instance x.

$$D_{sent} = (B_1, B_2, B_3, B_4, B_5, \ldots, B_N) \tag{1}$$

$$P_{comt} = (V_1, V_2, V_3, V_4, V_5, \ldots, B_K) \tag{2}$$

$$S_{categ} = (l_1, l_2, l_3, l_4, l_5, \ldots, l_r) \tag{3}$$

The output is represented as:

$$E_x = F_{max}(V_r, P_r) D_{sent} \tag{4}$$

Table 4
Overview of category-wise data collection.

Higher Category	Basic Category	No. of Total Comments	No. of Total Tokens	No. of Unique Tokens	Topic Keywords (Top 2)	No. of Higher Category Comments
Positive	Love	56,631	534,125	53,177	"সুন্দর"(nice), "ভালোবাসি"(love)	149,809
	Enthusiasm	37,965	523,566	64,533	"আকুল"(eager), প্লিজ(please)	
	Happy	26,596	200,403	24,929	"ভালো(good), "মাশাআল্লাহ"(masha Allah)	
	Fun	24,351	241,592	30,447	"হাসতে"(laugh), "মজা"(recreation)	
	Surprise	3501	16,715	4373	"চমৎকার"(excellent), "অবাক"(surprised)	
	Relief	765	3109	168	"জিতসো"(win), "বাচলাম"(survived)	
Negative	Angry	27,054	338,737	47,497	"নাস্তিক"(atheist), "শয়তান"(devil)	49,726
	Sad	6101	67,749	15,029	"আহারে"(scream), "কষ্ট"(trouble)	
	Sexual	5854	37,011	7173	"হট"(hot), "মাল"(sexy)	
	Disgust	3888	26,444	7177	"ফালতু"(nonsense), "আবাল"(stupid)	
	Boring	2494	14,925	2152	"ভাল্লাগেনা"(disliked), "চোরের"(thief's)	
	Worry	1810	10,693	541	"অসম্ভব"(impossible), "হবেনা"(never)	
	Fear	1442	7335	760	"মাফ"(forgive), "পারবা"(capable)	
	Hate	1083	4915	195	"ছি"(shit), "লজ্জা"(shame)	
Neutral	Neutral	3928	18,831	5266	"কেমন"(how), "আচ্ছা"(right)	3928
Total		**203,463**	**204,6150**	**165,319**		**203,463**

3.4. Balance dataset

When a dataset's distribution of examples among its various classes is noticeably unbalanced, the term "data imbalance" is used (Chawla et al., 2002). In many situations in real life, the issue of imbalanced data sets might occur. A learning model's ability to reliably predict actual sentiment category may be hampered by this class imbalance. When one class has much more examples than the other, the data is imbalanced, which results in models that are biased and predictions that are incorrect for the minority class. Learning models can produce more precise predictions for all classes, particularly for the minority class, by balancing the data. This enhances decision-making and guards against biased results. To improve forecasts, address real-world events, and improve decision-making by eliminating prejudice towards the dominant class, imbalanced data must be balanced (Chawla et al., 2002).

3.4.1. SMOTE

By creating synthetic samples for the minority class, SMOTE (Synthetic Minority Over-sampling Technique) is an oversampling technique used to correct class imbalance. It seeks to boost the dataset's representation of the minority class and enhance the effectiveness of machine learning models (Chawla et al., 2002). There are many variants of SMOTE such as the fundamental SMOTE, SMOTENC, SMOTEN, ADASYN, BorderlineSMOTE, KMeansSMOTE, SVM-SMOTE etc (Chawla et al., 2002). But in this work, we did not implement any specialized variant of SMOTE rather we used the fundamental SMOTE only. The description of the fundamental SMOTE is provided below:

1. Consider a member of a minority class, T_i.
2. From the minority class examples, determine T_i's P closest neighbors. A parameter set by the user determines the value of P.
3. Randomly select one of the k nearest neighbors, T_j.
4. Create a synthetic example, Enew, by using the following equation to interpolate between T_i and T_j: $T_{new} = T_i + (T_j - T_i)\ b$

Here, b is a randomly chosen constant in the range [0, 1]. The equation calculates the difference between T_i and T_j and scales it by b. Adding this scaled difference to T_i creates a new example, T_{new}, which lies on the line segment between T_i and T_j. The minority class space is effectively expanded by SMOTE's generation of a collection of synthetic instances by repeating this process for various minority class examples. To generate a balanced representation of the classes, these artificial instances are subsequently included in the initial dataset.

3.5. Data preprocessing

Comments of the microblogging sites contain both Bangla and English punctuation, hashtag (eg.#), emoticon, slang etc and so on Chowdhury and Chowdhury (2014). So the raw comments always contain irrelevant characteristics and noise, it is very important to eliminate them from the dataset (Akther et al., 2022). Noisy raw data cannot correctly categorize the actual sentiment this is why preprocessing of Bangla comments is so important. In this work, we have performed tokenization, removal of punctuation, emoticon, non-Bangla words, stop words removal and stemming as preprocessing steps.

3.5.1. Tokenization

During the tokenization process comments are divided into sentences and the sentences are divided into words. The Bangla comment "আপনার কাজগুলো আমার খুব ভালোলাগে", [i like your works very much] after tokenizing become "আপনার"(your), "কাজগুলো"(works), "আমার"(i), "খুব"(very), "ভালোলাগে"(like).

3.5.2. Punctuation, non-Bangla words and emoticon removal

The commonly used punctuation marks in Bangla are "।", "?", "!", "-", "," etc and so on. Punctuation marks and special characters and symbols especially "#"(hashtags), ©, & and braces have been excluded from the dataset. Non-Bangla words especially English words and unnecessary emoticons are also removed from one version of the dataset (Shafin et al., 2020). The comment "what??? সে শেখ হাসিনাকেও অপমান করলো", (what??? he also insulted Sheikh Hasina) after the removal of punctuation, non-Bangla words and emoticon become "সে শেখ হাসিনাকেও অপমান করলো", (he also insulted Sheikh Hasina).

3.5.3. Stop words removal

The function words that are used repeatedly in a language but do not have any domain based meaning *i.e.* those words are repeated in any domain are mainly referred to as stop words (Rahman and Dey, 2018). There are many noteworthy stop words available in Bangla such as "বরং"(rather), "কিন্তু"(but), "নতুবা"(or), "যদি"(if), "তুমি"(you), "অতএব" (therefore), "এখন"(now) etc. We have excluded all stop words from our developed dataset using the Bangla stop words[1] list. The Bangla comment "বাংলাদেশের মানুষ এত দারুন দারুন নাটক থাকতে ভারতের সিরিয়াল কেনো দেখে বুঝিনা", (don't understand why the people of Bangladesh watch Indian serials when there are so many great dramas) become "বাংলাদেশের মানুষ দারুন দারুন নাটক ভারতের সিরিয়াল বুঝিনা", (don't understand the people of Bangladesh Indian serials many great dramas) after removing the stop words. The stop words "এত" (so), "থাকতে"(when there are), "কেনো" (why), "দেখে"(watch) have been removed from the comment.

[1] https://www.ranks.nl/stopwords/bengali

Table 5
Top 10 frequent n-grams without removing stop words of the dataset.

Rank	Unigram	Frequency	Bigram	Frequency	Trigram	Frequency
1	না (no)	31,973	অনেক সুন্দর (very nice)	4083	প্লিজ প্লিজ প্লিজ (plz plz plz)	1433
2	কি (what)	18,341	ভালো লাগে (feel good)	2794	অনেক ভালো লাগে (feel very good)	602
3	করে (do)	17,559	খুব সুন্দর (really nice)	2162	খুব ভালো লাগে (really feel good)	472
4	অনেক (many)	15,637	অনেক ভালো (very good)	1859	হিরো আলম ভাই (Hero Alam bro)	464
5	এই (this)	15,527	খুব ভালো (really good)	1841	হা হা হা (ha ha ha)	453
6	আমার (my)	14,505	হিরো আলম (Hero Alam)	1670	বিশ্বাস করে না (do not believe)	416
7	আর (and)	14,110	মনে হয় (may be)	1618	খুব ভালো লাগলো (feel very good)	384
8	আমি (i)	13,958	ছি ছি (bo! bo!)	1382	দেখা শুরু করলাম (i started watching)	366
9	ভালো (good)	13,461	ভালো লাগলো (feel good)	1180	অনেক সুন্দর লাগছে (looks very nice)	345
10	সুন্দর (nice)	13,225	অনেক অনেক (many many)	1017	কথায় কান দিয়েন (listen to the words)	326

3.5.4. Stemming

Stemming is the process of reducing a word to its base or root form Chowdhury and Chowdhury (2014). Stemming algorithms aim to remove suffixes from words so that they can be matched with other words that have the same root. For example, the words "চুক্তি", "চুক্তিটিকে", "চুক্তিটিতে", "চুক্তিটির", "চুক্তিটা", "চুক্তিজনিত", "চুক্তিগুলো", "চুক্তিভিত্তিক" all have the same root word "চুক্তি". By stemming these words, we can match them with other words that have the same root word, such as "চুক্তিগত" or "চুক্তিকৃত". The Bangla comment "গতকাল মা ছাড়া শিশুটির ছবি ফেসবুকে দেখে ভীষন কষ্টপেয়েছিলাম", (yesterday i was saddened to see the picture of the motherless child on facebook) become "গতকাল মা ছাড়া শিশু ছবি ফেসবুক দেখ ভীষন কষ্ট পেয়ে" after stemming is performed. We have implemented stemming on our dataset as a final pre-processing step.

3.6. Statistical dataset visualization and analysis

Statistical analysis is a very important aspect almost in every subject to get usual observations from experiments. In this section, we have analyzed various statistical language phenomena on the dataset such as the usage of n-grams, average word length, character level analysis, frequency of characters, Zipf's law and type token information respectively. From these analysis, we can have a good understanding about the dataset and its linguistics usefulness.

3.6.1. Unigram

An n-gram, also known as a unigram in the domains of probability and computational linguistics, is made up of just one element from a particular sample of text or speech (Akther et al., 2022). At this level, non-stemmed unigrams for the entire dataset have been extracted from the preprocessed data. Table 5 represents a list of the top 10 unigrams, bigrams and trigrams in the dataset. Any language's function words or stop words are always its most common words. They are crucial for determining a dataset's quality. We found that 32.84% of the total tokens in our dataset are function words. We have taken out the stop words from one version of the dataset in order to get the content words. Table 6 lists the top 10 frequently occurring unigrams, bigrams and trigrams after the stop words have been eliminated. Additionally, we have recorded the total number of unique unigrams present in the dataset, providing insights into the diversity and vocabulary richness of the dataset. Without removing the stop words "না"(no) is the first ranked word in the dataset having an occurrence frequency of 31,973 and after removing the stop words "ভালো"(good) is the first ranked word in the dataset having an occurrence frequency of 13,461.

3.6.2. Bigram

A bigram is a pair of next-to-one words from a specific passage of text. For $n = 2$, a bigram is an n-gram. In many applications such as computational linguistics and NLP based works, the frequency distribution of every bigram in a text is frequently employed for a quick statistical analysis of text (Akther et al., 2022). At this level, we have extracted bigrams from non-stemmed words. The number of bigrams utilizing four *threshold frequencies* are displayed in Table 7. The total number of times two consecutive words appeared together in a dataset is referred to as the frequency or count of that bigram. We have utilized 4 *threshold frequencies*: 20, 50, 100, and 200 to observe the behavior of bigrams in the dataset. The term *threshold frequency* describes the upper boundary of whether to accept or reject a bigram from all bigrams. A bigram frequency (count) must be at least 20 to be accepted; otherwise, it will be rejected. This is known as the *threshold frequency* of 20. The proposed dataset contains 8463 unique bigrams out of a total of 944,682 bigrams using *threshold frequency* 20. When we have increased the *threshold frequency* to 200, then the total no. of bigrams and no. unique bigrams decreased to 131,758 and 321 respectively. The most frequent bigram of the proposed dataset is "অনেক সুন্দর"(very nice) which appears 4083 times together in the dataset.

3.6.3. Trigram

A trigram is a grouping of three adjacent words or phrases from a text or speech sample. For $n = 3$, a trigram is an n-gram (Akther et al., 2022). Like bigrams, the frequency distribution of trigrams can be useful for a straightforward statistical analysis of text. The same *threshold frequencies* have also been utilized to extract trigrams from the dataset. Table 8 displays the effect of *threshold frequencies* on the trigrams. The proposed dataset contains 1875 unique trigrams out of a total of 85,208 trigrams using *threshold frequency* 20. Interestingly, when the *threshold frequency* is increased to 200, the total number of

Table 6
Top 10 frequent n-grams after removing stop words of the dataset.

Rank	Unigram	Frequency	Bigram	Frequency	Trigram	Frequency
1	ভালো (good)	13,461	হিরো আলম (Hero Alam)	1671	হিরো আলম ভাই (Hero Alam bro)	464
2	সুন্দর (nice)	13,225	সুন্দর লাগছে (looks nice)	885	হা হা হা (ha ha ha)	453
3	নাটক (drama)	12,429	সুন্দর নাটক (beautiful drama)	748	বাজে কথায় কান (listen to bad words)	334
4	আপু (sister)	10,096	অসাধারণ নাটক (great drama)	744	ভবিষ্যতে একটা ব্র্যান্ডের (a brand in the future)	318
5	কথা (words)	8075	পরকালে বিশ্বাস (belief in the afterlife)	665	অসাধারণ একটা নাটক (a wonderful drama)	303
6	অসাধারণ (wonderful)	7426	পম পম (pom pom)	654	এক কথায় অসাধারণ (amazing in a word)	215
7	বিশ্বাস (belief)	5712	হাসতে হাসতে (laughingly)	579	হাসতে হাসতে শেষ (laughingly done)	208
8	নাস্তিক (atheist)	5487	মোশাররফ করিম (Musharraf Karim)	571	হেদায়েত দান করুক (may he give guidance)	206
9	ভিডিও (video)	5236	দান করুক (provide)	535	একটা রিপ্লাই ভিডিও (a reply video)	168
10	মানুষ (people)	4885	লাইক কমেন্ট (like comment)	503	উন্নয়ন উন্নয়ন উন্নয়ন (development development development)	158

Table 7
Threshold frequency wise Bigram.

Threshold frequency	No. of all bigrams	No. of all unique bigrams
20	494,682	8463
50	317,844	2442
100	214,827	926
200	131,758	321

Table 8
Threshold frequency wise Trigram.

Threshold frequency	No. of all trigrams	No. of all unique trigrams
20	85,208	1875
50	39,936	358
100	23,355	106
200	13,425	36

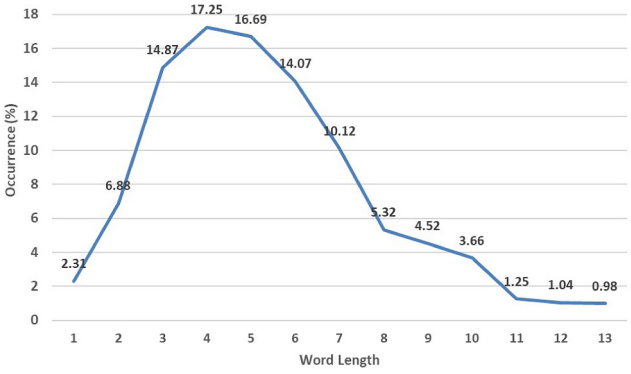

Fig. 3. Usage of words Vs. word length of the dataset.

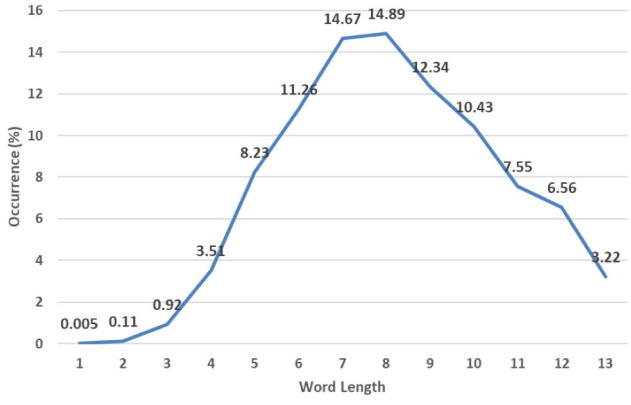

Fig. 4. Usage of unique words Vs. word length of the dataset.

trigrams is 13,425 and no. of unique trigrams is 36 only. The top 2 frequent trigrams of the proposed dataset are "প্লিজ প্লিজ প্লিজ"(plz plz plz) and "অনেক ভালো লাগে"(feel very good) which appear 1433 and 602 times together in the dataset.

3.6.4. Average word length

The dataset contains a total of 116,60,683 characters (excluding punctuation and spaces), with an average of 5.69 letters per word. An average of 4.5 letters are used in each word in everyday English text (Dash, 2005). Bangla words are longer than English words because they begin with 11 vowels, 39 consonants and 20 allographs whereas English words starts with only 5 vowels and 21 consonants (Akther et al., 2022). Fig. 3 plots the percentage of words used in our dataset versus word length. The top 2 occurrence frequencies are recorded for 4 and 5 letters per word of 17.25% and 16.69% respectively. We found that 32.84% of the total tokens in our dataset are function words. Fig. 4 is affected due to the frequent use of function words. The average letters per word for the unique words is increased to 7.84 (see Fig. 4) where we have recorded the top 2 occurrence frequencies for 7 and 8 letters per word of 14.67% and 14.89% respectively. Table 9 describes the relationship between the word length and frequency of n-letter (n = 1 to 7) words where we have shown the top 10 n-letter words in our dataset.

Table 9
Top 10 frequent N-letter (n = 1 to 7) words of the dataset.

1-letter	%	2-letter	%	3-letter	%	4-letter	%	5-letter	%	6-letter	%	7-letter	%
ও	0.078	না (no)	0.247	করে (do)	0.145	অনেক (many)	0.134	আপনার (you)	0.087	সুন্দর (nice)	0.113	অসাধারণ (won)	0.063
এ	0.033	কি (what)	0.152	আমি (i)	0.116	আমার (my)	0.123	তোমার (your)	0.069	আল্লাহ (Allah)	0.054	বিশ্বাস (belief)	0.049
ই	0.009	এই (this)	0.131	আপু (sister)	0.085	ভালো (good)	0.115	প্লিজ (plz)	0.047	কিন্তু (but)	0.041	নাস্তিক (atheist)	0.041
ত	0.007	আর (and)	0.117	ভাই (brother)	0.076	নাটক (drama)	0.106	ভিডিও (video)	0.045	আমাদের (our)	0.039	সাপোর্ট (support)	0.033
ঐ	0.006	তো (so)	0.088	কথা (words)	0.067	একটা (one)	0.094	মানুষ (people)	0.041	আপনাকে (you)	0.036	ধন্যবাদ (thanks)	0.031
আ	0.003	কে (who)	0.081	খুব (very)	0.063	জন্য (for)	0.074	দেখতে (to see)	0.031	কমেন্ট (comment)	0.032	আল্লাহর (Allah's)	0.019
ম	0.003	যে (that)	0.071	হবে (will be)	0.057	থেকে (from)	0.073	তাহলে (then)	0.028	তোমাকে (you)	0.029	মানুষের (people's)	0.017
র	0.003	টা (the)	0.060	তার (his)	0.053	আপনি (you)	0.069	লাগছে (feeling)	0.027	নাটকের (drama's)	0.016	রিপ্লাই (reply)	0.015
খ	0.002	এর (of)	0.058	আছে (have)	0.051	তুমি (you)	0.066	হয়েছে (done)	0.023	দেখলাম (seen)	0.015	আপনাদের (yours)	0.011
হ	0.002	হয় (is)	0.054	মনে (in mind)	0.049	কিছু (some)	0.054	অভিনয় (acting)	0.022	কিভাবে (how)	0.011	সুন্দরী (belle)	0.008

Table 10
Percentage of occurrence of each letter of the dataset.

letter	%	letter	%	letter	%	letter	%	letter	%
অ	0.55	ঘ	0.06	থ	0.48	ষ	0.37	ৌ	0.03
আ	1.61	ঙ	0.04	দ	1.87	স	2.16	্	3.05
ই	1.44	চ	0.71	ধ	0.39	হ	1.24	ৎ	0.03
ঈ	0.01	ছ	1.00	ন	4.72	ং	0.31	ী	0.00
উ	0.26	জ	0.99	প	1.76	া	10.05	ড়	0.17
ঊ	0.00	ঝ	0.07	ফ	0.32	ি	4.32	ঢ়	0.00
এ	0.94	ঞ	0.03	ব	2.90	ী	0.38	য়	1.20
ঐ	0.01	ট	1.31	ভ	0.80	ু	1.95		
ও	0.54	ঠ	0.10	ম	2.73	ূ	0.06		
ঔ	0.00	ড	0.30	য	1.27	ৃ	0.06		
ক	4.25	ঢ	0.02	র	5.59	ে	6.90		
খ	0.83	ণ	0.18	ল	2.90	ৈ	0.02		
গ	0.84	ত	2.64	শ	0.77	ো	1.66		

3.6.5. Character level analysis

Instead of considering words as the basic unit of analysis, character-level analysis focuses on the individual characters that make up the text. By examining individual characters and their context, different errors or inconsistencies can be identified, and appropriate corrections can be suggested (Akther et al., 2022).

Based on our dataset, we have determined the percentage of times each Bangla character occurs. The top 5 frequently used characters, according to our dataset, are "া", "ে", "র", "ন", "ক" and the least frequently used characters are "ঔ", "ঙ", "ৌ", "ঢ়", "ঐ" respectively. Table Table 10 demonstrates the percentage of times each Bangla character is used in our dataset. A statistical study of the dataset at the character level reveals that 5.36% of the characters are vowels, 45.01% are consonants and 48.79% are allographs. The top 2 most frequently used characters are both allographs, they are "া"(Aa-kar) and "ে"(e-kar) covering 10.05% and 6.90% of characters of the dataset. Another character level statistical study on the vocabulary (unique words of the dataset) have been conducted to observe how it behaves. The percentage of occurrence of initial letter of unique words of the dataset are listed in Table 11. We have noticed that the likelihood of the letters "ক", "ব", "আ", "ন", "স" occurring as the first alphabet is higher than it is for the other letters. The words that starts with the alphabet "ক" covers 11.47% words of the dataset.

3.6.6. Zipf's law

It is impossible to expect human inspection to guarantee the quality of a dataset with millions of words. So, Zipf's distribution of the dataset is examined to check whether it is reflecting the vocabulary usage according to human nature or not. According to Zipf's law, there is a correlation between a word's frequency (d) and its rank (p) in the list (if all the words in a big dataset are listed in order of their frequency of occurrence) (Manning and Schutze, 1999). According to Zipf's law:

$$d \propto \frac{1}{p}$$

For example, this means that the 50th most frequent word should appear with twice the frequency of the 100th most frequent word and so on. Fig. 5 depicts the Zipf's curve of the dataset where x and y axis represents logarithmic rank and frequency of words respectively. The curve is roughly linear (see Fig. 5), so it is proved that our dataset is holding the Zipf's law approximately.

3.6.7. Hapax legomena and vocabulary growth

Table 12 shows the type token information of the dataset. The total number of word types are 165,319 and the number of word types occurring once are 108,599. About half of the words in the dataset are referred to as hapax legomena, which appears only once (Akther et al., 2022). In our dataset, we have more than half words belonging to hapax legomena. The vocabulary growth rate is measured as:

$$G = \frac{\mathrm{W}(1)}{\mathrm{N}} \tag{5}$$

Here the parameters G, $W(1)$ and N represents the vocabulary growth rate, the number of word types occurring once and the total number of words in the dataset respectively. For our dataset, the vocabulary growth rate is (108,599/204,6150) = 0.053 which is reasonable.

Table 11
Percentage of occurrence of initial letter of unique words of the dataset.

letter	%	letter	%	letter	%	letter	%	letter	%
অ	3.04	ঘ	0.26	থ	0.99	ষ	0.01	ৌ	0.00
আ	8.57	ঙ	0.00	দ	5.31	স	6.15	্	0.00
ই	0.72	চ	1.96	ধ	0.59	হ	4.36	ৎ	0.00
ঈ	0.07	ছ	1.05	ন	6.26	ং	0.00	ী	0.00
উ	0.76	জ	2.51	প	5.40	া	0.00	ড়	0.00
ঊ	0.00	ঝ	0.06	ফ	0.86	ি	0.00	ঢ়	0.00
এ	5.04	ঞ	0.00	ব	6.92	ী	0.00	য়	0.00
ঐ	0.06	ট	0.97	ভ	3.21	ু	0.00		
ও	1.00	ঠ	0.27	ম	5.03	ূ	0.00		
ঔ	0.01	ড	0.32	য	2.30	ৃ	0.00		
ক	11.47	ঢ	0.08	র	1.38	ে	0.00		
খ	1.50	ণ	0.00	ল	1.81	ৈ	0.00		
গ	1.66	ত	4.82	শ	1.43	ো	0.00		

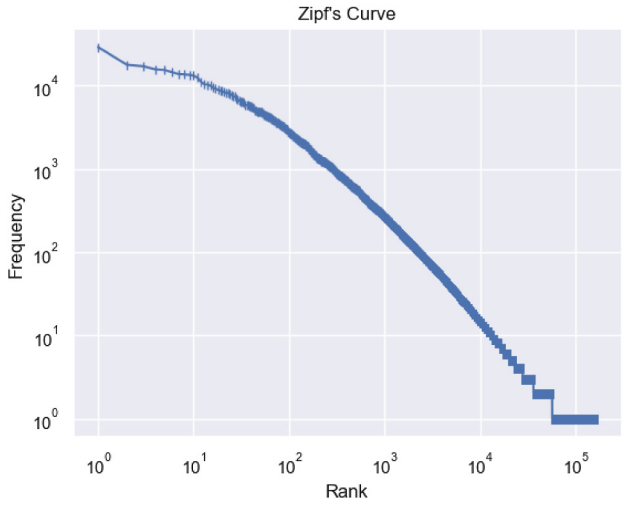

Fig. 5. Zipf's curve of the dataset.

Table 12
Type token information of the dataset.

Words frequency count	No. of words
Total number of word types:	165,319
Word-types occurring once:	108,599
Word-types occurring Twice:	41,186
Word-types occurring 3–50 Times:	306,382
Word-types occurring 51–100 Times:	123,121
Word-types occurring 100–1000 Times:	546,685
Word-types occurring 1000–5000 Times:	508,160
Word-types occurring 5000–10000 Times:	205,424
Word-types occurring More than 10000 Times:	206,593

3.7. Feature extraction

Extracting relevant features from textual data is an important aspect in NLP (Rahman and Dey, 2018). Models based on ML cannot directly deal with data in textual form, so there should be an intermediate process that will bridge the connection between the raw Bangla texts data and the ML model by transforming the raw textual data into some kind of strategic numerical form *i.e.* by extracting relevant features from texts broadly known as feature extraction (Islam and Alam, 2023a; Sharmin and Chakma, 2021). These features can be used further to train ML models in order to perform various classification tasks such as document classification, sentiment analysis and many other classification tasks (Tabassum and Khan, 2019). There are several well-known metrics for extracting features from texts such as BOW, TF-IDF, Word Embeddings and so on Prottasha et al. (2022).

Table 13
Feature vectors for BOW.

[3,	1,	1,	2,	0,	0,	0,	0,	0]
[3,	0,	0,	0,	1,	1,	0,	0,	0]
[3,	0,	0,	2,	0,	0,	1,	1,	1]

3.7.1. Bag of words(BOW)

BOW is frequently used in NLP that works on the basis of term frequencies (Kabir et al., 2023). This is the one of the simplest way of extracting features from texts. Considering a Bangla sample dataset of three comments as "আমার ভালো লাগছে না ।", "আমার মন খারাপ।", "আমার সাথে দুষ্টামি করো না ।", (i do not feel good. i am upset. do not mess with me), counting BOW and get the features as "আমার"(my): 3, "ভালো"(good): 1, "লাগছে"(feel): 1, "না"(no): 2, "মন"(am): 1, "খারাপ" (upset): 1, "সাথে"(with): 1, "দুষ্টামি"(mess): 1, "করো"(do): 1, for simplicity we do not consider the steps of the preprocessing. Thus the dataset will produce the feature vectors for the three comments as in Table 13.

3.7.2. TF-IDF

It is the most widely used feature extraction metric for NLP based classification tasks (Rahman and Dey, 2018) which is calculated according to Eq. (8). The number of times a word occurs in a document is counted as Term Frequency: TF (Eq. (6)) and how important a word is in a document is measured by the Inverse Document Frequency: IDF (Eq. (7)) (Hassan et al., 2022). In order to extract features from Bangla comments, we employed the well-known feature extraction metric $TFIDF$ (Term-Frequency - Inverse Document Frequency).

$$TF = \frac{\mathrm{T}}{\mathrm{K}} \tag{6}$$

$$IDF = \log_e \frac{\mathrm{D}}{\mathrm{L}} \tag{7}$$

$$TFIDF = TF - IDF \tag{8}$$

Eq. (6) stands for TF where T is the frequency of a word in a comment and K demonstrates the total number of words in that comment, while Eq. (7) is for IDF where D denote the total number of comments for SA and L is the number of comments that contain the concerned word. Therefore $TFIDF$ is determined using Eq. (8).

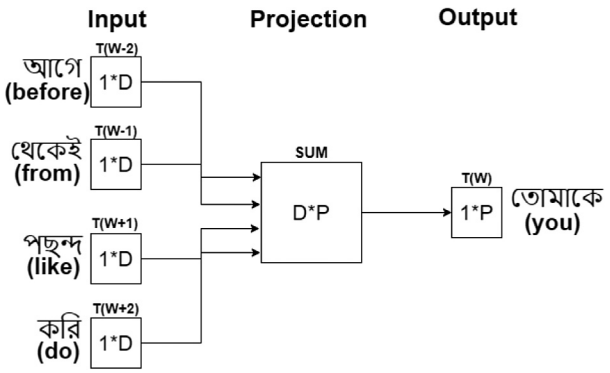

Fig. 6. Word embedding using CBOW.

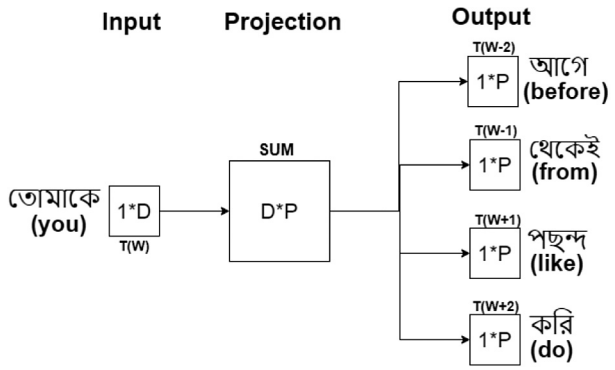

Fig. 7. Word embedding using Skipgram.

3.7.3. TF-IDF-ICF

The term ICF stands for Inverse Class Frequency introduced by Wang and Zhang (2010) which is calculated according to Eq. (9):

$$ICF = \log_e 1 + \frac{P}{Q} \tag{9}$$

$$TF \times IDF \times ICF = TF - IDF - ICF \tag{10}$$

Here P denotes the number of total categories and Q is the number of categories that contain the concerned word. Therefore $TF \times IDF \times ICF$ is measured by Eq. (10).

3.7.4. Word embeddings

Word embedding is a vector based feature extraction metric used in NLP where each word is converted into a fixed sized vector of real numbers (Sumit et al., 2018). Words are represented in a high dimensional space using word embedding where the proximity of similar words is very high, in fact the similar words together form a cluster of words. It can be implemented using Word2Vec, FastTest or GloVe methods based on the mechanism CBOW or skipgram (Sumit et al., 2018). Table 14 describes the generation process of the training samples for word embeddings. The CBOW model learns through context words (Camacho-Collados and Pilehvar, 2018) and tries to predict the target word (Fig. 6) whereas skipgram model tries to predict its neighbors (context words) using the current word (Fig. 7). The word embedding models are based on shallow neural networks of input layer, projection or hidden layer and an output layer with a softmax activation. Here D is the number of words in the vocabulary and P is the size of the embedding vector (see Figs. 6 and 7).

So far, we have discussed the pros and cons of the Word2Vec model that deals with words, a common problem with this approach is the out-of-vocabulary (OOV) words that it cannot handle. So, an extension of the Word2Vec model is introduced further to solve this issue and facilitating the advantage of morphological analysis on character level n-grams, it is known as FastText model which can also be implemented using CBOW or Skipgram (Rafat et al., 2019).

The mechanism is same with a slight change in the input layer instead of pure words it uses character n-grams to fit the neural network. Table 15 describes the generation process of character n-grams for FastText model. For example the word "ভালোবাসা"(love) after breaking into Bangla characters (vowels, consonants and their short forms *i.e.* kar and fola [allographs]) will become "ভ +(া)আ +ল +(ে া)ও +ব +(া)আ +স+(া)আ" considering the length of the character n-gram as 3, the produced n-grams are "ভা", "ভাল", "আলো", "লোব", "ওবা", "বাস", "আসা" and "সা". Instead of relying on context words like Word2Vec and FastText, another word embedding model called GloVe (Global Vector) works based on global dataset statistics and create fixed sized vectors using a co-occurrence matrix (Cerqueira et al.). Bangla-BERT (Bangla Bidirectional Encoder Representations from Transformers) is another pre-trained model developed by Google AI works on the basis of transformers and attention mechanism (Bhattacharjee et al., 2022).

We have utilized different hybrid feature extraction techniques in this work. Fig. 8 briefly describes the process of feature extraction using the hybrid method skipBangla-BERT method. Bangla-BERT (Bangla Bidirectional Encoder Representations from Transformers) has two types of encoder representation, one is Bangla-BERT base (12 encoders) and the other is Bangla-BERT large (24 encoders). We have utilized the pre-trained Bangla-BERT base model[2] along with the Skipgram shallow neural network model together. The pre-trained Bangla-BERT base consists of 12 layers, 768 hidden layers, 12 self attention heads and 110 million total number of parameters (Bhattacharjee et al., 2022) whereas Skipgram consists of a shallow neural network of 1 input layer, 1 projection or hidden layer and 1 output layer (Rafat et al., 2019). The pre-trained Bangla-BERT base model works on the basis of two mechanisms namely masked language modeling (MLM) and next sentence prediction (NSP) (Bhattacharjee et al., 2022). The first token of every sequence is represented by a special token ⟨CLS⟩ and for separating each sentences another special token ⟨SEP⟩ is used. In Fig. 8, we denote the input masked vectors, input embeddings and embedding layers as BW_i, Bt_i and E_i respectively. Let consider two consecutive dummy sentences "খুব সুন্দর", "বিষয়টি খুবই মারাত্মক"(very nice. this is very serious) will become ⟨CLS⟩ ⟨"খুব"⟩ ⟨"সুন্দর"⟩ ⟨SEP⟩ ⟨"বিষয়টি"⟩ ⟨"খুবই"⟩ ⟨"মারাত্মক"⟩. The masked input representation of the sentences will pass through the 12 encoders of the pre-trained model and built by aggregating the corresponding token embeddings, segment embeddings and position embeddings respectively (Bhattacharjee et al., 2022). Thus the encoders will generate a global padded embeddings for all the tokens. The next sequential layer is the concatenation layer having a feed forward neural network that will generate the final version of embedded vectors from the end of Bangla-BERT base model. The output of the concatenated layer will be the input of the next sequential Skipgram layer which will further projected and output a 1*768 vectors of final embeddings for each comment in the dataset.

3.8. Bangla sentiment analysis model

For performing sentiment analysis on Bangla, we have implemented Bernoulli Naive Bayes (BNB), Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) as ML models, Random Forest (RF), XGboost (XGB), Gradient Boost (GB) as EL models and Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (Bi-LSTM), Convolutional Neural Network (CNN) as DL models and CNN-BiLSTM as a hybrid DL model. Except the hybrid model, all the other algorithms are base algorithms. In this section we will describe the hybrid CNN-BiLSTM model. The architecture of the CNN-BiLSTM model is depicted

[2] https://huggingface.co/sagorsarker/bangla-bert-base

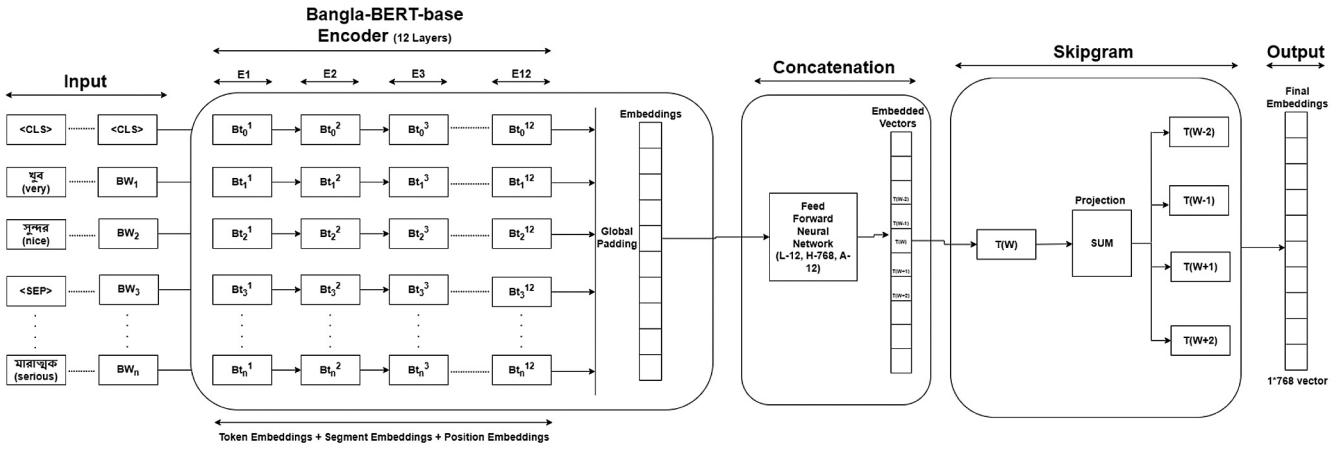

Fig. 8. Feature extraction using skipBangla-BERT mechanism.

Table 14
Training samples generation process for word embeddings using window size 5.

Source Bangla Text	Training Samples Generated From Source Text
"আমি অনেক আগে থেকেই তোমাকে পছন্দ করি", [i have liked you for a long time ago]	(অনেক, আমি) (অনেক, আগে) (অনেক, থেকেই)
"আমি অনেক আগে থেকেই তোমাকে পছন্দ করি", [i have liked you for a long time ago]	(আগে, আমি) (আগে, অনেক) (আগে, থেকেই) (আগে, তোমাকে)
"আমি অনেক আগে থেকেই তোমাকে পছন্দ করি", [i have liked you for a long time ago]	(থেকেই, অনেক) (থেকেই, আগে) (থেকেই, তোমাকে) (থেকেই, পছন্দ)
"আমি অনেক আগে থেকেই তোমাকে পছন্দ করি", [i have liked you for a long time ago]	(তোমাকে, আগে) (তোমাকে, থেকেই) (তোমাকে, পছন্দ) (তোমাকে, করি)
"আমি অনেক আগে থেকেই তোমাকে পছন্দ করি", [i have liked you for a long time ago]	(পছন্দ, থেকেই) (পছন্দ, তোমাকে) (পছন্দ, করি)

Table 15
Character n-gram generation process of FastText model.

Word	Character Analysis	N-gram Length	Character N-grams
ভালোবাসা (love)	ভ +(া)আ +ল +(ে া)ও +ব +(া)আ +স+(া)আ	3	ভা, ভাল, আলো, লোব, ওবা, বাস, আসা, সা
ভালোবাসা (love)	ভ +(া)আ +ল +(ে া)ও +ব +(া)আ +স+(া)আ	4	ভাল, ভালো, আলোব, লোবা, ওবাস, বাসা, আসা
শুভকামনা (good wishes)	শ +(ু)উ + ভ + ক+ (া)আ + ম + ন +(া)আ	3	শু, শুভ, উভক, কাম, আমন, মনা, না
শুভকামনা (good wishes)	শ +(ু)উ + ভ + ক+ (া)আ + ম + ন +(া)আ	4	শুভ, শুভক, উভকা, ভকাম, কামন, আমনা, মনা

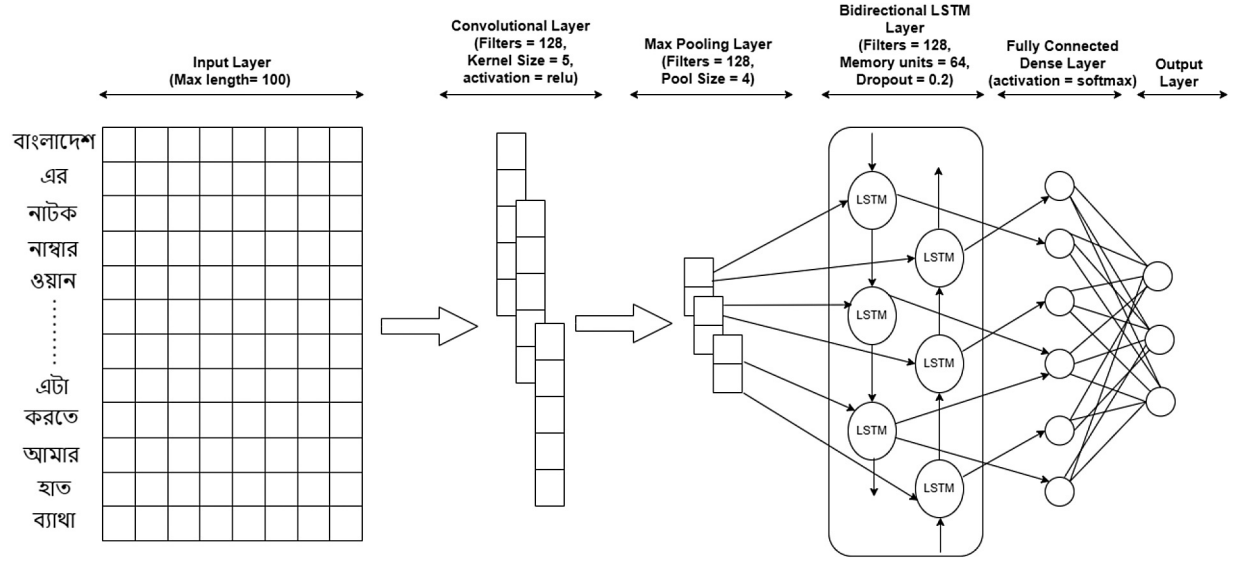

Fig. 9. Architecture of the CNN-BiLSTM model.

in Fig. 9. It consists of six consecutive layers such as the input layer, convolutional layer, max pooling layer, bidirectional LSTM layer, fully connected dense layer and the output layer.

The first layer is an embedding layer (input layer), it is generally used to convert the integer encoded word indices to dense vectors. It takes the input sequence and converts each word index to a fixed size dense vector of 100 (*max length*). *Max length* is the size of the word vectors. The *input length* parameter specifies the length of the input sequences (number of words). The second layer is a one-dimensional convolutional layer with 128 filters and a *kernel size* of 5. The activation function used is *ReLU* (Rectified Linear Unit), which introduces non-linearity to the model. The third layer is the max

pooling layer with a *pool size* of 4. It is used to reduce the spatial dimensions of the data and capture the most important features from the convolutional layer. The fourth layer contains a Bidirectional LSTM (Long Short-Term Memory) with 64 units (32 forward memory units and 32 backward memory units). Bidirectional LSTMs process the input sequence in both forward and backward directions, allowing the model to capture contextual information from both sides of the sequence. The *dropout* value of 0.2 and *recurrent_dropout* of 0.2 are used to apply dropout regularization to the LSTM layer to prevent overfitting. The fifth layer is the fully-connected layer where each neuron is connected to every neuron in the previous layer, and each connection has its own weight. Thus, it is very expensive in terms of memory (weights) and computation (connections). This layer flattens the input feature representations into a feature vector and performs the function of high-level reasoning. This layer functions by $softmax$ activation. The output layer is responsible for providing the prediction for test instance to either any sentiment category from 15 predefined categories. So, in the output layer, there are 15 channels available where each channel corresponds to a predefined sentiment category. Section 4 describes more about the numerical details of different layers of CNN-biLSTM model in brief.

4. Experimental results analysis and discussion

As datasets and resources are the preliminary obstacle for dealing with Bangla NLP based works, so dataset is the prime concern when we are dealing with sentiment analysis on Bangla language (Kabir et al., 2023; Nafisa et al., 2023). So, the principle target of this work is to develop a new comprehensive sentiment dataset for Bangla language. According to our goal, we have developed a large Bangla sentiment dataset of 203,463 comments. We have examined the proposed work by using Python language (Integrated development environment: Jupyter Notebook, Google Colab) with device specifications: Processor — Intel(R) Core(TM) i5-7200U CPU @2.50 GHz 2.71 GHz, RAM - 8.00 GB (7.90 GB usable), System type — 64-bit operating system, x64-based processor, OS — Windows 10 Pro edition. To observe its baseline evaluation we have implemented different machine learning (ML), ensemble learning (EL) and deep learning (DL) algorithms. To evaluate the performance of the proposed Bangla sentiment analysis models, we have used several evaluation metrics such as Accuracy, Precision, Recall, and F1-score. They are formulated as follows:

$$Accuracy = \frac{TP + TN}{P + N} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F \times measure = \frac{2 - Precision - Recall}{Precision + Recall} \tag{14}$$

Where,
TP: correct positive prediction,
FP: incorrect positive prediction,
TN: correct negative prediction,
FN: incorrect negative prediction,
P: TP+FP,
N: TN+FN.

Classification accuracy refers to the ratio of correct predictions to total number of predictions made by the built model (Eq. (11)). Precision is the ratio of true positives to the true positives and false positives prediction (Eq. (12)). Recall is defined as the ratio of true positives to the true positives and false negatives (Eq. (13)). F1-score or F-measure is the balance measure to express the performance in a single quantity. It is the harmonic mean (Eq. (14)) of precision and recall (Prottasha et al., 2022). For performing sentiment analysis on Bangla, we have implemented Bernoulli Naive Bayes (BNB), Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) as ML models, Random Forest (RF), XGboost (XGB), Gradient Boost (GB) as EL models and Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Convolutional Neural Network (CNN) as DL models and CNN-BiLSTM as a hybrid DL model. In this section, we will describe the parameter tuning process for the best fit model, experimental results of different algorithms, their comparisons, different findings etc.

Table 16
Split the dataset into training and test sets based on 15 categories.

Category	Total samples	Training set	Test set
Love	56,631	45,305	11,326
Enthusiasm	37,965	30,372	7593
Happy	26,596	21,277	5319
Fun	24,351	19,481	4870
Surprise	3501	2801	700
Relief	765	612	153
Angry	27,054	21,644	5410
Sad	6101	4881	1220
Sexual	5854	4684	1170
Disgust	3888	3111	777
Boring	2494	1996	498
Worry	1810	1448	362
Fear	1442	1154	288
Hate	1083	867	216
Neutral	3928	3143	785
Total	**203,463**	**162,776**	**40,687**

Table 17
Split the dataset into training and test sets based on 3 categories.

Category	Total samples	Training set	Test set
Positive	149,809	119,848	29,961
Negative	49,726	39,745	9981
Neutral	3928	3143	785
Total	**203,463**	**162,776**	**40,687**

4.1. Split the dataset

The proposed dataset contain a total of 203,463 comments from social media. We split our dataset into 80% for training dataset and 20% for test dataset. The overview of the distribution of training and test sets based on 15 and 3 categories are shown in Tables 16 and 17 respectively. We have to use the training datasets to train different learning models so that they can predict unknown instances according to their training, and to know about how well the models are being trained, we have to use the test dataset to measure its performance through different metrics such as precision, recall, f1-score, accuracy and so on.

4.2. CNN-BiLSTM model

The architecture of the CNN-BiLSTM model is depicted in Fig. 9. In this section we will briefly describe about the parameters setting, experimental performance of different layers and final outcomes.

4.2.1. Setting parameters

The input layer contain the word embedding vectors of a fixed size. So, a question can be arisen that what will the *max length* of each embedding vector? We have tuned this parameter (*max length*). The impact of max length parameter on CNN-BiLSTM model is given in Table 18. For max length tuning, we have taken the values of max length as 32, 64, 100, 128 and 256. The experimental results show that the highest training accuracy (0.95) and validation accuracy (0.93) as well as the lowest training loss (0.35) and validation loss (0.36) are found when max length is 100. Therefore, we set 100 as the value for *max length*.

Table 18
The impact of max length on CNN-BiLSTM model.

Max length	Training accuracy	Validation accuracy	Training loss	Validation loss
32	0.87	0.87	0.47	0.43
64	0.91	0.89	0.41	0.39
100	**0.95**	**0.93**	**0.35**	**0.36**
128	0.90	0.90	0.40	0.44
256	0.88	0.91	0.44	0.46

Table 19
The impact of learning rate on CNN-BiLSTM model.

Learning rate	Training accuracy	Validation accuracy	Training loss	Validation loss
1e−2	0.85	0.85	0.58	0.51
1.5e−2	0.88	0.84	0.51	0.54
2e−3	0.92	0.89	0.47	0.45
2.5e−3	**0.96**	**0.94**	**0.40**	**0.39**
3e−2	0.89	0.90	0.48	0.43
3.5e−3	0.90	0.88	0.46	0.42

Table 20
The impact of batch size on CNN-BiLSTM model.

Batch size	Training accuracy	Validation accuracy	Training loss	Validation loss
5	0.91	0.90	0.62	0.65
6	0.92	0.89	0.65	0.63
7	0.92	0.90	0.60	0.55
8	0.93	0.92	0.57	0.58
30	0.95	0.93	0.57	0.54
32	**0.96**	**0.94**	**0.53**	**0.54**

Table 21
The impact of epochs on CNN-BiLSTM model.

Epochs	Training accuracy	Validation accuracy	Training loss	Validation loss
200	0.84	0.81	0.71	0.68
300	0.87	0.82	0.69	0.61
400	0.92	0.85	0.58	0.60
500	**0.95**	**0.93**	**0.42**	**0.45**
600	0.95	0.93	0.46	0.45
700	0.95	0.92	0.49	0.47

In a similar fashion the experimental results show that the *learning rate* of 2.5e−3 is the best fitted value for CNN-BiLSTM model. The impact of *learning rate* is given in Table 19. It produces a training accuracy of 0.96 and validation accuracy of 0.94 which outperform the others.

We have also tuned the *batch size* parameter for the proposed CNN-BiLSTM model. The total number of training samples those are used in one epoch is referred to as *batch size* (Alam et al., 2017). Table 20 briefly describes the impact of batch size on CNN-BiLSTM model. For the sake of *batch size* tuning, we have taken the values of *batch size* as 5, 6, 7, 8, 30 and 32. For the small values of *batch sizes* (5, 6, 7 and 8), the training and validation accuracy's are slightly lower while for the large values (30 and 32) the training and validation accuracy's are slightly better. We have recorded the best performance with *batch size* 32 and set it as our optimal value for *batch size*.

Epoch is another hyper-parameter that needs to be tuned. Epoch can also be termed as iteration or cycle. The impact of epochs on CNN-BiLSTM model is shown in Table 21. For the tuning purpose of no. of epochs, we have considered the values as 200, 300, 400, 500, 600 and 700. For small values of epochs (200, 300 and 400) the training and validation accuracy's are slightly lower while for the large values (500, 600 and 700) the training and validation accuracy's are slightly better but same. The accuracy did not increase after 500 epochs. So, we have set the epochs as 500 as optimal value.

Table 22
Parameters of CNN-BiLSTM model with their optimal measurements.

Parameter	Measurement/Value
Max length	100
Conv1D (filters)	128
Conv1D (kernel size)	5
MaxPooling1D (pool size)	4
Bi-LSTM memory units	64
Batch size	32
Learning rate	2.5e−3
Dropout	0.2
Activation function	ReLU
Predictive function	Softmax
Loss function	Sparse categorical cross-entropy
Optimizer	Adam
Metrics	Accuracy
Epochs	500

Table 22 illustrates the parameters of CNN-BiLSTM model with their optimal measurements. The one-dimensional convolutional layer contains 128 filters and a kernel size of 5, the pool size of the one dimensional max pooling layer is 4, the total memory units in the bidirectional LSTM layer is 64 having a dropout of 0.2 to prevent overfitting. The activation and predictive functions are ReLU (Rectified Linear Unit) and Softmax respectively. Sparse categorical cross-entropy is the loss function, adam is the optimizer and accuracy is the performance metrics for CNN-BiLSTM model.

4.2.2. Analysis of different layers in CNN-BiLSTM model

Fig. 9 depicts the architecture of the proposed CNN-BiLSTM model. The first layer is an embedding layer (input layer), it is generally used to convert the integer encoded word indices to dense vectors. It takes the input sequence and converts each word index to a fixed size dense vector of 100 (max length). The second layer is the convolutional layer with 128 filters and a kernel size of 5. The activation function used is ReLU (Rectified Linear Unit), which introduces non-linearity to the model. The kernel weights of the initial 16 filters of the convolutional layer are shown in Fig. 10. The change in the weights of the kernel in the first 16 consecutive filters are random. The weights in the output channel 1 changes drastically, the weight value starts from 0.038 and gradually decreases to negative values and then again increases to 0.084. Again in the output channel 2 of the convolutional layer, the initial weight value is −0.02, further it decreases more up to weight value −0.043, then again the value increased to 0.046. The final weight values are all positive on the initial 7 output channels of the convolutional layer while for 8th, 9th, 11th and 16th output channels are all negative weights. Fig. 11 shows the kernel weights adaptation schemes of final 16 output channels of the convolutional layer. The 113th output channel of the convolutional layer starts with a positive weight 0.043 which finally adapts to −0.008. The 128th output channel starts with a positive weight 0.058 and then drastically fluctuates to −0.06, then again rapidly increases to 0.039, then again decreases to 0.031 and finally stables to weight value 0.069. The third layer is the max pooling layer with a pool size of 4. It is used to reduce the spatial dimensions of the data and capture the most important features from the convolutional layer.

The activation status of the initial and final 16 filters of the max pooling layer in CNN-BiLSTM model are shown in Figs. 12 and 13 respectively. Activation values for some of the channels are zero (channels 1, 4, 5, 7, 10, 11, 13, 14, 16, 113, 118, 119, 121, 122, 124, 127 and 128) while 0.0289, 0.001925 and 0.01878 are the activation values for channels 2, 15 and 126 respectively. So, the spatial dimensions are being reduced to a great extent compared to the convolutional layer. The fourth layer contains a Bidirectional LSTM (Long Short-Term Memory) with 64 units (32 forward memory units and 32 backward memory units). Bidirectional LSTMs process the input sequence in both forward

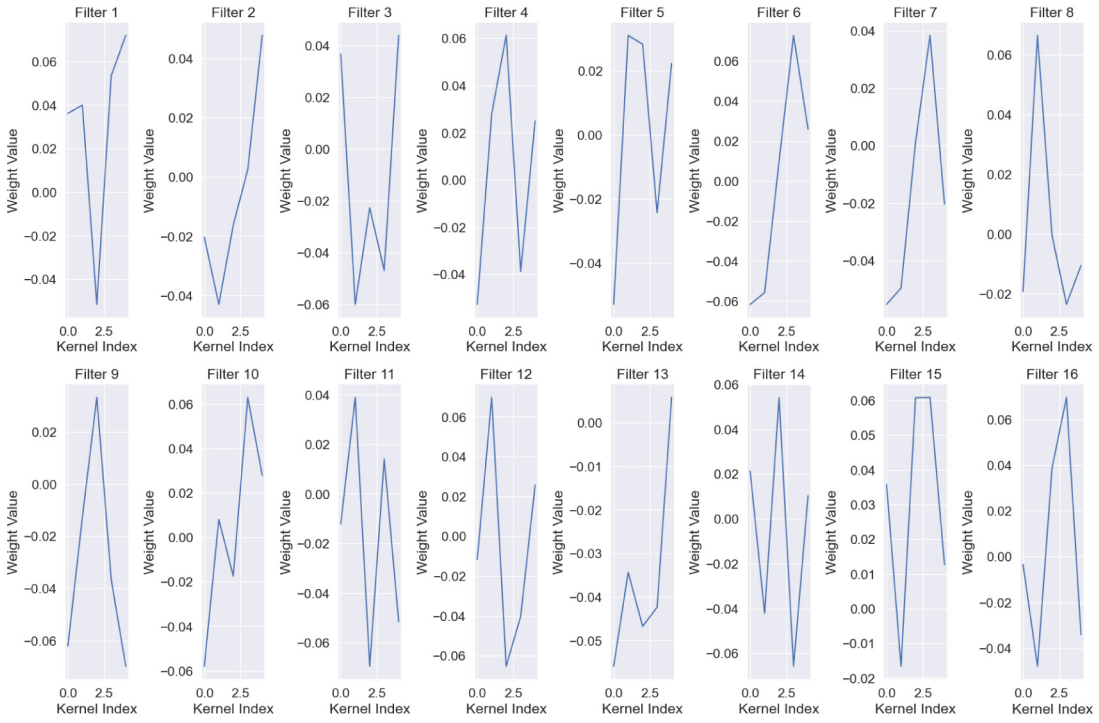

Fig. 10. Kernel weights of Convolutional layer in CNN-BiLSTM model (initial 16 filters).

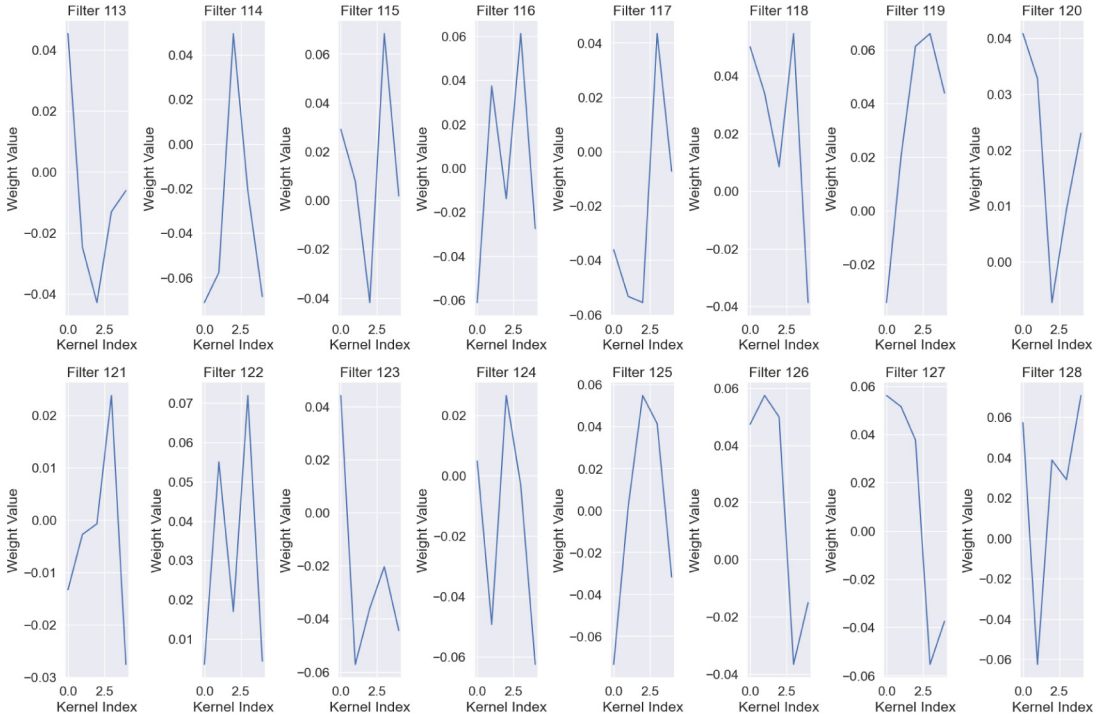

Fig. 11. Kernel weights of Convolutional layer in CNN-BiLSTM model (final 16 filters).

and backward directions, allowing the model to capture contextual information from both sides of the sequence. The dropout value of 0.2 and recurrent_dropout of 0.2 are used to apply dropout regularization to the LSTM layer to prevent overfitting. The brief graphical overview of the memory units of the bidirectional LSTM layer in CNN-BiLSTM model is shown in Fig. 14. The fourth and final layer is the fully connected dense layer that is responsible for providing the prediction for test instance to either any sentiment category from 15 predefined categories. So, in the output layer, there are 15 channels available where each channel corresponds to a predefined sentiment category. Fig. 15 depicts the activation status of the fully connected dense layer for the proposed CNN-BiLSTM model.

4.3. Experimental results analysis

The proposed work mainly focuses on creating a comprehensive dataset for Bangla SA and discovering an efficient feature extraction metric, then apply various ML, EL and DL algorithms and make

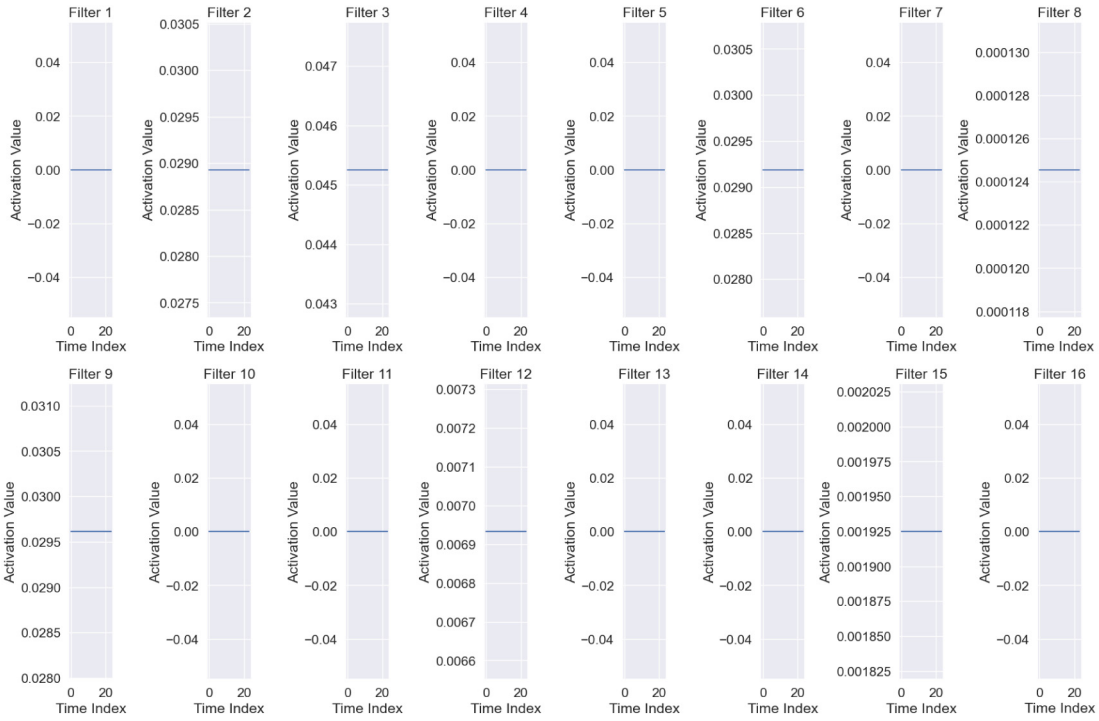

Fig. 12. Activation status of Max pooling layer in CNN-BiLSTM model (initial 16 filters).

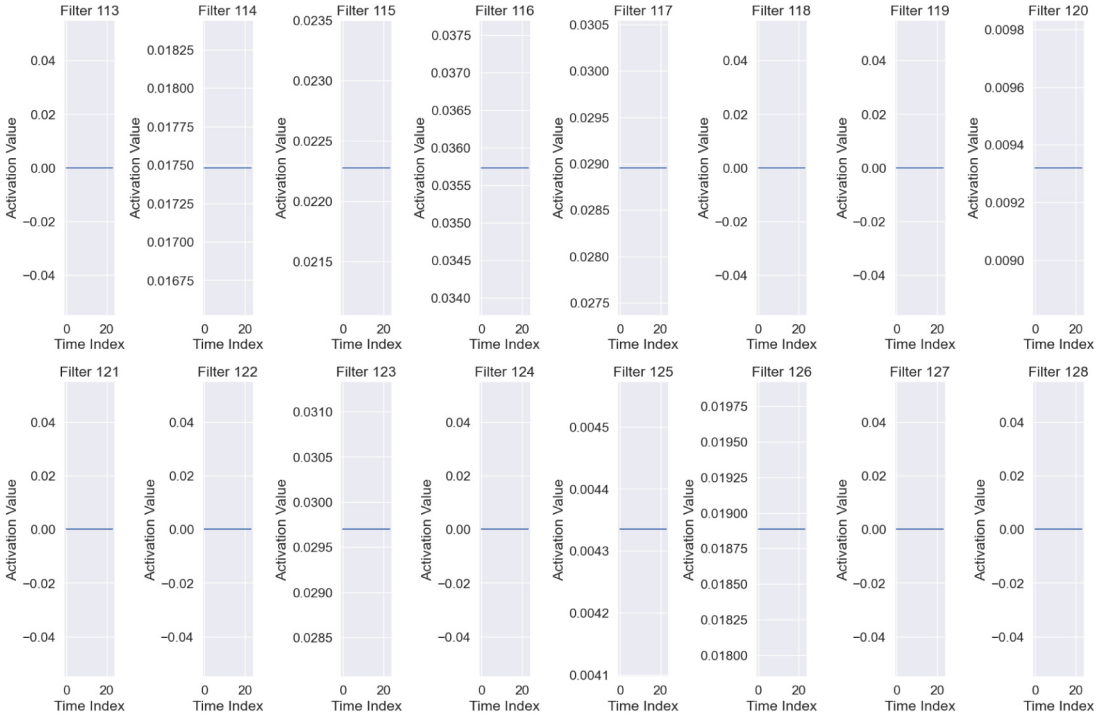

Fig. 13. Activation status of Max pooling layer in CNN-BiLSTM model (final 16 filters).

a comparative analysis among them to find the optimal model as well as feature metric. Table 23 describes the proposed dataset at a glance. In this work, we have developed total 21 different hybrid feature extraction techniques such as BOW+2-Gram, BOW+3-Gram, TF-IDF+2-Gram, TF-IDF+3-Gram, TF-IDF-ICF+2-Gram, TF-IDF-ICF+3-Gram, Word2Vec+CBOW(gensim), Word2Vec+Skipgram (gensim), Word2Vec+CBOW+Skipgram (gensim), Word2Vec+CBOW (tensorflow), Word-2Vec+Skipgram (tensorflow), Word2Vec+CBOW+Skipgram (tensorflow), FastText+CBOW, FastText+Skipgram, FastText+CBOW+Skipgram, GloVe+CBOW, GloVe+Skipgram, GloVe+CBOW+Skipgram, Bangla-BERT+CBOW, skipBangla-BERT, Bangla-BERT+CBOW+Skipgram and implemented ML, EL and DL algorithms to evaluate them. In this work, we have implemented Bernoulli Naive Bayes (BNB), Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) as ML models, Random Forest (RF), XGboost (XGB), Gradient Boost (GB) as EL models and Recurrent Neural Network (RNN), Long Short Term Memory (LSTM),

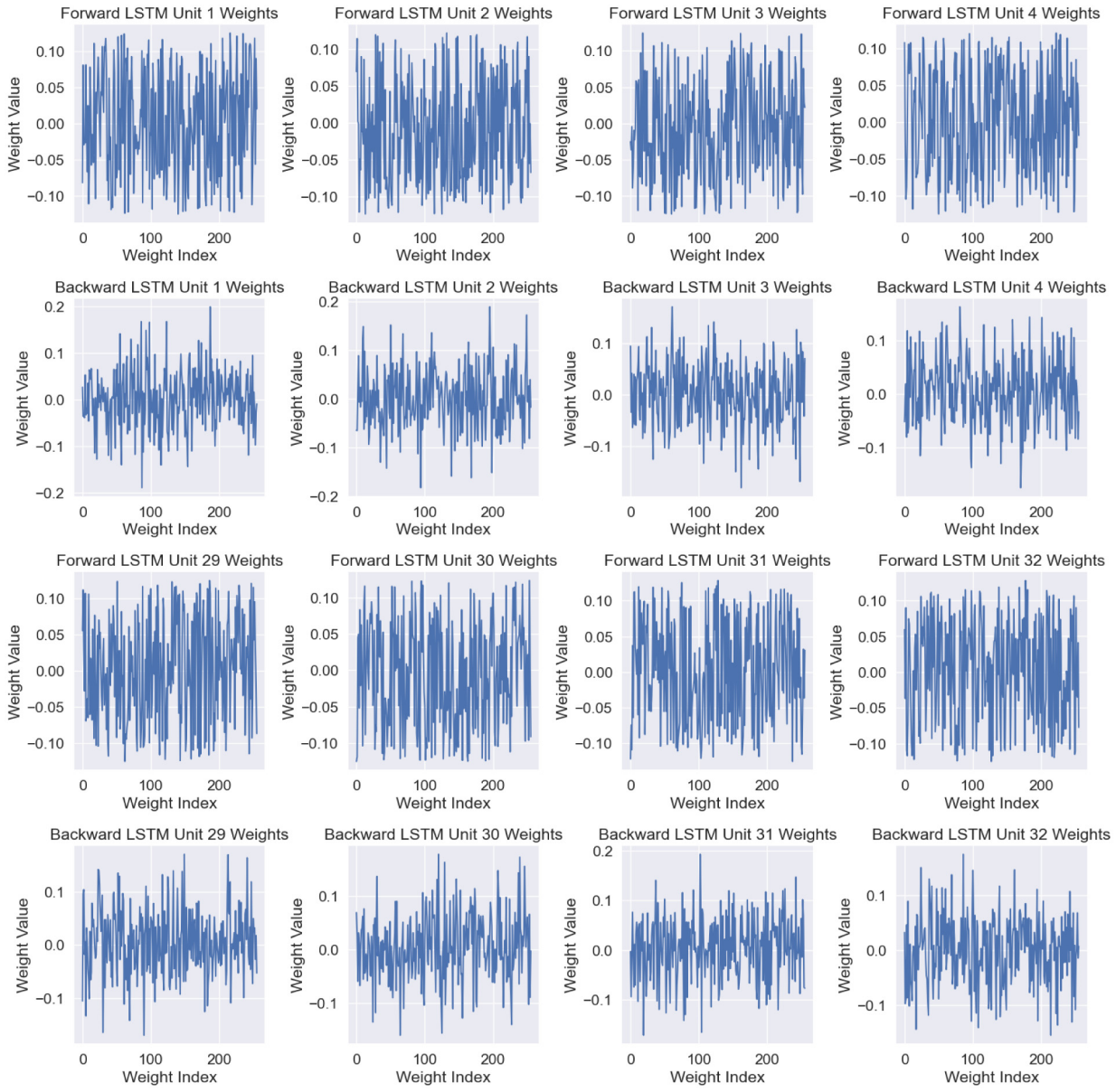

Fig. 14. Memory units of Bidirectional LSTM layer in CNN-BiLSTM model (initial and final 4 memory units for both forward and backward steps).

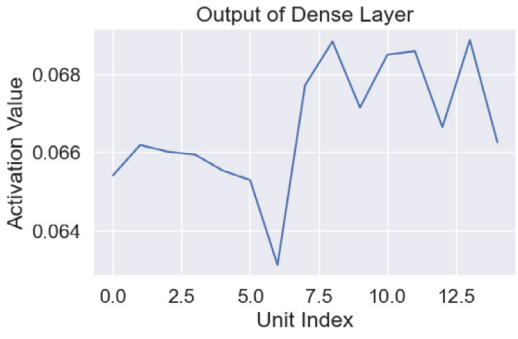

Fig. 15. Activation status of fully connected dense layer in CNN-BiLSTM model.

Bidirectional LSTM (Bi-LSTM), Convolutional Neural Network (CNN) as DL models and CNN-BiLSTM as a hybrid DL model.

The brief overview of the category-wise data collection is given in Table 4 that shows the developed Bangla sentiment dataset is imbalanced which can generate overfitted results in different learning models. So, we have used the synthetic minority oversampling technique (SMOTE) to balance our dataset as well as get rid of this overfitting situation. Table 24 demonstrates the effect of different model performance after balancing the dataset. For both 15 and 3 sentiment categories, BNB has the worst results compared to other algorithms while SMOTE significantly improves its accuracy from 57.91% to 66.85% (15 categories) and 75.26% to 83.31% (3 categories). From the ML domain SVM outperforms the other 4 algorithms for both 15 and 3 categories along with and without SMOTE. Results from ML domain algorithms without SMOTE show that 15 categories provide better results than 3 categories in DT, LR and SVM algorithms while BNB and KNN provide better results with respect to 3 categories. But after balancing the dataset using SMOTE, all the model performance except DT show that results of 3 categories are better. From the EL domain RF outperforms the other 2 algorithms for both 15 and 3 categories along with and without SMOTE. The results of all three models with SMOTE from EL have been increased significantly. SMOTE significantly improves the accuracy of

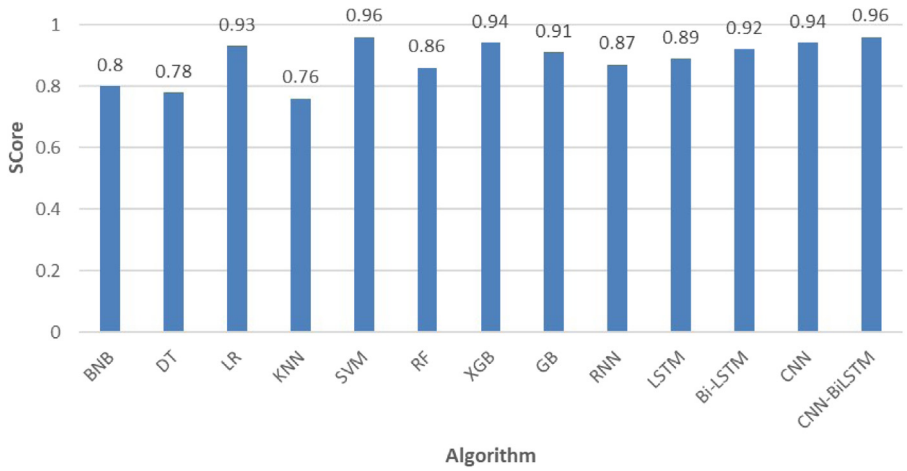

Fig. 16. Precision scores of the implemented algorithms.

Table 23
The proposed dataset at a glance.

Feature	Measurement/Result	Comment
Dataset size	203,463	Moderately good
Total no. of words	204,6150	Good
Total no. of characters	116,60,683	Good
Vocabulary size	165,319	Good
Data annotation quality?	94.67%	Accuracy
Dataset balance?	No	Not Good
No. of categories considered	15 and 3	Covering more sentiments
Function words	32.84%	Good
Top 10 unigrams	All are stopwords	As expected
Average word length	5.69	N/A
Average unique word length	7.84	N/A
Zipf's law	Roughly follow	Good
Hapax legomina	More than half of the dataset	As expected
Vocabulary growth rate	0.053	Good

RF from 80.98% to 85.77% and 80.13% to 91.16% for 15 and 3 categories respectively. From the DL domain CNN-BiLSTM outperforms the other 4 algorithms for both 15 and 3 categories along with and without SMOTE. The hybrid model outperforms all the algorithms from ML, EL and DL and achieves an accuracy of 95.71% with SMOTE. From the above results analysis, it is observed that there is a relationship between the number of categories and the model performance. Let the number of categories and the model performance be c and p respectively. The relationship between this two parameters are as follows:

$$p \propto \frac{1}{c}$$

When the number of categories increases to 15 categories, the performance of the model decreases and vice versa. So, for 3 categories, the performance of most models is better than for 15 categories scenario.

The obtained results of applied algorithms using 21 different hybrid feature metrics based on 15 and 3 categories are shown in Tables 25 and 26 respectively. The best achieved results are highlighted in bold signs. Among ML algorithms based on 15 sentiment categories, SVM (classification accuracy 84.88%) outperforms all other methods using skipBangla-BERT while EL and DL algorithms achieved highest accuracy 85.77% (Bangla-BERT+CBOW+Skipgram) and 90.24% skipBangla-BERT using RF and the hybrid model CNN-BiLSTM. The feature metric skipBangla-BERT performs better in most of the cases compared to the other feature metrics. So, we have shown two separate tables to show the best results found from this feature metric. The obtained results for 15 and 3 categories using skipBangla-BERT feature metric are summarized and shown in Tables 27 and 28 respectively. Examining skipBangla-BERT mechanism on 15 sentiment categories, it is observed that SVM, RF and CNN-BiLSTM model achieved the best results of 84.88%, 85.34% and 90.24% accuracy from ML, EL and DL doamins respectively. In case of the results obtained based on 3 sentiment categories (see Tables 26 and 28): SVM, XGB and CNN-BiLSTM acquired highest accuracy 92.37%, 92.55% and 95.71% from ML, EL and DL respectively using the hybrid feature extraction method skipBangla-BERT. The TF-IDF-ICF also performed well and obtained better results than traditional TF-IDF. The experimental results show that Skipgram outperforms CBOW (observe Tables 25 and 26).

To measure the performance of the implemented algorithms, we have considered the commonly used feature metrics such as precision, recall, f1-score and accuracy. Precision is the ratio of true positives to the true positives and false positives prediction (Eq. (12)). The precision scores of all the implemented algorithms are shown in Fig. 16. SVM and CNN-BiLSTM models achieved the highest precision score of 0.96 whereas XGB and CNN acquired 2nd highest precision score of 0.94 and LR obtained 0.93. These 5 models are classifying unknown test instances more precisely than the others. The average precision scores for DL, EL and ML domains are 0.92, 0.90 and 0.85 respectively. From the implemented algorithms KNN produced the worst precision score of 0.76. Recall is defined as the ratio of true positives to the true positives and false negatives (Eq. (13)). Fig. 17 depicts the recall scores of all the implemented algorithms. LSTM got the highest recall value of 0.99 and CNN, CNN-BiLSTM and RNN obtained 2nd highest recall of 0.98 and RF obtained 0.94. Very high precision and recall values almost near to 1 is desirable. It proves the efficiency and effectiveness of the trained models. The average recall values for DL, EL and ML domains are 0.98, 0.93 and 0.86 respectively. As like the value of precision, KNN also produced the worst recall score of 0.82. F1-score or F-measure is the balance measure to express the performance in a single quantity. It is the harmonic mean of precision and recall (Eq. (14)). F1-scores of all the implemented algorithms are shown in Fig. 18. The hybrid model CNN-BiLSTM obtained the highest f1-score of 0.96 whereas CNN got the 2nd highest f1-score of 0.95 and Bi-LSTM and SVM achieved 0.94. The average f1-scores from DL, EL and ML domains are 0.94, 0.91 and

Table 24
Effect of different model performance after balancing the dataset.

Domain	Algorithm	Accuracy without SMOTE for 15-categories (%)	Accuracy with SMOTE for 15-categories (%)	Accuracy without SMOTE for 3-categories (%)	Accuracy with SMOTE for 3-categories (%)
ML	BNB	57.91	66.85	75.26	83.31
	DT	78.15	84.37	76.89	83.21
	LR	80.12	83.77	78.34	88.74
	KNN	59.89	72.69	69.95	81.58
	SVM	80.52	84.88	79.58	92.37
EL	RF	80.98	85.77	80.13	91.16
	XGB	79.47	83.59	79.92	92.55
	GB	75.67	84.91	80.03	91.87
DL	RNN	80.19	88.91	82.17	93.14
	LSTM	81.11	86.90	83.23	94.37
	Bi-LSTM	81.73	85.96	82.65	94.68
	CNN	80.38	88.33	82.91	94.55
	CNN-BiLSTM	82.16	90.24	83.64	95.71

Table 25
Obtained results of applied algorithms using different feature extraction techniques based on 15 sentiment categories.

Feature	Accuracy (%)												
	BNB	DT	LR	KNN	SVM	RF	XGB	GB	RNN	LSTM	Bi-LSTM	CNN	CNN-BiLSTM
BOW+2-Gram	61.13	73.25	74.51	39.78	77.63	80.31	79.88	80.03	N/A	N/A	N/A	N/A	N/A
BOW+3-Gram	61.33	73.29	74.50	39.91	77.67	80.23	79.87	79.98	N/A	N/A	N/A	N/A	N/A
TF-IDF+2-Gram	65.72	78.93	77.67	42.96	80.34	82.52	82.31	82.19	N/A	N/A	N/A	N/A	N/A
TF-IDF+3-Gram	65.81	78.97	77.98	43.10	80.77	82.63	81.69	82.33	N/A	N/A	N/A	N/A	N/A
TF-IDF-ICF+2-Gram	64.33	80.14	78.40	43.22	82.19	83.12	82.13	82.44	N/A	N/A	N/A	N/A	N/A
TF-IDF-ICF+3-Gram	64.27	80.31	78.42	43.23	82.76	83.12	82.15	82.53	N/A	N/A	N/A	N/A	N/A
Word2Vec+CBOW (gensim)	28.71	58.16	60.67	57.90	62.61	71.18	70.13	70.69	72.39	73.65	74.68	75.89	82.31
Word2Vec+Skipgram (gensim)	35.95	57.84	62.94	60.62	67.98	71.79	71.05	69.98	74.23	75.63	76.92	80.39	83.26
Word2Vec+CBOW+ Skipgram (gensim)	28.55	57.81	60.43	56.89	64.69	71.19	69.72	68.87	73.13	76.98	75.97	80.14	82.79
Word2Vec+CBOW (tensorflow)	24.13	55.79	51.07	43.26	60.53	64.67	63.19	62.74	79.54	81.26	82.39	81.37	83.76
Word2Vec+Skipgram (tensorflow)	25.26	57.81	54.32	44.17	61.29	64.98	63.55	61.87	79.98	82.05	82.31	82.91	82.77
Word2Vec+CBOW+ Skipgram (tensorflow)	24.89	55.86	53.68	43.29	60.95	64.83	63.14	61.68	78.81	82.39	80.39	79.68	81.26
FastText+CBOW	56.82	67.64	71.62	**72.69**	72.86	76.21	73.42	72.87	77.37	79.81	80.45	80.11	82.38
FastText+Skipgram	**66.85**	67.78	71.61	72.58	72.59	76.29	74.51	73.27	79.91	81.29	84.25	83.22	84.27
FastText+CBOW+ Skipgram	59.87	67.74	71.64	72.62	72.89	76.23	75.85	72.96	78.63	80.94	82.71	81.21	83.57
GloVe+CBOW	55.72	66.52	71.23	57.89	72.43	74.89	75.76	70.83	76.39	78.35	79.42	78.98	80.53
GloVe+Skipgram	65.35	67.39	70.54	58.93	71.99	75.61	74.38	69.87	78.13	75.53	81.32	80.32	81.26
GloVe+CBOW+ Skipgram	61.37	65.74	71.27	64.79	70.99	72.94	71.29	70.89	73.41	78.91	79.47	78.96	80.93
Bangla-BERT+CBOW	63.47	84.26	83.49	67.91	84.67	85.79	83.41	84.62	88.89	86.41	86.49	87.91	89.13
skipBangla-BERT	64.36	83.19	**83.77**	66.43	**84.88**	85.34	**83.59**	**84.91**	**88.91**	85.73	**85.96**	**88.33**	**90.24**
Bangla-BERT+CBOW+ Skipgram	64.53	**84.37**	83.73	67.05	84.79	**85.77**	82.97	84.77	88.86	**86.90**	86.81	88.13	89.91

0.85 respectively. Among all the models, KNN got the worst f1-score of 0.79, as it obtained the worst precision and recall values. Among all the implemented algorithms, KNN is the only learning model that is lazy learner. It has no training phase. This is the reason for its worst outcomes. The proposed dataset contain human Bangla comments, so it is very rear to have the use of standard language always. Rather it can have local Bangla words, local folk words, slang's etc. So, training is a must to get promising outcomes from a model. Except KNN, the other models are eager learners and they produced significant results in both 15 and 3 categories versions of the dataset.

The performance of a learning model depends mostly on the extracted features from the dataset. Feature extraction is very important in text mining related works. So, it is an important aspect to know which feature extraction method is more efficient with our present task at hand. In our work, we have examined different feature extraction techniques. The average performance of different feature extraction methods are summarized in Fig. 19, where Bangla-BERT outperforms all other methods with an average accuracy of 92.24%. FastText model shows an accuracy of 81.09% being the 2nd highest metric for feature extraction. TF-IDF-ICF metric outperforms the traditional TF-IDF and

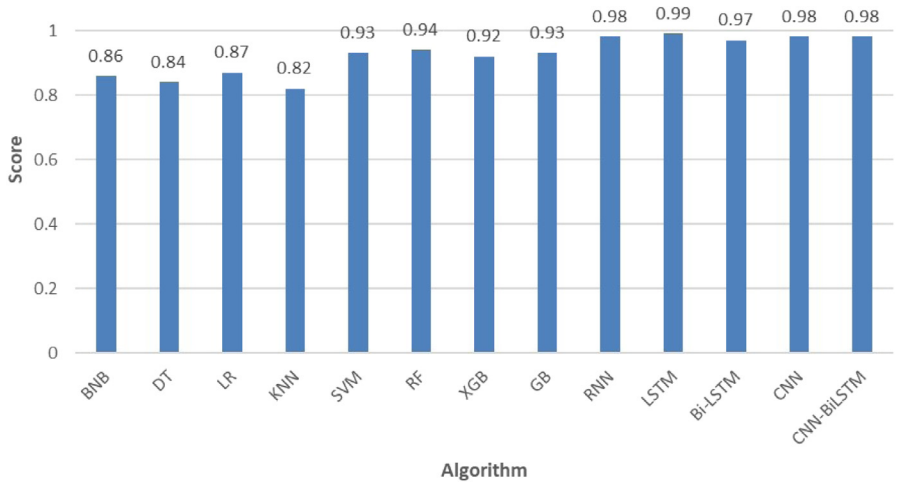

Fig. 17. Recall scores of the implemented algorithms.

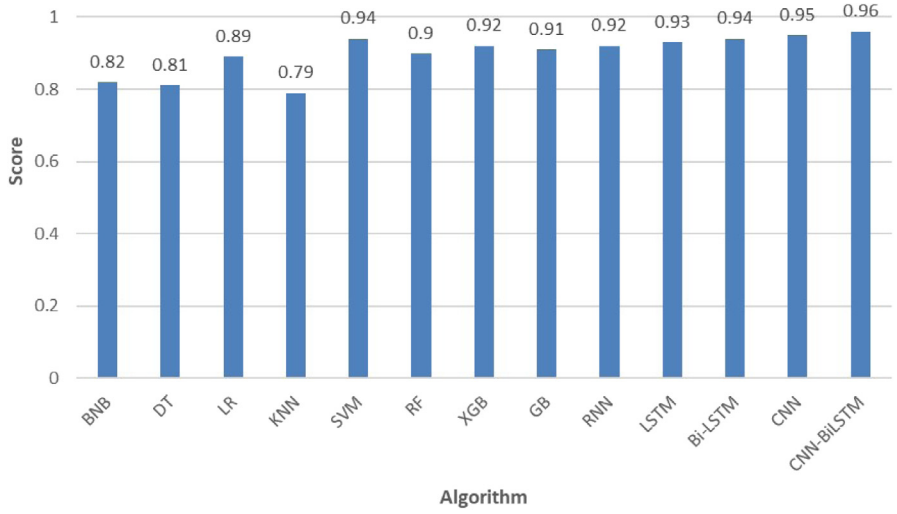

Fig. 18. F1-scores of the implemented algorithms.

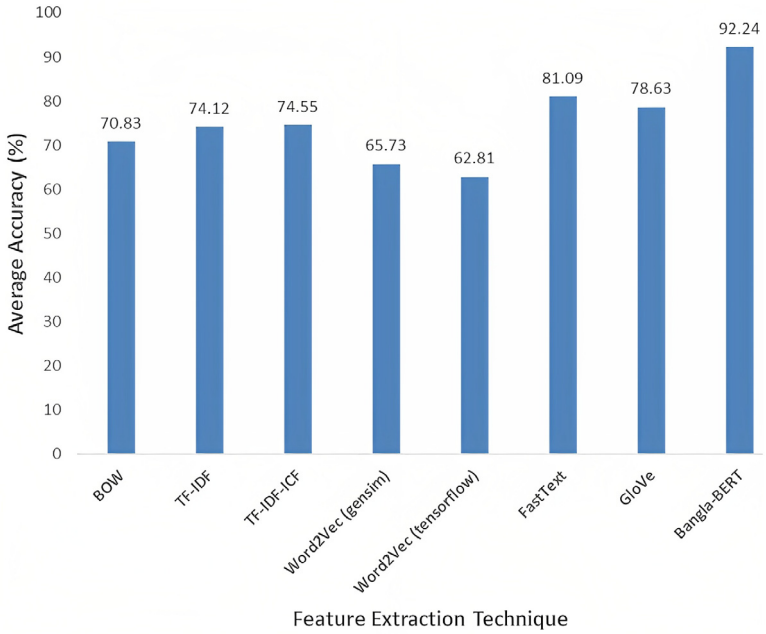

Fig. 19. Average performance of different feature extraction techniques.

Table 26
Obtained results of applied algorithms using different feature extraction techniques based on 3 sentiment categories.

Feature	Accuracy (%)												
	BNB	DT	LR	KNN	SVM	RF	XGB	GB	RNN	LSTM	Bi-LSTM	CNN	CNN-BiLSTM
BOW+2-Gram	80.32	82.37	83.04	57.33	82.43	83.78	82.47	82.88	N/A	N/A	N/A	N/A	N/A
BOW+3-Gram	80.13	81.69	82.34	59.75	81.66	83.96	82.39	81.89	N/A	N/A	N/A	N/A	N/A
TF-IDF+2-Gram	82.69	88.23	88.04	58.39	89.47	90.33	90.45	89.97	N/A	N/A	N/A	N/A	N/A
TF-IDF+3-Gram	83.05	88.21	88.19	61.46	89.48	90.41	90.23	90.17	N/A	N/A	N/A	N/A	N/A
TF-IDF-ICF+2-Gram	83.18	88.95	88.61	58.20	89.92	91.07	90.86	89.99	N/A	N/A	N/A	N/A	N/A
TF-IDF-ICF+3-Gram	**83.31**	89.02	**88.74**	59.16	90.01	**91.16**	89.56	90.45	N/A	N/A	N/A	N/A	N/A
Word2Vec+CBOW (gensim)	65.82	78.83	80.84	81.11	81.86	85.78	84.35	83.26	83.49	83.99	84.74	84.56	89.57
Word2Vec+Skipgram (gensim)	69.19	79.92	81.46	80.31	83.20	86.44	84.53	84.36	82.43	87.94	86.24	85.41	90.13
Word2Vec+CBOW+ Skipgram (gensim)	64.83	78.54	80.91	81.09	81.89	85.99	84.49	85.23	84.58	86.78	89.05	88.17	92.48
Word2Vec+CBOW (tensorflow)	54.72	73.35	75.66	71.26	78.59	80.46	82.33	81.16	83.45	85.90	87.43	89.99	89.93
Word2Vec+Skipgram (tensorflow)	56.13	77.91	78.52	74.89	79.94	80.09	84.06	83.34	84.65	86.19	89.72	87.70	92.30
Word2Vec+CBOW+ Skipgram (tensorflow)	54.97	77.18	76.99	74.57	78.92	79.99	84.14	82.73	81.94	86.31	90.18	90.02	91.61
FastText+CBOW	65.82	76.64	80.61	81.11	81.86	85.21	86.49	84.45	85.73	89.90	88.15	91.15	92.35
FastText+Skipgram	75.85	76.78	81.27	**81.58**	82.34	85.29	87.95	88.21	89.55	90.31	92.19	90.95	93.54
FastText+CBOW+ Skipgram	75.83	76.63	80.97	81.61	81.93	85.23	85.69	87.89	90.14	90.78	91.44	91.08	93.25
GloVe+CBOW	67.23	75.19	78.47	79.63	80.96	85.35	84.14	86.51	88.90	86.05	87.68	89.96	91.55
GloVe+Skipgram	73.89	75.82	80.94	80.38	81.53	85.67	85.19	88.98	84.93	87.34	89.95	90.42	92.33
GloVe+CBOW+ Skipgram	76.12	76.07	79.57	80.44	81.29	85.12	85.32	87.93	85.66	86.26	89.92	90.49	92.28
Bangla-BERT+CBOW	80.13	83.16	86.91	78.92	92.14	91.03	92.34	90.68	92.89	94.13	94.45	93.44	94.47
skipBangla-BERT	82.15	83.17	87.72	80.14	**92.37**	91.09	**92.55**	**91.87**	93.03	**94.37**	94.66	94.26	**95.71**
Bangla-BERT+CBOW+ Skipgram	81.99	**83.21**	87.79	79.54	92.29	91.14	92.47	91.74	**93.14**	94.26	**94.68**	**94.55**	95.27

Table 27
Best performance measurements of different domains using (skipBangla-BERT) based on 15 categories.

Domain	Best model	Precision	Recall	F1-score	Accuracy (%)
ML	SVM	0.81	0.86	0.83	84.88
EL	RF	0.78	0.91	0.84	85.34
DL	CNN-BiLSTM	**0.86**	**0.93**	**0.89**	**90.24**

Table 28
Best performance measurements of different domains using skipBangla-BERT based on 3 categories.

Domain	Best model	Precision	Recall	F1-score	Accuracy (%)
ML	SVM	0.96	0.91	0.93	92.37
EL	XGB	0.89	**0.99**	0.94	92.55
DL	CNN-BiLSTM	**0.97**	0.94	**0.95**	**95.71**

BOW metrics. Word2Vec along with gensim and tensorflow libraries show an accuracy of 65.73% and 62.81% respectively. In our experiment, we have found that Word2Vec performed worst compared to the other feature metrics.

The comparison among existing recent works and our proposed work is illustrated in Table [?] where we have measured the dataset used, no. of categories, feature metric, used model, f1-score and accuracy as comparative features. In 2022 (Bhowmik et al., 2022) a method was proposed for Bangla sentiment analysis using extended lexicon dictionary and deep learning algorithms. They used two aspect based datasets on Cricket and Restaurant and obtained highest accuracy of 84.18% using the hybrid method BERT-LSTM. Another method from 2022 (Prottasha et al., 2022) used six datasets and obtained highest f1-score and accuracy with CNN-BiLSTM model of 0.93 and 94.15%. The very recent work (Kabir et al., 2023) used a moderately large dataset of 158,065 instances and achieved highest f1-score 0.93 using BERT model. Another new work (Bitto et al., 2023) used a dataset of only 1400 reviews from food delivery startup and achieved an accuracy of 91.07% and f1-score of 0.85. Our proposed dataset contain more instances than the compared 4 other existing recent works. To the best of our knowledge, we have created one of the largest document-level Bangla SA corpus of 203,463 comments from social media. Most of the work on Bangla sentiment analysis consider only 3 basic sentiment categories (positive, negative and neutral). But in this work we have examined sentiment analysis on both 15 categories and 3 categories and noticed that accuracy (*i.e.* results) and no. of categories are inversely proportional to each other. Moreover, the proposed work also discover a new hybrid feature extraction method (skipBangla-BERT) for Bangla textual data which outperforms 20 other hybrid methods. We have implemented 13 different algorithms from 3 different domains from ML, EL and DL, among them the hybrid method (CNN-BiLSTM) from DL domain outperforms the others. The best achieved accuracy is 90.24% in 15 categories and 95.71% in 3 categories. Another graphical performance metric that is very useful for machine learning algorithms is the receiver operating characteristic (ROC) curve (Bradley, 1997). We have shown the ROC curves for both the 15 and 3 categories in Figs. 20 and 21 respectively. Individual area under curve (AUC) values are also given in the ROC curves to observe the significance of each category (see Table 29).

Table 29
Comparison among existing works and our proposed work.

Name	Year	Dataset used	No. of categories	Feature metric	Best model	F1-Score	Accuracy (%)
Bhowmik et al. (2022)	2022	2900 & 2600	3	Word2Vec	BERT-LSTM	N/A	84.18
Prottasha et al. (2022)	2022	2900 & 2600 & 4 others	3	BERT	CNN-BiLSTM	0.93	94.15
Kabir et al. (2023)	2023	(BanglaBook) 158,065	3	BOW+ N-Gram	BERT	0.93	N/A
Bitto et al. (2023)	2023	1400	2	Word2Vec	LSTM	0.85	91.07
Islam and Alam (2023b)	2023	44,491	3	TF-IDF	CNN-BiGRU	0.90	90.96
Mia et al. (2024)	2024	6500	3	TF-IDF, GloVe	BERT	0.65	65.00
Our Proposed (SA_Bangla)	**2024**	**(BangDSA) 203,463**	**3**	**skipBangla-BERT**	**CNN-BiLSTM**	**0.96**	**95.71**
			15	**skipBangla-BERT**	**CNN-BiLSTM**	**0.91**	**90.24**

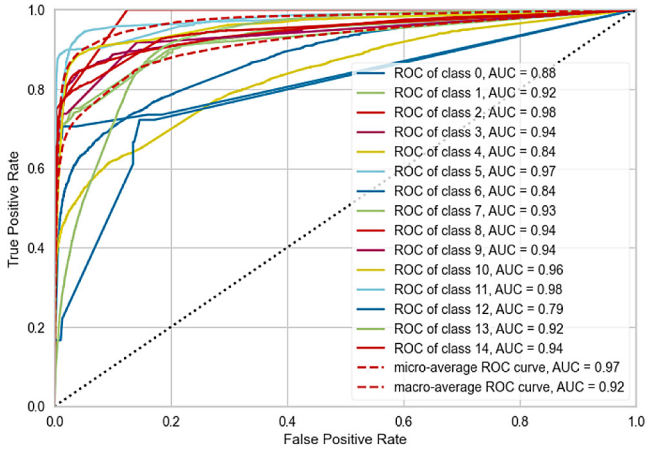

Fig. 20. ROC curve of CNN-BiLSTM model based on 15 categories.

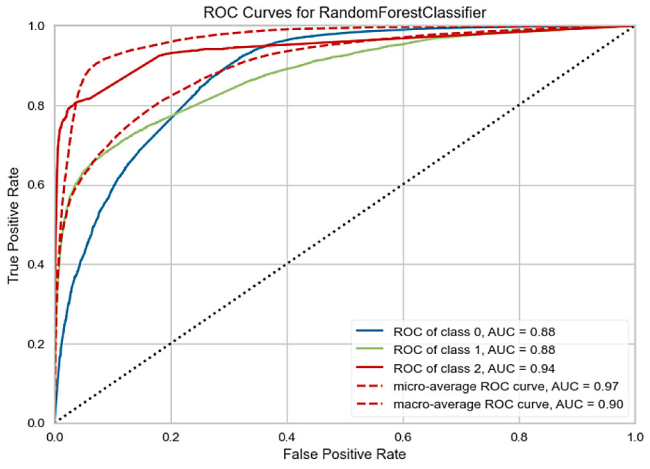

Fig. 21. ROC curve of CNN-BiLSTM model based on 3 categories.

4.4. Statistical test (friedman test)

To observe the statistical significance of the obtained results for both 15 and 3 categories, we have examined the non-parametric statistical test called Friedman test (Liu and Xu, 2022) on the results obtained for the best performed models from ML, EL and DL domains. In case of 15 categories SVM, RF and CNN-BiLSTM are the best models from ML, EL and DL domains (see Table 25). In a similar fashion, for the 3 categories SVM, XGB and CNN-BiLSTM are the best models from ML, EL and DL domains (see Table 26). The obtained accuracy scores for these 3 models using different features are taken to run the Friedman test. The environment of the test: the level of significance is 0.05. An open source online statistical calculator[3] is used to run the test. The calculation summary of the $\chi_r 2$ statistic is shown in Figs. 22 and 23 for the 15 and 3 categories respectively.

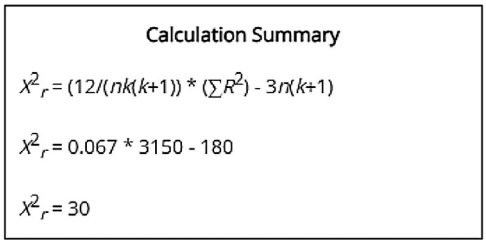

Fig. 22. Friedman test based on 15 categories.

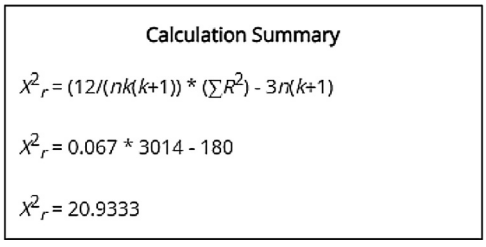

Fig. 23. Friedman test based on 3 categories.

The obtained results from the test based on 15 categories, Friedman $\chi_r 2$ statistic is 31.05, the degrees of freedom $(df) = 2$, p-value < 0.00001. The result is significant based on 15 categories at $p < 0.05$. The obtained results from the test based on 3 categories, Friedman $\chi_r 2$ statistic is 20.9333, the degrees of freedom $(df) = 2$, p-value is 0.00003. The result is significant based on 3 categories at $p < 0.05$. So, both the test results are significant at $p < 0.05$.

4.5. Limitation of the proposed work

Though the proposed dataset (BangDSA) is one of the largest document level sentiment analysis datasets, but it is not a balanced one.

[3] https://www.socscistatistics.com/tests/friedman/default.aspx

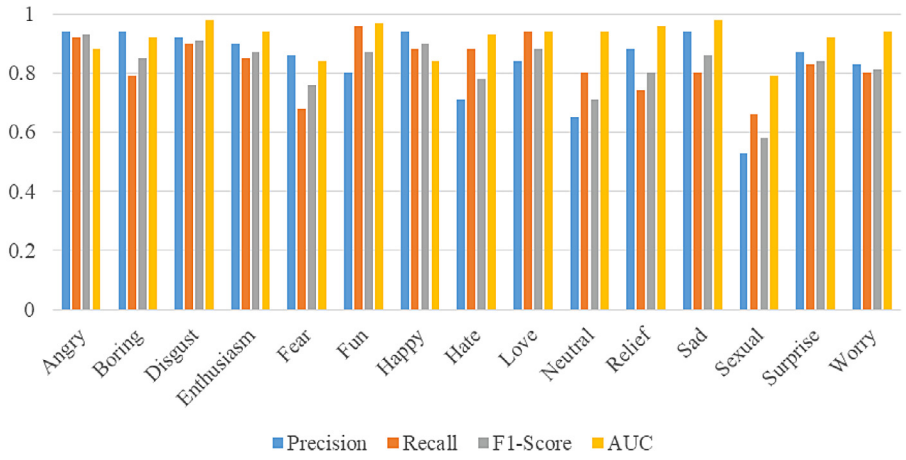

Fig. 24. Performance of 15 sentiment categories using skipBangla-BERT.

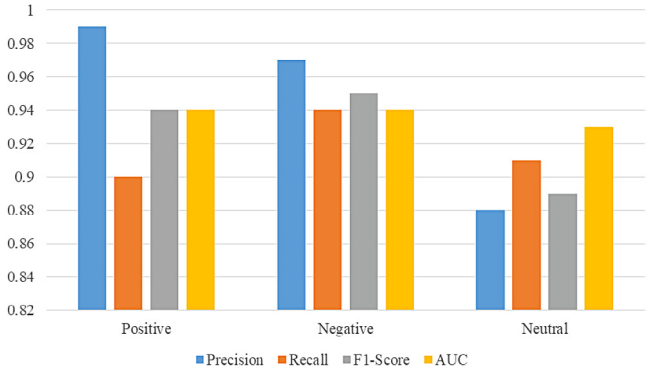

Fig. 25. Performance of 3 sentiment categories using skipBangla-BERT.

Some categories such as the *surprise*, *relief*, *worry*, *fear*, and *hate* contain very less instances compared to the other categories which can cause the overfitting situations. The performance comparison among the 15 and 3 sentiment categories using skipBangla-BERT are given in Figs. 24 and 25 respectively. The *Fear*, *Hate* and *Neutral* categories did not perform well compared to the other sentiment categories, as these categories have less number of samples in training. We expect that by increasing the training samples of those categories the performance can be increased more. Though feature importance understanding is essential for the digital development of a language. But in this work, we did not consider any explainable artificial intelligence (AI) techniques. The fundamental problem of AI and learning-related tasks is the lack of interpretability of a model being performing well or poorly, which features are responsible for classifying a document into a certain category, which are the top most important features, and so on.

5. Conclusion

In this modern technologically advanced world, Sentiment Analysis (SA) is a very important topic in every language due to its various trendy applications. But it is a matter of regret that a few research works have been conducted in Bangla language on SA compared to other languages due to lack of publicly available datasets and resources. The principle objective of this work is to develop a new comprehensive dataset for Bangla SA, then search for a better hybrid feature metric that can assist different learning models to make efficient predictions, and finally find one or more efficient learning models from either machine learning, ensemble learning or deep learning techniques. The proposed dataset contain 203,493 Bangla comments collected from 5 microblogging sites. The comments are annotated into 15 predefined sentiment categories by 5 native annotators and we have validated the annotation process by different 40 native Bangla speakers with a validation accuracy of 94.67%. The proposed dataset approximately follows the Zipf's law, covering 32.84% function words with a vocabulary growth rate of 0.053, tagged both on 15 and 3 categories. So, it can be said that it is a good dataset indeed. The proposed work also focuses on examining different hybrid feature extraction techniques. TF-IDF-ICF outperforms the traditional TF-IDF and BOW method, 3-gram slightly performs better than 2-gram. Among the word embedding models FastText performs better than Word2Vec and GloVe, Skipgram surpasses the traditional CBOW mechanism. This work examined 21 different hybrid feature extraction techniques and found the novel method (skipBangla-BERT) which outperforms all other techniques. For the classification task, we have implemented 5 ML algorithms; 3 EL algorithms; 4 DL algorithms and a hybrid DL method CNN-BiLSTM. Among the implemented algorithms, KNN obtained the worst performance as it is a lazy learner, KNN achieved highest 72.69% and 81.58% accuracy for 15 and 3 categories respectively; rest of the other classifiers are all eager learners and they require training phase. The hybrid method CNN-BiLSTM along with feature metric skipBangla-BERT produced the best results in both 15 and 3 categories versions of the dataset. The best acquired accuracy for the CNN-BiLSTM model is 90.24% in 15 categories and 95.71% in 3 categories. It is noticed by experiments that the number of categories and model's performance is inversely proportional. A statistical test (Friedman test) was performed on the obtained results to observe the statistical significance with 0.05 level of significance. The statistical test shows that the obtained results are significant in both 15 and 3 categories. In the future, we want to enrich and balance our dataset more and convert it to a representative one for Bangla SA and explore evolutionary algorithms for extracting features from texts.

CRediT authorship contribution statement

Md. Shymon Islam: Formal analysis, Investigation, Methodology, Visualization, Writing – original draft, Resources, Software, Validation. **Kazi Masudul Alam:** Conceptualization, Supervision.

Declaration of competing interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Authors would like to thank the Computer Science and Engineering Discipline, Khulna University, Bangladesh for providing resources and

time for the design and implementation of this research work and also thank to some students from the Department of Computer Science and Engineering, North Western University, Bangladesh for being involved in the dataset collection and annotation process of this research work.

References

Akther, A., Islam, M.S., Sultana, H., Rahman, A.R., Saha, S., Alam, K.M., Debnath, R., 2022. Compilation, analysis and application of a comprehensive bangla corpus kumono. IEEE Access 10, 79999–80014. http://dx.doi.org/10.1109/ACCESS.2022.3195236.

Alam, M.H., Rahoman, M.M., Azad, M.A.K., 2017. Sentiment analysis for bangla sentences using convolutional neural network. In: 20th International Conference of Computer and Information Technology. (ICCIT), pp. 1–6. http://dx.doi.org/10.1109/ICCITECHN.2017.8281840.

Alvi, N., Talukder, K.H., Uddin, A.H., 2022. Sentiment analysis of bangla text using gated recurrent neural network. In: International Conference on Innovative Computing and Communications Advances in Intelligent Systems and Computing, vol. 1388, pp. 77–86. http://dx.doi.org/10.1007/978--981--16--2597--8_7.

Amin, A., Hossain, I., Akther, A., Alam, K.M., 2019. Bengali VADER: A sentiment analysis approach using modied VADER. In: International Conference on Electrical, Computer and Communication Engineering. (ECCE), pp. 1–6. http://dx.doi.org/10.1109/ECACE.2019.8679144.

Azmin, S., Dhar, K., 2019. Emotion detection from bangla text corpus using naive bayes classifier. In: 4th International Conference on Electrical Information and Communication Technology. (EICT), pp. 1–6. http://dx.doi.org/10.1109/EICT48899.2019.9068797.

Bhattacharjee, A., Hasan, T., Ahmad, W., Mubasshir, K.S., Islam, M.S., Iqbal, A., Rahman, M.S., Shahriyar, R., 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. In: Findings of the Association for Computational Linguistics: NAACL 2022. pp. 1318–1327. http://dx.doi.org/10.18653/v1/2022.findings--naacl.98.

Bhowmik, N.R., Arifuzzaman, M., Mondal, M.R.H., 2022. Sentiment analysis on bangla text using extended lexicon dictionary and deep learning algorithms. Array 3, 100123. http://dx.doi.org/10.1016/j.array.2021.100.

Bitto, A.K., Bijoy, M.H.I., Arman, M.S., Mahmud, I., Das, A., Majumder, J., 2023. Sentiment analysis from Bangladeshi food delivery startup based on user reviews using machine learning and deep learning. Bull. Electr. Eng. Inform. 12, 2282–2291. http://dx.doi.org/10.11591/eei.v12i4.4135.

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. 30 (7), 1145–1159. http://dx.doi.org/10.1016/S0031--3203(96)00142--2.

Camacho-Collados, J., Pilehvar, M.T., 2018. From word to sense embeddings: a survey on vector representations of meaning. J. Artificial Intelligence Res. 63 (1), 743–788. http://dx.doi.org/10.1613/jair.1.11259.

Cerqueira, T., Ribeiro, F.M., Pinto, V.H., Lima, J., Gonçalves, G., Glove prototype for feature extraction applied to learning by demonstration purposes. Appl. Sci. 12 (21), 2076–3417. http://dx.doi.org/10.3390/app122110752.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. J. Artificial Intelligence Res. 16, 321–357. http://dx.doi.org/10.1613/jair.953.

Chowdhury, S., Chowdhury, W., 2014. Performing sentiment analysis in bangla microblog posts. In: IEEE International Conference on Informatics, Electronics & Vision. (ICIEV), pp. 1–6. http://dx.doi.org/10.1109/ICIEV.2014.6850712.

Dash, N.S., 2005. Corpus Linguistics and Language Technology: With Reference to Indian Languages. Mittal Publications, New Delhi, India.

Habibullah, M., Islam, M.S., Jahura, F.T., Biswas, J., 2023. Bangla document classification based on machine learning and explainable NLP. In: 6th International Conference on Electrical Information and Communication Technology. (EICT), pp. 1–6. http://dx.doi.org/10.1109/EICT61409.2023.10427766.

Hassan, A., Amin, M.R., Azad, A.K.A., Mohammed, N., 2016. Sentiment analysis on bangla and romanized bangla text using deep recurrent models. In: International Workshop on Computational Intelligence. (IWCI), pp. 51–56. http://dx.doi.org/10.1109/IWCI.2016.7860338.

Hassan, M., Shakil, S., Moon, N.N., Islam, M.M., Hossain, R.A., Mariam, A., Nur, F.N., 2022. Sentiment analysis on bangla conversation using machine learning approach. Int. J. Electr. Comput. Eng. (IJECE) 12, 5562–5572. http://dx.doi.org/10.11591/ijece.v12i5.pp5562--5572.

Islam, M.S., Alam, K.M., 2023a. An empiric study on bangla sentiment analysis using hybrid feature extraction techniques. In: 14th International Conference on Computing Communication and Networking Technologies. (ICCCNT), pp. 1–7. http://dx.doi.org/10.1109/ICCCNT56998.2023.10308114.

Islam, M.S., Alam, K.M., 2023b. Sentiment analysis on bangla food reviews using machine learning and explainable NLP. In: 26th International Conference on Computer and Information Technology. (ICCIT), pp. 1–6. http://dx.doi.org/10.1109/ICCIT60459.2023.10441309.

Junaid, M.I.H., Hossain, F., Upal, U.S., Tameem, A., Kashim, A., Fahmin, A., 2022. Bangla food review sentimental analysis using machine learning. In: IEEE 12th Annual Computing and Communication Workshop and Conference. (CCWC), pp. 0347–0353. http://dx.doi.org/10.1109/CCWC54503.2022.9720761.

Kabir, M., Mahfuz, O.B., Raiyan, S.R., Mahmud, H., Hasan, M.K., 2023. BanglaBook: A large-scale bangla dataset for sentiment analysis from book reviews. Comput. Lang. http://dx.doi.org/10.48550/arXiv.2305.06595.

Liu, J., Xu, Y., 2022. T-friedman test: A new statistical test for multiple comparison with an adjustable conservativeness measure. Int. J. Comput. Intell. Syst. 15 (29), 29. http://dx.doi.org/10.1007/s44196--022--00083--8.

Manning, C., Schutze, H., 1999. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA, USA.

Mia, M., Das, P., Habib, A., 2024. Verse-based emotion analysis of bengali music from lyrics using machine learning and neural network classifiers. Int. J. Comput. Digital Syst. 15 (1), 359–370. http://dx.doi.org/10.12785/ijcds/150128.

Nafisa, N., Maisha, S.J., Masum, A.K.M., 2023. Document level comparative sentiment analysis of bangla news using deep learning-based approach LSTM and machine learning approaches. Appl. Intell. Ind. 4.0 198–211.

Prottasha, N.J., Sami, A.A., Kowsher, M., Murad, S.A., Bairagi, A.K., Masud, M., Baz, M., 2022. Transfer learning for sentiment analysis using BERT based supervised fine-tuning. Sensors 22, 4157. http://dx.doi.org/10.3390/s22114157.

Rafat, A.A.A., Salehin, M., Khan, F.R., Hossain, S.A., Abujar, S., 2019. Vector representation of bengali word using various word embedding model. In: 2019 8th International Conference System Modeling and Advancement in Research Trends. (SMART), pp. 27–30. http://dx.doi.org/10.1109/SMART46866.2019.9117386.

Rahman, F., 2019. An annotated bangla sentiment analysis corpus. In: International Conference on Bangla Speech and Language Processing. (ICBSLP), pp. 1–5. http://dx.doi.org/10.1109/ICBSLP47725.2019.201474.

Rahman, M.A., Dey, E.k., 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. Data 3, 15. http://dx.doi.org/10.3390/data3020015.

Rashid, M.R.A., Hasan, K.F., Hasan, R., Das, A., Sultana, M., Hasan, M., 2024. A comprehensive dataset for sentiment and emotion classification from Bangladesh e-commerce reviews. Data Brief 53, 110052. http://dx.doi.org/10.1016/j.dib.2024.110052.

Shafin, M.A., Hasan, M.M., Alam, M.R., Mithu, M.A., Nur, A.U., Faruk, M.O., 2020. Product review sentiment analysis by using NLP and machine learning in bangla language. In: 23rd International Conference on Computer and Information Technology. (ICCIT), pp. 1–5. http://dx.doi.org/10.1109/ICCIT51783.2020.9392733.

Sharmin, S., Chakma, D., 2021. Attention-based convolutional neural network for bangla sentiment analysis. AI Soc. 36, 381–396. http://dx.doi.org/10.1007/s00146--020--01011--0.

Sumit, S.H., Hossan, M.Z., Al Muntasir, T., Sourov, T., 2018. Exploring word embedding for bangla sentiment analysis. In: International Conference on Bangla Speech and Language Processing. pp. 1–5. http://dx.doi.org/10.1109/ICBSLP.2018.8554443.

Tabassum, N., Khan, M.I., 2019. Design an empirical framework for sentiment analysis from bangla text using machine learning. In: International Conference on Electrical, Computer and Communication Engineering. (ECCE), pp. 1–5. http://dx.doi.org/10.1109/ECACE.2019.8679347.

Tuhin, R.A., Paul, B.K., Nawrine, F., Akter, M., Das, A.K., 2019. An automated system of sentiment analysis from bangla text using supervised learning techniques. (ICCCS), pp. 360–364. http://dx.doi.org/10.1109/CCOMS.2019.8821658.

Wang, D., Zhang, H., 2010. Inverse-category-frequency based supervised term weighting scheme for text categorization. p. 15, arXiv preprint arXiv:1012.2609.