Emotion Recognition from Bengali Texts using Natural Language Processing

Md Moinul Islam[1]
moinul.islam@student.oulu.fi

Taufiq Ahmed[1]
taufiq.ahmed@student.oulu.fi

K H M Burhan Uddin[1]
burhan.uddin@student.oulu.fi

Efta Khairul Bashar[1]
efta.bashar@student.oulu.fi

[1]Department of Computer Science and Engineering
[1]University of Oulu

Abstract

Emotion is an abstract concept that varies from person to person depending on context and mode of expression, however, there is a pattern in detecting emotions by classifying analogous events. Considering that most individuals now prefer text messaging to other forms of communication, understanding emotion from a text is currently of utmost importance. There has been a lot of work done recently to identify emotions in English or other widely spoken languages. However, very little research has been done to date on emotion detection using Bangla texts. We were constrained in our ability to construct the model since there was no benchmark corpus or state-of-the-art model available for Bangla texts. However, to work on this project, we worked with the only publicly available emotion dataset for Bangla texts consisting six classes - happy, sad, anger, fear, disgust and surprise. We then tokenized the data, extracted the relevant features using different embedding techniques and applied different machine learning and deep learning techniques to classify the texts into six types of emotions. The performances of the models are later evaluated and comparative results are analyzed.

1 Introduction

Text is a thriving source of knowledge, but it can be a bit challenging and time-consuming to get insight because it is unstructured. Emotion classification from text has been more prominent in recent years due to the increased usage of emotions in marketing, political science, psychology, human-computer interaction, and artificial intelligence. In text classification, very much anything may be arranged, structured and categorized. New articles, for example, may be ordered by subject and tickets for assistance can be handled urgently, chat chats by language can be arranged, etc. This includes an independent model for text analysis and one of the most active fields in natural language processing (NLP), data mining and text mining [32]. The rise of social micro-blogging sites, news portals, online markets and other applications produces a very large volume of data every day in text format. This type of text categorization, therefore, plays a key role in the rapid use of huge amounts of text data.

Text emotion classification includes automatic assignment from a specified list of categories. Due to the growing number of virtual platform users and the quick production of online material, reading emotions in online information is crucial. This applies to customers, corporations, experts and other individuals. There are six basic emotions according to Ekman, which are happiness, sadness, fear, anger, surprise, and disgust [10]. These emotions are mostly identifiable by facial expressions. Those feelings can also be extracted from texts.

Online data and computational tools have accelerated emotion classification research in languages with extensive resources, such as English, Arabic, Chinese, and French. Bengali is understudied in emotion detection, despite substantial research and solid models in commonly spoken languages. Bengali textual data has grown due to extensive Internet and digital technology use. Bengali language processing research challenges include extracting emotions from massive data sets. Lack of tools, benchmark corpora, difficult language structures and limited resources generate complexity.

This research proposal outlines a thorough approach and a set of objectives with the purpose of developing a model for classifying emotions in Bengali text data. The models were developed by utilizing a publicly accessible dataset. The primary objectives of this research are as follows:

1. We developed a robust emotion detection model for Bengali texts-based data, capable of classifying Bengali texts into six different emotional categories.
2. We enhanced the performance metrics and compared them by evaluating the publicly available dataset along with different feature engineering and embedding techniques.
3. We evaluated the model's effectiveness in classifying the emotions from the unseen data and assessed its performance on them.
4. We investigated the performance of different machine learning and deep learning techniques on the Bangla text-based corpus for emotion classification.

The rest of this research proposal is organized as follows: Section 2 discusses the recent works in the current domain. Section 3 represents an overview of the dataset, the methodology and the performance evaluation metrics. Section 4 summarizes our research plan. Finally, Section 5 provides the contributions of each of the team members involved.

2 Related Works

This section briefly describes the previous works on text classification and emotion detection and learns about different methods helpful in classifying emotions. Basic lexical analysis may count word and word frequencies to perform basic operations, such as trying to categorize articles by subject. However, the semantics contained in a particular document could not be understood. The analytical process is carried on via data mining and by extension, text data mining. Data mining [1, 33] searches for hidden connections and other complicated data sets patterns i.e, classification, grouping, link-analysis, decision-making etc. Researchers also adopted or created statistical methods that, in conjunction with machine learning Algorithms and more profound linguistics, provide text functionalities such as semantic disambiguation. Many studies in categorizing text, for example spam classification [24, 6], automated categorization [30, 31], detection of suspicious profiles [2, 34], resume classification and keywords extraction [22, 15, 20] etc. have been carried out by utilizing text classification. In addition, a number of well-known text classification problems techniques and algorithms already exist, such as Naive Bayes Classifier, Support Vector Machine (SVM), Logistics Regression, Neural Networks [29, 16, 17, 21] etc.

Despite the limited availability of previous works in emotion detection from Bangla text, we have explored a few related research endeavors. In [3], the authors used Support Vector Machine as a classification method and three categories of variables were examined based upon, (i) Keyword and Micro-blogging Characteristics, (ii) Semantic Lexicons and (iii) Distributional Semantic Model (DSM). In [13], Long Short Term Memory (LSTM) which is a recurrent model based on deep learning process was used for analysis and for the construction of the Bangla lexicon, SentiWordNet has been employed by the authors. In [8], the authors proposed an emotional analysis of the words and sentences of Bengali blogs and news which employs the classification of Conditional Random Field (CRF) and the construction of an emotional corpus and lexicon was discussed. In [9], the author offers a comprehensive overview of the recent advancements in textual emotion recognition (TER) using deep neural networks. The survey categorizes these advancements into three key areas, word embedding with semantic and emotional knowledge, integration of emotional knowledge into deep learning architectures and effective training strategies for TER models. The authors also identified challenges such as the need for large-scale emotional datasets, handling fuzzy emotional boundaries and supplementing incomplete emotional information in texts. They proposed potential research directions, including data integration, multi-label emotions and the exploration of multi-modal interactions to enhance the field of TER and provide valuable insights for future research in this domain.

In another research [25], the authors investigate the application of attention networks in the field of emotion recognition during dyadic conversations. They introduced a novel attention mechanism that assesses the impact of past utterances on the current speaker's emotional state without requiring a decoder network, using innovative self-attention techniques. Their approach effectively modeled long-term temporal dynamics and achieved superior performance on the IEMOCAP benchmark compared to existing methods, suggesting promising avenues for research in Online Social Network (OSN) analysis tasks. In [5], the authors described sentiment analysis utilizing a six-monitor bootstrapping technique and a rule-based classification using two separate algorithms, SVM and Maximum Entropy in Bangla micro-blog postings. They reached a precision respectively of 93% and 85%. In [23], the authors presented a technique utilizing the search model in the field of the text mining process, TF-IDF (Term frequency-inversion document frequency). In [26], the authors utilized the Mutual Information (MI) for the process of functional selection and also for the classification procedure, Multinomial Naive Bayes (MNB). For the Bangla training dataset with negation, they obtained 87.79 percent accuracy.

To predict emotions from Bangla texts, the authors in [7] used machine learning, deep neural network and transformer-based algorithms using a Bangla text corpus consisting of more than 6,000 text lines with just six basic feelings. According to the results, the XLM-R outperforms all other models with an accuracy of 69.73% in the f1-score. The shortcomings of this paper are the variety of Bangla language processing methods and the lack of sufficient data in Bengali corpora to accurately predict & correctly classify emotions. The authors in [28] curated a Bangla emotion corpus from Facebook comments on socio-economic and political issues, aiming to extract six basic emotions and conducted experiments using five traditional machine learning models, Naive Bayes, Decision Tree, k-Nearest Neighbors, Support Vector Machine (SVM) and K-Means Clustering. SVM with a non-linear RBF kernel emerged as the top-performing model, achieving an accuracy of 52.98%, a significant improvement over the baseline, yet it still doesn't perform better in classifying emotions from unseen texts.

3 Methodology

The primary goal of our research is to develop a model capable of classifying suspicious and non-suspicious texts and identifying emotions from six distinct categories within them. Figure 1 shows an abstract view of our methodology.

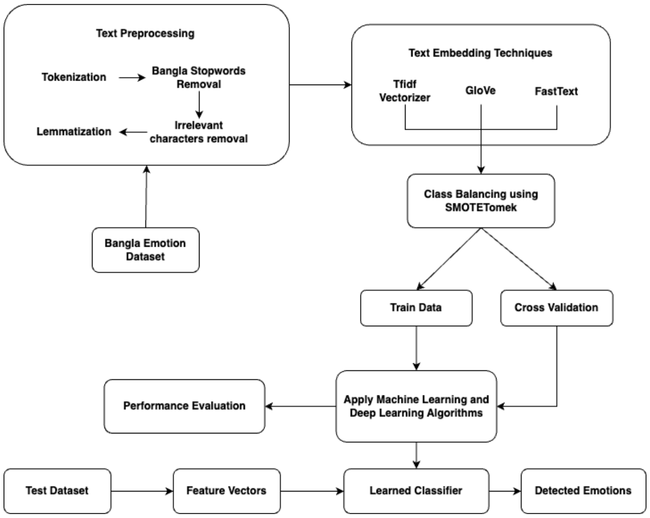

Figure 1: Proposed Model for Emotion Detection from Bangla Text

3.1 Dataset

Due to the benchmark corpus's unavailability for Bengali texts, we intend to use an existing Bengali text corpus, BEmoC [14], to classify the emotions. The BEmoC dataset consists of a total of 6700 text documents that have undergone preprocessing and annotation. The data-gathering methodology employed by the crawlers was characterized by selectivity, focusing on capturing texts that provide evidence in favor of Ekman's six emotions. Initially, the labeling was determined by the majority voting technique. To mitigate the potential influence of annotation bias, the experts and annotators engaged in thorough discussions and reached a consensus on the appropriate labels to be assigned. The assessment of inter-annotator agreement involved the utilization of coding reliability and Cohen's kappa scores to ensure the quality of the annotations. The inter-coder reliability of 93.1% and Cohen's Kappa coefficient of 0.91 demonstrate the high quality of the corpus. According to the data, Facebook accounted for the highest number of Bengali texts. This was followed by YouTube, blogs and online news portals.

3.2 Data Statistics

This dataset [14] contains 6243 text documents after the preprocessing. The number of instances in each class is represented in Table 1 and Table 2 displays the statistics of the training set.

Classes	Training Set	Validation Set	Testing Set
Joy	908	120	114
Fear	700	89	83
Disgust	1233	155	165
Sadness	942	129	119
Anger	621	67	71
Surprise	590	64	73

Table 1: Amount of instances in each class [14]

Class	Total words	Unique Words	Average Words per text
Joy	20885	7346	23.40
Fear	14766	5072	21.09
Disgust	27192	7212	22.35
Sadness	22727	7398	24.13
Anger	14914	5852	24.02
Surprise	13833	5675	23.45

Table 2: Statistics for each class [14]

3.3 Data Preprocessing

Text preprocessing is an important stage in natural language processing. In order to fill in missing values, smooth noisy data, to detect or eliminate the outliers and to resolve discrep- ancies, data pre-processing is necessary. All texts are prepared by the preprocessor in our model. Our emotional classifier model was trained using the core body of the text. A list of terms and their frequencies have been represented in our text document. We have utilized a stop words list which includes words that don't categorize text. These words are eliminated by matching the stop word list in our preprocessing section. It was helpful to enhance the efficiency of the model, since redundancy reduces the functionality of our model and raises the model's computational complexity. Text pre-processing includes removing words and symbols from feature space, and lowering noise and dimension. This helps classifier models learn important features and improve performance. In the domain of Natural Language Processing (NLP), the quality of data profoundly influences the efficacy of resultant models. Preprocessing is paramount, especially for datasets like the Bengali emotion set, which is derived from the bustling world of social media comments. Such comments often encapsulate the eclectic nature of online interactions, replete with emojis, links, and even embedded HTML.

Given this backdrop, cleaning and structuring the data transcends mere standard practice. It involves meticulously removing platform-specific elements like emojis, links, and HTML tags. By doing so, we ensure that our models are primed to identify and learn from relevant patterns, unobstructed by the noise characteristic of social media texts.

1. Tokenization will be used to distinguish words in the document set separated by white spaces or new lines and present them as tokens for processing.
2. Removal of special symbols, emoticons or digits (e.g., [0-9, a-z, @,!,#,$,%,&,..,ff,|,:,],+) from data collection will be done since they do not help categorize text.

3. Removing stop words helps classifier models capture key traits as most feature extraction methods favor word frequency. A conventional Bangla stop-word list will be used to remove them from the dataset.
4. Noise, in the context of text data, refers to any piece of information that doesn't contribute meaningfully to the content's essence or may lead the model astray during training.
5. Text data, especially when scraped from web pages, often contains residual HTML tags. These tags, which are instrumental for web browsers, are mere noise for NLP tasks and hence are stripped away.
6. URLs, while crucial for web navigation, don't typically offer semantic value for text analytics. Their diverse and inconsistent structures can mislead models. Thus, removing them ensures that models aren't perplexed by these irrelevant strings.
7. Emojis, though expressive, are graphical symbols that can introduce variability into the dataset. Their removal simplifies the text, paving the way for a more consistent dataset.

3.4 Text Normalization

Initially, once the texts were denoised, the next step involves structuring it and tokenization is the primary step in that direction. BLTK (Bangla Language Toolkit) offers specialized tools for Bangla text and its tokenizer efficiently breaks down text into individual words or tokens. This granularity is essential for models to process and understand text. Bangla, with its rich morphological structure and diverse vocabulary, presents unique challenges. Tokenization allows models to capture the essence of the language at a granular level, helping in better representation and understanding. By removing stopwords, the dataset becomes more concise, allowing models to focus on words that carry richer semantic value. This can lead to improved model efficiency and performance. Given BLTK's specialization in Bangla, it offers a curated list of stopwords tailored for Bangla text. This ensures that the removal process is both comprehensive and nuanced. Lemmatization is the process of reducing inflected words to their word stem or root form. It ensures that words derived from the same root are treated as equivalents, making data more consistent for modeling. Bangla, with its rich morphological variations, can benefit immensely from lemmatization. The Bangla lemmatizer, by reducing words to their root forms, ensures that the dataset remains dense with information while being devoid of unnecessary variations.

3.5 Feature Extraction

Due to the great number of unique words and phrases, text data can be extremely dimensional. Feature extraction reduces dimensionality by translating text data into manageable characteristics and improves machine learning models. Prominent techniques for feature extraction in natural language processing (NLP) encompass TF-IDF vectorizer and other word embedding techniques such as GloVe, FastText etc. We have applied such techniques and compared the performance for different strategies.

1) TF-IDF: The TF-IDF [19] is the product of two statistics, frequency of terms and frequency of reverse documents. The exact values of these statistics can be determined in numerous methods. TF-IDF is used for extracting features in the documents containing Bangla texts to utilize the Term Frequency (t, d), the raw number of a term in a document, i.e, the number of times the term, t takes place in document, d and Inverse Document Frequency is a measure of whether a phrase in all documents is frequent or rare. N is the total number of documents in the corpus and $(d\epsilon D : t\epsilon D)$ is the number of documents where term, t appears. Final weighting scheme of $tf - idf$ is,

$$tf - idf(t, d) = (0.5 + 0.5 * \frac{f_{t',d}}{max_{t'\epsilon d} f_{t',d}}) * log\frac{N}{n_t} \tag{1}$$

2) FastText Embedding: FastText [18] is an open-source, free, lightweight library that allows users to learn text representations and perform text classification tasks efficiently. It was developed by Facebook's AI Research (FAIR). FastText is known for its ability to generate word embeddings (dense vector representations) for words and subwords, and it can be used for various natural language processing (NLP) tasks, including text-based emotion classification. The first step is to train word embeddings on the text data. FastText can learn word representations based on the context in which words appear. This involves learning vector representations for words and subwords (character-level n-grams). Based on the dataset, we used the trained FastText model to convert the text data into word embeddings. For each text sample, we can average the word embeddings of the words in the text to obtain a fixed-length vector representation. After training the model, we evaluate its performance on a validation set. We used standard evaluation metrics like accuracy, precision, recall and F1-score to measure the classification performance. Once the model was trained and evaluated, we used it to classify the emotions of new text samples.

3) GloVe Embedding: GloVe [27] (Global Vectors for Word Representation) is another popular word embedding method, similar to FastText. It can be used in emotion text classification tasks to capture semantic relationships between words and improve the performance of text classification models. We started by obtaining pre-trained GloVe word embeddings. GloVe provides pre-trained embeddings trained on large text corpora. We downloaded these pre-trained embeddings from the GloVe website or used Python libraries like SpaCy or Gensim to load them into the model. We created an embedding layer in the machine learning and neural network architectures to map words to their corresponding GloVe embeddings. This layer was used to convert text data into vector representations. Using pre-trained GloVe embeddings helped capture the semantic relationships between words and improve the model's ability to understand the emotional context within the text. The choice of architecture and hyperparameters depends on the specific characteristics of the dataset and the nature of the emotion classification task.

There are pre-trained vectors for the Bengali language available for both GloVe and FastText that have been trained using generalized Bengali wiki dump data. We trained both embedding techniques with window size 5 and the embedding dimension was the length of the total feature vectors. We discovered that deep learning models perform well on pre-trained vectors using the dataset.

3.6 Class Balancing Technique

In the Bengali emotion dataset, a notable variation in the distribution of emotion classes is observed, reflecting a potential data imbalance issue. While "disgust" is represented by 24.69% of samples, emotions like "surprise" and "anger" comprise only 11.81% and 12.43% respectively. Such disparities can make machine learning models lean towards predicting emotions with higher representation, like "disgust", thereby compromising their efficacy in correctly classifying lesser-represented emotions such as "surprise" or "anger". This skew can impede the model's real-world applicability, where a balanced recognition of all emotions is essential. Additionally, standard evaluation metrics may not truly reflect the model's performance across all classes, as a model might attain high scores largely by accurately predicting the predominant emotion, sidelining its potential inefficiencies with the minority emotions. We used a hybrid balancing technique, SMOTE-Tomek links in this research.

SMOTE-Tomek [4] is a hybrid resampling technique tailored to address data imbalances by merging the strengths of oversampling and class-cleaning mechanisms. Specifically, SMOTE augments the minority class by creating synthetic instances: for each minority sample, it alters its feature vector based on a random weighted difference with a neighbor, thereby producing a new synthetic sample. On the other hand, Tomek Links refines the data by identifying and removing instances from the majority class that are proximate to those of the minority, enhancing the clarity of decision boundaries. Opting for SMOTE-Tomek in this study is rooted in its multifaceted approach to data equilibrium. It not only rectifies the issue of under-representation through SMOTE but also purges potential noise and ambiguities with Tomek Links, bestowing the dataset with a balanced and purified structure. This approach not only mitigates the risk of overfitting, commonly associated with oversampling but also fosters heightened model sensitivity, ensuring that no emotion class, especially the minority ones, remains overshadowed during classification.

In this study, the training dataset, marked by imbalances in emotions like surprise, anger, and fear, was harmonized using the SMOTE-Tomek technique. Initially, under-represented emotions were pinpointed based on dataset distribution. Subsequently, SMOTE was employed to synthesize samples for these minority classes, striving for an optimal balance. This augmented dataset was then refined by detecting and excising Tomek links, sharpening the decision boundaries between emotion classes. After the SMOTE-Tomek application, a thorough review of the class distribution was conducted to confirm a balanced representation, thereby priming the data for efficient model training.

3.7 Machine Learning Techniques

After extracting features from the texts, these features are used to train our machine-learning model. Logistic Regression, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Decision Tree algorithms are used for training purpose. For classification, logistic regression is employed where the objective of classification is to forecast the target class. When it comes to logistic multinomial regression, we have applied logistic regression methods to predict the target class (total 06), e.g. disgust, anger or joy. The purpose of using the Support Vector Machine algorithm in our research is to locate an N-dimensional hyper-plane, where, N is the number of features extracted from the documents. It determines a limit among the classes, which maximizes the margin, i.e. the distance among the data in these classes. For each of the potential values of that input variable in a decision tree, an inner node corresponds to one of the input variables and children are provided with edges for each. Each leaf is the target variable value given the input variable values represented by the root to the leaf path.

3.8 Neural Networks

1) Multi-Layer Perceptron: Multi-Layer Perceptron [11] is a fundamental type of artificial neural network commonly used in machine learning and deep learning. It serves as the basis for more complex neural network architectures. An MLP consists of multiple layers of interconnected nodes or neurons, each layer typically being fully connected to the next layer. The input layer of an MLP contains neurons that represent the features or inputs to your model. Each neuron in this layer corresponds to a feature in the dataset. Each neuron in a hidden layer takes input from every neuron in the previous layer and produces an output that is passed to every neuron in the next layer. The presence of multiple hidden layers allows MLPs to learn complex, non-linear relationships within the data. The output layer produces the final result or prediction. Activation functions are applied to the outputs of neurons in the hidden layers. They introduce non-linearity into the model, allowing it to approximate complex functions. Training the MLP involves using an optimization algorithm to adjust the weights and biases based on the difference between the predicted outputs and the actual targets. For MLP, we used a 3-layer neural network with two dense layers. The first dense layer is connected to 128 neurons with Relu activation function and in the second layer we have 6 neurons as our output label with softmax activation. We used “Adam” optimizer with categorical cross-entropy loss function.

2) Convolutional Neural Networks: Convolutional neural networks, or CNNs [12], were inspired by biological phenomena, specifically by the organization of neurons in these networks, which is like that of the visual cortex in animals. Within this biological framework, a cortical neuron’s receptive field is the precise region of the visual field within which it reacts to inputs. The complete visual field is covered by the partial overlap of these individual neurons and receptive fields. Convolutional Neural Network (CNN) can conduct convolution operations on neighboring cells, it is a reliable option for processing data arranged in a grid pattern, such as a 15x15 grid. The approach yields a thinner and denser grid that successfully captures intercellular linkages that are concealed within the dimensions of the kernel. The vectorial representations produced by word embeddings for input tokens provide this intermediary layer with its matrix structure. Its dimensions allow us to use a 1D convolutional network with more precision. We chose the RELU activation function since it has a faster computing speed than the hyperbolic tangent function. We used a MaxPooling1D technique to downsample the values on top of the CNN layer, which decreased the number of parameters in the model and its computational complexity. To avoid overfitting in Convolutional Neural Networks (CNNs) and other deep learning models, dropout is a regularization approach that is frequently employed. When a model learns to perform extraordinarily well on training data but finds it difficult to generalize its performance to test or unseen data, this phenomenon is known as overfitting.

3) Bidirectional Long Short Term Memory: Recurrent Neural Networks (RNNs) are a unique category of neural architectures specifically crafted to process sequential data by maintaining a state or "memory" across sequences. However, while RNNs excel at handling short-term dependencies, they falter when tasked with longer sequences due to vanishing and exploding gradient issues. Addressing this, Long Short-Term Memory (LSTM) networks, a subtype of RNNs, employ specialized ’gates’ to control information flow, enhancing their ability to capture extended dependencies. Building on the foundational strengths of LSTMs, Bidirectional LSTMs (BiLSTMs) process input sequences from both ends, capturing past and future contextual information concurrently. This dual processing allows BiLSTMs to offer a comprehensive understanding of a word's context, which is pivotal for tasks like emotion recognition in text. Given the inherent need for understanding both preceding and succeeding word contexts in emotion classification, BiLSTMs were deemed the optimal choice for this research endeavor.

The foundation of our neural model is a sequential architecture, beginning with an embedding layer. This layer converts input sequences of word tokens into dense vectors of fixed size, which are trainable. The embeddings provide a compact way of representing words and capturing their semantic meanings. Following the embedding layer, the architecture incorporates bidirectional LSTM layers. As discussed, these layers help in capturing the temporal dependencies from both preceding and succeeding sequences, making the model adept at understanding context in a more holistic manner. To prevent the model from overfitting to the training data, dropout layers are interspersed. Dropout, a regularization technique, randomly sets a fraction of input units to 0 at each update during training, which helps ensure that the model generalizes well to unseen data. The model concludes with a dense layer with softmax activation. This layer produces probability distributions over the classes (in this case, emotions) and determines the most likely class for the input text. The model was trained using a categorical cross-entropy loss function, optimized by the Adam optimizer. Metrics, including accuracy, were used to gauge the performance of the model during training and validation.

3.9 Model Training & Performance Evaluation

1) Model Training: Based on the research plan, we implemented several machine learning and deep learning-based algorithms with their training parameters to classify emotions from texts. The performances of the models were evaluated according to the existing metrics such as accuracy, precision, recall and F1-score for each of the emotion classes. Cross-validation was also done to assess the model's robustness. Finally, a comparative analysis is provided along with different applied strategies in terms of classifying emotions from Bengali texts.

The deep learning models were trained for a predefined number of epochs, ensuring adequate learning while avoiding overfitting. An appropriate batch size was chosen to balance computational efficiency and gradient accuracy. During training, a portion of the dataset was reserved as validation data. This data wasn't involved in the training process but was used to evaluate the model's performance after each epoch. Monitoring performance on the validation set helped in the early detection of overfitting. To prevent the model from overfitting to the training data, dropout layers are interspersed. Dropout, a regularization technique, randomly sets a fraction of input units to 0 at each update during training, which helps ensure that the model generalizes well to unseen data. The models conclude with a dense layer with softmax activation. This layer produces probability distributions over the classes (in this case, emotions) and determines the most likely class for the input text. The models were trained using a categorical cross-entropy loss function, optimized by the Adam optimizer. Metrics, including accuracy, were used to gauge the performance of the model during training and validation. In this study, multiple strategies were employed to combat overfitting. As mentioned earlier, dropout layers were incorporated into the model architecture. Additionally, the performance on the validation set was continually monitored, and early stopping could be employed to halt training if the validation performance started deteriorating, ensuring a model that is both accurate and generalizable.

2) Performance Evaluation: To evaluate our proposed model, we used several evaluation metrics. The number of false positives, true positives, false negatives, and true negatives is reported in the confusion matrix. In our proposed model, if the number of texts are labeled and classified as happiness, it's true positive, TP and if not labeled and classified correctly, then it's true negative, TN. If the number of texts are labeled as happiness, but not classified correctly, then it's false negative, FN and if the no. of documents are not labeled correctly, but classified as happiness, it's false positive, FP. These metrics are used to calculate other evaluation measures such as, precision, recall, f1-score and accuracy.

The performance of traditional machine learning models using **TF-IDF** vectorization on text data as shown in Table 3 is as follows, Logistic Regression achieved the highest precision at 61.89%, indicating a good ability to make accurate positive predictions. It also demonstrated a balanced F1-score of 60.37% and an accuracy of 60%. SVM performed well with a precision of 63.43% and an accuracy of 62.72%, indicating accurate positive predictions and overall classification performance. Random Forest also displayed respectable precision at 58.61% and an accuracy of 57.92%. Decision Tree and KNN had lower precision, recall, F1 scores, and accuracy, with Decision Tree achieving

46.08% accuracy and KNN at 41.60%. In summary, Logistic Regression, SVM, and Random Forest exhibited relatively good performance, while Decision Tree and KNN lagged behind in terms of accuracy and precision. The choice of the most suitable model may depend on specific task requirements and considerations.

Methods	Precision	Recall	F1-score	Accuracy (%)
Logistic Regression	0.6346	0.6250	0.6281	62.5
Decision Tree	0.4890	0.4856	0.4868	48.56
Random Forest	0.5995	0.5946	0.5891	59.45
SVM	0.6545	0.6506	0.6459	65.06
KNN	0.4763	0.4679	0.4588	46.79

Table 3: Performace Analysis for Traditional ML Methods using TF-IDF

Using **FastText** embeddings for text data as shown in Table 4, Logistic Regression achieved a precision of 55.91%, indicating decent accuracy in making positive predictions. It also demonstrated an F1 score of 53.98% and an accuracy of 53.76%. Decision Tree had relatively lower precision, recall, F1 score, and accuracy, with the highest being recall at 36.64%. Random Forest displayed a precision of 53.91% and an accuracy of 51.68%, while the F1 score was comparatively lower at 48.22%. SVM showed a precision of 56.97%, an F1 score of 54.91%, and an accuracy of 54.56%, indicating relatively accurate positive predictions and good overall classification performance. KNN had moderate precision and recall, with an F1 score of 44.40% and an accuracy of 45.12%. In summary, using FastText embeddings, the models' performance varied, with SVM and Logistic Regression achieving relatively better results in terms of precision, recall, F1 score, and accuracy, while Decision Tree and KNN lagged behind in performance metrics. Random Forest exhibited intermediate results.

Methods	Precision	Recall	F1-score	Accuracy (%)
Logistic Regression	0.5591	0.5376	0.5398	53.76
Decision Tree	0.3649	0.3664	0.3645	36.64
Random Forest	0.5391	0.5168	0.4822	51.68
SVM	0.5797	0.5456	0.5491	54.56
KNN	0.4595	0.4512	0.4440	45.12

Table 4: Performace Analysis for Traditional ML Methods using FastText

Using **Glove** embeddings for text data as found in Table 5, Logistic Regression achieved a precision of 54.74%, indicating decent accuracy in making positive predictions. It also demonstrated an F1 score of 53.57% and an accuracy of approximately 53.37%. Decision Tree had relatively lower precision, recall, F1 score, and accuracy, with the highest being recall at 38.46%. Random Forest displayed a precision of 52.53% and an accuracy of approximately 51.12%, while the F1 score was comparatively lower at 48.17%. SVM showed a precision of 56.66%, an F1 score of 55.60%, and an accuracy of approximately 55.29%, indicating relatively accurate positive predictions and good overall classification performance. KNN had moderate precision and recall, with an F1 score of 40.20% and an accuracy of approximately 41.67%. In summary, when using Glove embeddings, the models' performance shows similar trends to the previous FastText results, with SVM and Logistic Regression achieving relatively better results in terms of precision, recall, F1 score, and accuracy, while Decision Tree and KNN have lower performance metrics. Random Forest exhibits intermediate results.

Methods	Precision	Recall	F1-score	Accuracy (%)
Logistic Regression	0.5474	0.5337	0.5357	53.36
Decision Tree	0.37	0.3846	0.3743	38.46
Random Forest	0.5253	0.5112	0.4817	51.12
SVM	0.5666	0.5529	0.5560	55.29
KNN	0.4238	0.4167	0.4020	41.67

Table 5: Performace Analysis for Traditional ML Methods using GloVe

For Multi-layer perceptron, **MLP**, we trained our model for 50 epochs and used test set for the validation. The training loss decreased rapidly until it reached near zero and stayed there after

25 epochs. On the contrary, the validation loss increased on every epoch. After 50 epochs we got approximately 51% accuracy on the test set while the training accuracy is over 95%. This is clearly a sign of overfitting which is understandable as we have a small dataset. The model displays varying precision, recall and F1-score for different emotions, with its best performance in recognizing "disgust" and "fear." However, it struggles to accurately predict "anger," "sadness," and "surprise." The overall accuracy is moderate at 51%, indicating room for improvement. While the model shows promise in recognizing specific emotions, enhancements in data quality, model architecture, or hyperparameter tuning are needed to enhance its overall effectiveness in capturing a wider range of emotions.

Convolutional Neural Network, **Conv1D** is the type of model used to process 1D sequences, such as text data. To extract hierarchical features from the input, it has three Conv1D layers with progressively smaller filter sizes, together with batch normalization, dropout, and max-pooling layers. The features are transformed into a 1D vector by the flattened layer and then processed by a dense layer with 1024 units and ReLU activation. When doing multi-class classification tasks, the last dense layer with a softmax activation is utilized. This architecture is designed to efficiently learn and categorize patterns inside 1D sequences, making it appropriate for applications like text classification. We trained this model for 50 epochs on the same train data for **TF-IDF**. The training accuracy rose steadily and reached over 95% after 20 epochs. The train loss also decreased in the same manner and nearly touched zero. However, the validation loss was quite fluctuating and mostly on the rise rather than falling. Finally, the accuracy of the test data was 0.5060 (approximately 51%). Again from the loss function, the overfitting is clearly evident. It performs relatively well in recognizing "disgust" and "fear," with F1-scores of 0.61 and 0.63, respectively. However, it struggles to accurately identify "anger," "joy," and "sadness," with an F1-score below 0.5. The overall accuracy stands at 0.51, indicating that the model correctly predicts the emotions for 51% of the instances in the dataset. With **FastText** embedding, the accuracy of the classifier improved. We ran 50 epochs using fasttext-embedded train data and evaluated the data on the test set. After 25 epochs the accuracy was 55.8% with a loss of 1.7432. We used **GloVe** embedding for the same model architecture with an embedding dimension of 300. After 25 epochs we get slightly lower accuracy on the test set (54.68%) but the total loss on the test set was a bit higher (2.3741) than Fasttext embedding.

In **Bi-LSTM**, the model was trained using sparse categorical cross-entropy as the loss function and Adam as the optimizer with a learning rate of 0.001. Overall, this architecture is designed for sequence classification tasks and leverages bidirectional LSTMs to capture contextual information from the input sequences. We ran our training set with this model for 10 epochs. After 10 epochs the accuracy on the test set was 57.44% with a total loss of 2.706. The loss curve shows clear signs of overfitting as the loss on the validation set was always increasing while the loss on the training set was reaching zero. With **FastText** embedding we got better results in terms of the loss and accuracy. The loss was steadily decreasing for both the train and test sets. After 10 epochs, we got a test set accuracy of approximately 57%. Also, the signs of overfitting were not present in the analysis. With **GloVe** embedding we got an overall test accuracy of approximately 54% in 10 epochs. The loss curve on the validation set decreased for some time but then reached the same level where it started in the end creating a bowl-shaped curve.

Overall, the performance of traditional ML algorithms, especially SVM showed superior performance. We got the best result in terms of accuracy and precision with SVM trained on tfidf vectorizer. In terms of deep learning algorithms, Bi-LSTM performed better than MLP and Convo1D. However, the classifier model couldn't outperform SVM. Due to the smaller dataset, the Neural Network models didn't perform well and showed clear overfitting across all variations. Moreover, our dataset has imbalanced classes so we balanced our dataset using a combined technique, SMOTE-Tomek to balance the dataset for each emotion class. For the Bangla language, there is a limitation in datasets that are publicly available to date. To improve our model's accuracy, we can take some measures as well. Such as collecting more data and balancing the emotion classes equally to avoid overfitting. Other applied algorithms perform well as well. The results of these algorithms are quite similar. We need to collect more data and enhance our dataset to improve the performance of other algorithms. We observed from the discussion above that, in the event of 'Surprise', the algorithms perform poorly, but for other emotions, Naive Bayes and other algorithms used in this research perform better.

Methods	Precision	Recall	F1-score	Accuracy (%)	Loss
MLP	0.51	0.51	0.51	51.00	3.7350
Convo1D	0.51	0.51	0.51	50.60	2.7387
Bi-LSTM	0.59	0.57	0.58	57.44	2.0706

Table 6: Performace Analysis for Traditional Deep Learning Methods using TF-IDF

Methods	Precision	Recall	F1-score	Accuracy (%)	Loss
Convo1D	0.56	0.55	0.55	55.80	1.7432
Bi-LSTM	0.53	0.52	0.52	54.88	1.2763

Table 7: Performace Analysis for Traditional Deep Learning Methods using FastText

Methods	Precision	Recall	F1-score	Accuracy (%)	Loss
Convo1D	0.55	0.54	0.54	54.68	2.3741
Bi-LSTM	0.53	0.52	0.52	55.40	1.4535

Table 8: Performace Analysis for Traditional Deep Learning Methods using GloVe

4 Discussion & Conclusion

The proposed study outlines a comprehensive technique for constructing an emotion classification model that is particularly designed for Bengali texts. The objective of this research is to provide a significant academic contribution to the field of emotion classification in languages with limited resources, specifically focusing on Bengali. Our study effort has been a noteworthy attempt in the field of natural language processing and emotion analysis. We established the capability of effectively identifying and categorizing emotions in Bangla text through thorough testing and analysis. Our research not only added to the increasing knowledge in the field of emotion identification but also addressed the unique issues of dealing with Bangla text, a complicated and underutilized language in this context. We have obtained outstanding results in reliably recognizing emotional states in Bangla text by using cutting-edge machine learning and deep learning techniques, offering vital insights into the emotional components of human communication in this language. Furthermore, our findings have practical implications for enterprises operating in Bangla-speaking regions, mental health monitoring, and tailored content selection. The capacity to extract emotion from Bangla text opens the way to more effective communication and improved user experience.

In Bangla Language Processing, emotion detection is a relatively new problem. It's a problem of text classification. The major challenge for our system is the development of a good-quality dataset that can be used to teach our system or to provide our system with a knowledge base that can meet our main goals. We faced problems with the lack of available data for our research. We, therefore, created a dataset for the implementation of this project of subjective emotional text. We developed a system that can define six basic emotions such as happiness, sadness, anger, fear, surprise and disgust. We applied, evaluated and demonstrated the accuracy of different machine learning algorithms. These results were useful in developing the system.

Our future work includes a system that not only confides in 6 classes but also widens our domain. We are following different approaches to help us predict the emotions more accurately. From our initial study, we have come to the conclusion that using the neural network approach, LSTM and Bi-directional LSTM encoder and even Transformer-based approaches to be precise with the weighted optimizer could give the capability of understanding both the semantic and syntactic meaning of a sentence because it can perform well for text-based datasets. As we progress forward, there are still areas that need to be explored and refined. Fine-tuning the models, expanding the dataset, and experimenting with multi-modal data sources should result in even more accurate emotion identification systems. Furthermore, in emotion analysis, we should include ethical and cultural considerations to ensure that our models respect privacy and cultural sensitivities.

5 Contributions

The contribution of the authors in this research is as follows:

Md Moinul Islam
Conceptualization, Methodology, Software, Writing (review and editing), LaTeX writing, Performance Evaluation, Validation

Taufiq Ahmed
Literature Review, Methodology (data collection, preprocessing, feature extraction), Software, Writing (original draft preparation)

K H M Burhan Uddin
Methodology (feature extraction, model implementation), Software, Writing (original draft preparation and editing)

Efta Khairul Bashar
Investigation, Literature Review, Methodology (feature extraction, model implementation), Software, Writing (original draft preparation)

References

[1] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[2] Salim Alami and Omar Elbeqqali. "Cybercrime profiling: Text mining techniques to detect and predict criminal activities in microblog posts". In: *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*. IEEE. 2015, pp. 1–5.

[3] Pierpaolo Basile and Nicole Novielli. "Uniba: Sentiment analysis of English tweets combining micro-blogging, lexicon and semantic features". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. 2015, pp. 595–600.

[4] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[5] Shaika Chowdhury and Wasifa Chowdhury. "Performing sentiment analysis in Bangla microblog posts". In: *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*. IEEE. 2014, pp. 1–6.

[6] Michael Crawford et al. "Survey of review spam detection using machine learning techniques". In: *Journal of Big Data* 2.1 (2015), pp. 1–24.

[7] Avishek Das et al. *Emotion Classification in a Resource Constrained Language Using Transformer-based Approach*. 2021. arXiv: `2104.08613 [cs.CL]`.

[8] Dipankar Das and Sivaji Bandyopadhyay. "Analyzing Emotion in Blog and News at Word and Sentence Level." In: Jan. 2009, pp. 1402–1414.

[9] Jiawen Deng and Fuji Ren. "A Survey of Textual Emotion Recognition and Its Challenges". In: *IEEE Transactions on Affective Computing* 14.1 (2023), pp. 49–67. DOI: `10.1109/TAFFC.2021.3053275`.

[10] Paul Ekman et al. "Basic emotions". In: *Handbook of cognition and emotion* 98.45-60 (1999), p. 16.

[11] Matt W Gardner and SR Dorling. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences". In: *Atmospheric environment* 32.14-15 (1998), pp. 2627–2636.

[12] Jiuxiang Gu et al. "Recent advances in convolutional neural networks". In: *Pattern recognition* 77 (2018), pp. 354–377.

[13] Asif Hassan et al. "Sentiment analysis on bangla and romanized bangla text using deep recurrent models". In: *2016 International Workshop on Computational Intelligence (IWCI)*. 2016, pp. 51–56. DOI: `10.1109/IWCI.2016.7860338`.

[14] MD Asif Iqbal et al. "Bemoc: A corpus for identifying emotion in bengali texts". In: *SN Computer Science* 3.2 (2022), p. 135.

[15] Md. Moinul Islam et al. “An Automated Candidate Selection System Using Bangla Language Processing”. In: *Intelligent Computing and Optimization*. Ed. by Pandian Vasant, Ivan Zelinka, and Gerhard-Wilhelm Weber. Springer International Publishing, 2021. ISBN: 978-3-030-68154-8.

[16] Thorsten Joachims. “Text categorization with support vector machines: Learning with many relevant features”. In: *European conference on machine learning*. Springer. 1998, pp. 137–142.

[17] Rie Johnson and Tong Zhang. “Deep pyramid convolutional neural networks for text categorization”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 562–570.

[18] Armand Joulin et al. “FastText.zip: Compressing text classification models”. In: *arXiv preprint arXiv:1612.03651* (2016).

[19] DongHwa Kim et al. “Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec”. In: *Inf. Sci.* 477 (2019), pp. 15–29.

[20] Vishnu M Menon and HA Rahulnath. “A novel approach to evaluate and rank candidates in a recruitment process by estimating emotional intelligence through social media data”. In: *2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*. IEEE. 2016, pp. 1–6.

[21] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. “Spam filtering with naive bayes-which naive bayes?” In: *CEAS*. Vol. 17. Mountain View, CA. 2006, pp. 28–69.

[22] Sagar More et al. “Automated CV Classification using Clustering Technique”. In: (2019).

[23] Muhammad Mahmudun Nabi, Md. Tanzir Altaf, and Sabir Ismail. “Detecting Sentiment from Bangla Text using Machine Learning Technique and Feature Analysis”. In: *International Journal of Computer Applications* 153.11 (Nov. 2016), pp. 28–34. ISSN: 0975-8887.

[24] Sarwat Nizamani et al. “Modeling suspicious email detection using enhanced feature selection”. In: *arXiv preprint arXiv:1312.1971* (2013).

[25] Harris Partaourides et al. *A Self-Attentive Emotion Recognition Network*. 2019. arXiv: `1905.01972 [cs.CL]`.

[26] Animesh Kumar Paul and Pintu Chandra Shill. “Sentiment mining from bangla data using mutual information”. In: *2016 2nd international conference on electrical, computer & telecommunication engineering (ICECTE)*. IEEE. 2016, pp. 1–4.

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[28] Md. Ataur Rahman and Md. Hanif Seddiqui. *Comparison of Classical Machine Learning Approaches on Bangla Textual Emotion Analysis*. 2019. arXiv: `1907.07826 [cs.CL]`.

[29] Irina Rish et al. “An empirical study of the naive Bayes classifier”. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.

[30] Fabrizio Sebastiani. “Machine learning in automated text categorization”. In: *ACM computing surveys (CSUR)* 34.1 (2002), pp. 1–47.

[31] Fabrizio Sebastiani. “Text categorization”. In: *Encyclopedia of Database Technologies and Applications*. IGI Global, 2005, pp. 683–687.

[32] Armin Seyeditabari, Narges Tabari, and Wlodek Zadrozny. “Emotion detection in text: a review”. In: *arXiv preprint arXiv:1806.00674* (2018).

[33] R Suresh and SR Harshni. “Data mining and text mining—a survey”. In: *2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*. IEEE. 2017, pp. 412–420.

[34] Andrea Tundis and Max Mühlhäuser. “A multi-language approach towards the identification of suspicious users on social networks”. In: *2017 International Carnahan Conference on Security Technology (ICCST)*. IEEE. 2017, pp. 1–6.