Bangla Text Classification Using Machine Learning and Deep Learning Techniques

by

Kamrus Salehin
17101164
Fahim Ahmed
21341060
Md. Ashifun Nabi
17301152
M. Kaosar Alam
17301117

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

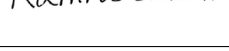
Department of Computer Science and Engineering
Brac University
September 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

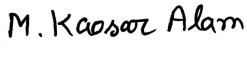
Student's Full Name & Signature:



Kamrus Salehin
17101164

Fahim Ahmed
21341060

Md. Ashifun Nabi
17301152



M. Kaosar Alam
17301117

Approval

The thesis/project titled "Bangla Text Classification Using Machine Learning and Deep Learning Techniques" submitted by

1. Kamrus Salehin (17101164)
2. Fahim Ahmed (21341060)
3. Md. Ashifun Nabi (17301152)
4. M. Kaosar Alam (17301117)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 26, 2021.

Examining Committee:

Supervisor:
(Member)

Faisal Bin Ashraf
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

At present, we have seen everything is getting digitized where technology almost takes full control over our life. As a result, a massive number of textual documents are generated on online platforms and news articles are no exception. People prefer to get connected with online news portals as they are updated every single hour. Newspaper articles have so many categories such as politics, sports, business, entertainment, etc. Recently, we have noticed the rapid growth and increase of Bangla online news portals on the internet. It will be helpful for the online readers to get recommended the preferable news category which assists them in locating desired articles. Manually categorizing news articles takes a huge time and effort. So, text categorization is necessary for the modern day, as enormous amounts of uncategorized data are an issue here. Although the study has improved in categorizing news articles greatly for languages such as English, Arabic, Chinese, Urdu, and Hindi. Among others, the Bangla language has shown little development. However, some approaches applied to categorize Bangla news articles, using some machine learning algorithms where resources were minimum. We have applied five machine learning classifiers and two neural networks to categorize Bangla news articles. To show the comparison between applied algorithms, which one is performing better, we have used four metrics that measure performance.

Keywords: Bangla news articles; Text categorization; Machine learning; Classifiers; Neural networks; Comparison

Acknowledgement

First and foremost, praise be to the Almighty. We would like to acknowledge the constant support of our supervisor, Faisal Bin Ashraf sir. He has accepted us as his thesis students and provides us helpful advice whenever we faced difficulties. We want to thank Brac University authority for allowing us to complete our thesis work from home during this unfavourable event of pandemic. Finally, we will not be able to achieve our goals without the continuous support from our parents. Thanks to their kind support and prayers, we are on the approach of graduating.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans have stretched the bounds of what they can think by mixing artificial and human brains as the world becomes more reliant on technology. Artificial Intelligence (AI) is the result of this process which has gained so much popularity all over the world. Natural language processing (NLP) is a topic of AI that has a significant impact in the research area of text classification. It is one of the famous techniques for searching, analyzing, understanding, and obtaining information from text-based data. Additionally, there are numerous human languages available, and each individual writes the content using their native language, such as Spanish, Bangla, Chinese, English etc. However, it is the computer's responsibility to analyze and determine the meaning of such texts and NLP enables computers to extract valuable meaning more intelligently. In recent years, NLP has grown much more popular. This technique is utilized in a variety of applications, including text classification, information extraction and tracing, speech tagging, and so on.

Text classification is a technique for classifying documents into a given set of categories using NLP. On the other hand, text classification is a challenging problem for the high dimensionality of the feature vector, which contains unimportant and unrelated data. Numerous approaches for reducing feature approaches have been presented for the purpose of removing unnecessary features and minimizing the dimension of the feature vector. A machine learning model uses a relevant and reduced feature vector to achieve better classification results.

As the smartphone users are increasing rapidly so activity of people on internet also rising as well. As a result, online content has increased significantly in recent years and news content is no exception. Those days are gone by when people used to wait for newspaper before breakfast. Nowadays people can update themselves with the latest news through online news portals in every seconds. Due to the growth of Bangla online news portal, a large numbers of Bangla news articles are published daily. So, the extensive and increased electronic availability of Bangla text documents enhances the necessity of automatic methods to analyze those text document's content. If text documents are categorized according to their appropriate categories, then it will be quick and efficient to search and retrieve information. There are some other usage of text classification besides news article classification such as email filtering, spam detection, sentiment analysis etc.

In our study, we have chosen the dataset of Bangla online news article which is consists of 12 different classes. We have preprocessed data by removing non-Bengali words, digits, punctuation and stopwords. Term frequency-inverse document frequency (TFiDF) is used to select feature. After that, we applied machine learning classifiers which are logistic regression, multinomial naive Bayes, support vector machine (SVM), random forest and extreme gradient boosting (XGBoost). Besides that we have also used two deep learning classifiers: multilayer perceptron and Long Short Term Memory (LSTM) model. Four performance metrics are used. LSTM provides the best result among all the classifiers that we have applied in our work. The details of these are presented in chapter 3, 4, 5 and 6.

1.1 Problem Statement

Text classification is a process that is used to overview the whole system. It is extremely useful for managing Web content, search engines, and email filtering, among other things. The advancement of technology has heightened interest in text categorization problems. There are plenty of Bangla language documents available on the internet, which are both valuable and tough to classify effectively into their corresponding semantic categories. The searching and information retrieval is quick and easy if documents are categorized among their relevant categories. Furthermore, a reader prefers to read the articles which is most interesting to him/her from screen while reading an electronic newspaper. Therefore, readers are most likely to be interested in receiving articles from their preferred places. Consumers anticipate receiving customized edition of newspapers with articles which are appropriate for them prominently displayed on the initial pages. This type of work is carried out on a variety of worldwide news websites and blogging platforms. Thus, text categorization is a task that has both commercial and labor-saving implications. In the text categorization field, extensive researches on many languages have been performed. Several supervised learning methods have been used to categorize text documents, which include K-Nearest Neighbor [15], Decision Trees [1], Naive Bayes [12] and Support Vector Machine [2]. These algorithms are popular in the text categorization field that also have been used for news text categorization for different language such as Hindi, Spanish, Arabic, Urdu etc. However, several methodologies are presented in Bangla language also but those studies worked with relatively small datasets [18]. Although, The Bangla language has a rich history and it's one of the most spoken languages all over the word Native speakers of Bangla language are approximately 8% of the world population [5]. Thus, it is important to automatically arrange and categorize Bengali text so that users may conveniently find relevant information.

1.2 Aim of study

We aim to categorize Bangla news content using the machine learning algorithms and neural networks. This work will open a scope for the future researchers as it will give them a short brief about the performances of our applied algorithms for Bangla news content dataset. To compare which supervised learning approach is performing better we have used four performance metrics which are accuracy, precision, recall and f1-score. Performances of the algorithms applied on the classification of Bangla text are shown in our work and our experiment were conducted on 75951 Bangla text documents that included twelve text categories.

1.3 Thesis Outline

In chapter 2, we have discussed some of the related works that have been done by other researchers. The classifiers we have used are described in chapter 3. Chapter 4 describes the dataset, how it has been collected and how we have preprocessed it. We have analyzed our findings in chapter 5. At the end we drew a conclusion about our work in the 6th chapter.

Chapter 2

Literature Review

In this chapter, we have covered several previous research work which is relevant to our thesis work. In this paper [4], the authors mentioned that typical approaches like Naïve Bayes (NB), vector space model (VSM), and LLSF classifiers were not good enough for categorizing Chinese text. They also mentioned the unavailability of Chinese corpus to evaluate the systems of categorizing Chinese text. This paper describes the implementations of the k Nearest Neighbor system (KNN), Support Vector Machines (SVM), and Adaptive Resonance Associative Map (ARAM) for categorizing Chinese text. KNN and SVM are used as these two are proven to be the best working methods for categorizing English text. Until then, ARAM had not been used for document classification. Authors built the Mandarin News Corpus-based, a People's Daily corpus which the Linguistic Data Consortium (LDC) had provided for evaluating these three machine learning methods. They analyzed and differentiated the capabilities of these methods by mining the knowledge of categorization from high-dimensional, sparse, and relatively noisy document feature vectors. The authors adopted a top-quality bi-gram model for the segmentation of every training documents into sets of tokens. KNN can be called as a lazy learning method because of its not performing any off-line learning to create a specific category knowledge representation. Optimal separating hyper-plane (OSH) across the training data points are identified by SVM, and it makes representative data instance-based classification decisions. From the patterns of input training, ARAM produces recognition categories. Recognition categories can be treated as associative clusters of the training patterns, which work like a representation of the categorization knowledge based on a dynamic rule. They found the output of ARAM is slightly better than that of KNN and SVM.

In this article [26] the authors have mainly focused on the performance of various text classifiers on Bangla language. For the dataset, they have collected a total of 8000 Bangla text documents with 8 domains and each domain consisting of 1000 Bangal text documents. After collecting data, preprocessing is done by tokenization. A total of 23,36,821 tokens were recovered from all of the text documents. After tokenization the stopwords are removed which results in a final of 10,91,960 terms retrieved. Then feature selection and feature extraction is performed. They have implemented text classification methods like Multilayer perceptron, Random forest, Support Vector machines, Naive Bayes Multinomial and KStar. For non-reduced set MLP had an accuracy of 98.30%, RF had an accuracy of 98.26%, SVM had an

accuracy of 51.79%, NBM had an accuracy of 96.50% and KStar had an accuracy of 95.61%. So after performing 15 fold cross validation and 700 training iterations, MLP had the highest average precision of 0.987. After conducting the experiments, the authors concluded that their proposed method performs very well in English and Bangla language as well.

Khorsheed and Thubaity [8] used diversified datasets for their work. Their datasets include poems, religious topics, forums, newspaper articles, and also web articles. After preprocessing the dataset by removing unnecessary punctuation, numbers, diacritics, they divide their data for training and test separately. They used a single word as a representative feature, and according to them, it has been proven to be efficient in a wide range of applications. Among thousands of features, they do not use all the features for classification. Feathers with higher values are selected as representative features. They used CHI squared method for feature selection to avoid overfitting and unexpected results. Their data representations include a matrix where rows correspond to the texts in the training data, the columns correspond to the selected feature, and the value of the cell represents the weight of the features in the text. They used the TFiDF method for weighting the data. They worked on Naive Bayes, decision tree, MLP, k-nearest neighbor, and support vector machine. They used an open-source tool named Rapid miner and Clementine, a data mining software from SPSS Inc. This tool has several functions for feature selection data representation and also has classification algorithms. So only by using the data sets, they got the results quickly. They made all the combinations of feature selections and data representations to decide what combination outputs the most accurate result on which type of data set. All of their data sets are Arabic datasets. According to them, classification accuracy was 97% for the Arabic poem dataset. They said that the Naïve Bayes classification has the highest average accuracy of 64.41% over the other algorithms. The second most was 60.26 % by support vector machine algorithm. However, when the CHI square term selection method were used, the SVM classifier shows the highest accuracy of 72.15%. The least accurate term selected is DF term selector. They also said the Boolean and the LTC data representation outperform the best result for classification. They made the table for every criterion to show the best combination.

An article on cyberbullying detection [31] authors used deep neural networks to identify any abusive text or comments in Bangla. In the paper the authors have proposed a binary classification model and multiclass classification to classify their model. The dataset consists of 44001 comments from various Facebook pages which are classified into five categories. After collecting the dataset, they have implemented methods like stopwords removal, string tokenization and padded sequence conversion to preprocess the data and Word2Vec embedding model for word embedding. Finally a hybrid model is formed using both a binary classification model and a multiclass classification model. The accuracy of the binary classification model is 87.91 percent, while the accuracy of the multiclass classification model is 85 percent when using the ensemble approach following the neural network. In comparison to current work in Bangla language processing, the authors claim to have achieved a reasonable level of accuracy.

In this article [14], the authors have proposed a text categorization system which is an efficient hybrid method is for stemming Arabic text. They have compared this method with other methods such as F-measure of Naïve Bayesian classifier and the Support Vector Machine classifier. In their work, they have mainly tried to introduce a new and efficient stemming method as pre-processing tools for Arabic language in Text Categorization as they believe there is a lot of text categorization systems for English and other languages but very few for the Arabic language. The authors have compared different types of Arabic Stemming methods such as Root-Based Approach, Stem-Based Approach, and Statistical Approaches and have found out that they are not a complete solution for the Arabic language. To solve this problem, they have introduced a new efficient stemmer as a hybrid algorithm of the three existing approaches. Their Hybrid method combines these three different methods with a bit of modification as well as Naïve Bayesian (NB) and Support Vector Machine (SVM) classifiers to perform text classification. After performing a series of experiments, they have conducted that no stemming methods are appropriate for the Arabic language as they do not have a high accuracy rate. So their proposed System for Arabic Text Categorization, which has used a Naïve Bayesian classifier and SVM classifier to perform text classification, has got better performance than other existing Arabic Text Categorization.

The author Saleh Alsaleem [10] has showed the comparison between two supervised learning methods SVM and NB to categorize Arabic text document. In their work, dataset was collected from Saudi newspapers which has 5121 documents consists of 7 categories. Recall, f1 and precision are the performance metrics which is used to measure the performances of the applied algorithms. They have demonstrated that the SVM classifier performed better against their collected Arabic text dataset in terms of mentioned performance metrics.

Another article on Bangla Text document categorization [19] Stochastic Gradient Descent (SGD) classifier has been used, which is a linear classifier optimization method. For this paper, authors have used a dataset consisting of newspaper articles which is divided into nine classes. In their model, first they have extracted features using term frequency and Inverse document frequency. From using distinct features, they have implemented the SGD algorithm. Finally using F1-score they have measured the performance of their given method. From their analysis, Ridge classifier had accuracy of 93.4%, Perceptron had accuracy of 91.43%,Passive-Aggressive had accuracy of 93.21%, SVM had accuracy of 93.80%, Naive-Bayes had accuracy of 92.17%, Logistic-Regression had accuracy of 93.11% and SGD classifier had highest accuracy of 93.85%.

A research was conducted on Urdu text classification [27] using three popular classifiers such as SVM, Naïve Bayes (NB), and KNN. In their research work, Urdu text dataset is used which is mainly collected from the newspaper articles and number of documents in the dataset is up to 16,678. The authors of this paper have also applied feature extraction method namely TF-IDF for scoring the features. To remove the least significant features some of the feature selection method is also used such as Chi-square, Gain Ration and Information gain. The result of this paper shows that the SVM classifier achieved relatively better with reasonable accuracy

(68.73%) and higher efficiency while compared to the other two classifiers..

In the article [2] based on learning features of text categorization using support vector machine, the author claims that one of the most important strategies for managing and organizing text data is text classification. The author of this paper has analyzed the advantages of Support Vector Machines in the research area of text categorization. Text categorization is the process of classifying documents into a set of predetermined categories in which a document is assigned to several categories, one category alone, or none at all. To begin, documents are converted into a format appropriate for a learning algorithm, with word stems being favored. The representation has a high degree of dimensionality. In this instance, the author utilized information gain to choose a subset of features, and the TFC variation was chosen because it increases efficiency by scaling the feature vector's dimensions with their inverse document frequency. The author has demonstrated how SVM works and the benefits of utilizing it, such as how it utilizes overfitting protection to handle huge feature spaces, how well it is suited for issues with dense concepts and instances, and how it can discover linear separators. We can see from the results of the experiments that SVMs regularly beat current techniques on text classification tasks. SVM also eliminates the requirement for feature selection, making text classification much more straightforward. Apart from comparisons with other approaches, it performs well in all experiments, avoiding any major failures.

Mandal and Sen [18] have used four machine learning techniques including Support Vector Machine (SVM), Naïve Bays, K-Nearest Neighbour (KNN) and Decision Tree to categorize Bangla web documents. The dataset that they have used for their research purpose is BD corpus has a total of 22,218 words and consists of 1000 documents with five categories including business, sports, health, technology and education. They have employed the feature extraction method tf-idf to extract significant features from the documents. In their study, evaluation metrics such as recall, precision, and F-measure are used to measure the performance of performed classifiers. In their experiment, SVM achieved the best result with 89.14% average accuracy and KNN performed the lowest with 74.24% average accuracy among all the applied classifiers. They have also shown the training time of each classifier where SVM performed with the smallest training time.

Chapter 3

Methodology

This chapter discusses about the methods we are using in our work. In order to get the most desired output, we have used five machine learning classifiers and two neural network models which have well performed in other languages for text classification. The brief description of our used methods are given below:

3.1 Logistic Regression

In machine learning field, logistic regression (LR) has become an important technique. Moreover, Logistic regression is one of the well-renowned methods for text categorization. In addition, logistic regression is also considered as a supervised learning algorithm. LR was frequently utilized in the realm of data retrieval. In past research, Logistic regression has been investigated in machine learning area, specifically in English text categorization [6] [7].
In Logistic Regression the main focused word is "logistic" which refers to a logistic function that performs classification operations in the algorithm. In addition, Logistic function is the foundation of the logistic regression model [17]. The following figure (3.1) [36] shows a logistic or sigmoid curve and the equation (3.1) is:

$$f(x) = \frac{M}{1 + e^{-k(x-x_0)}} \tag{3.1}$$

Here, e = Euler's number, $x_0 = x$ value of the sigmoid's midpoint, M = maximum value of curve, k = steepness or logistic rate of the curve and $f(x)$ = function output.

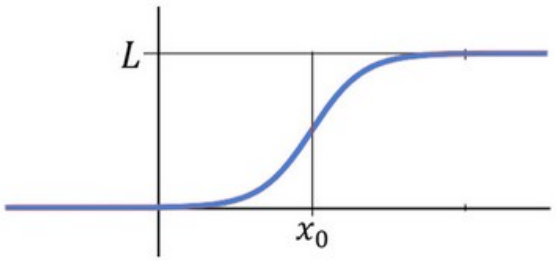


Figure 3.1: Logistic curve

Logistic Regression is used to classify categorical dependent data by utilizing predictor variables. It converts probability scores from categorical dependent variables, hence creating a connection between categorical variable which is dependent and a continuous variable which is independent. Binary logistic regression and multinomial logistic regression are two forms of logistic regression. The primary distinction between these two sorts is the type of labels used in them. Multinomial logistic regression is performed when the labels include several values [24]. The model's one of the fundamental assumptions is that the independent and dependent variables do not share a linear relationship [9]. When it comes to text classification the Logistic regression model identifies a vector including variables, assesses for each input variable coefficients, then assumes the text class using a word vector.

3.2 Multinomial Naive Bayes

The multinomial Naive Bayes (MNB) algorithm is a type of approach that uses probabilistic learning which is widely used in text classification. MNB frequently use a parameter learning approach known as Frequency Estimate when given a collection of labeled data [3]. It computes appropriate frequencies from data to estimate word probabilities. Bayes rule is used to categorize data by picking the class which most probably have produced the instance. The Bayes theorem, which was developed by Thomas Bayes, determines the likelihood of an event occurring established on previous information of the event's circumstances. It is based on equation (3.2). For any variables x and y,

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)} = \frac{p(x,y)}{p(y)} \tag{3.2}$$

From the point of view of Bayesian learning, which assumes that word distributions in texts are created by a certain parametric model, the parameters of which will be estimated from the training data. Equation (3.3) shows multinomial naive Bayes model.

$$p(x|y) = \frac{p(x)\prod_{i=1}^{n} p(w_i|x)^{fi}}{p(d)} \tag{3.3}$$

Here, fi is the amount of times a word w_i appears in a text. Given the class value x, $p(w_i|x)$ is the probability which is dependant on a word w_i will appear in a document d, n is the amount of unique words in the document. $p(x)$ is the probability of finding a document with the class label x in the collections of document. A generative parameter learning technique called estimate of frequency, which is essentially the frequency that is relative in data, can be used to estimate the parameters in equation (3.3). Using the relative frequency of the term w_i in texts which belongs to class x, frequency estimate calculates the conditional probability $p(w_i|x)$.

3.3 SVM

Support Vector Machine (SVM) is a model that is based on supervised learning. It can analyze data for regression analysis and classification. It basically tries to find a boundary in a 2-dimensional space, between the data points of two different classes of a dataset. This boundary is called hyperplane. In general, the purpose is finding a hyper-plane which will maximize the distinction of the data-points to the relevant classes in a n-dimensional space. The closest data-points to that hyperplane are called Support Vectors. For, binary classification the hyperplane equation is:

$$w * x - b = 0 \tag{3.4}$$

Here, w is the usual direction of the hyper-plane, b is a threshold form and the value of x can vary for different instances. For a particular data point w, if the equation become positive then w will belong to a class. If the equation become negative, w will belong to another class.

Normally, multiclass classification is not supported by SVM. It supports binary classification. But SVM can support multiclass classification by using two different approaches. One-vs-One and One-vs-All.

In our thesis, we have used scikit-learn API [13] and it uses One-vs-One approach. In this approach, the multiclass problem has to be broken down into multiple binary classification problems and each pair of classes will have a binary classifier. That means for classifying data points from n classes dataset, the classifier will use n(n-1)/2 SVMs. Our dataset has 12 classes, so 66 SVMs are used here.

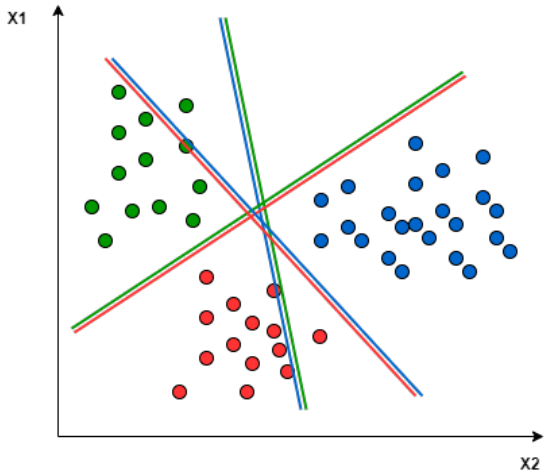


Figure 3.2: One-vs-One approach

In the above figure (3.2) [32], a three-class classification is shown with One-vs-One approach. Here, the blue-green line separates the green and blue points with maximum distance. Though the line is not concerned about the red points.

3.4 Random Forest

The random forest algorithm also comes under supervised learning algorithm that generates a forest consisting of several trees. Additionally, it is a widely used algorithm and can be used to solve regression and classification problems, but in most cases this algorithm is used for the purpose of solving classification problems. This method refers to a collection of decision trees which forms their nodes during the stage of preprocess [16]. The best feature is chosen from a random selection of features after numerous trees have been constructed [29]. Another approach that is generated employing the decision tree technique is to generate a decision tree. As a result, random forest is made up of those trees which generally classify new objects based on the given vector. All decision trees created are used to classify data. If we give that class tree votes, the most voted classification will be selected by the random forest from all of the forest's trees. In the random forest, there are some chances of mistakes based on two parameters where the first one is if a correlation exists between two trees in a forest, the rate of error will be increased. Secondly, all trees carry their own unique weight. For this, a powerful classifier is a tree with a lower error rate and vice versa.
Random forest has several characteristics, including:
1. It can perform efficiently in a large dataset and also Deals with a large number of input variables without deleting any of them.
2. It describes the variables that are significant in classification.
3. The trees or forests that are created can also be kept for later use.

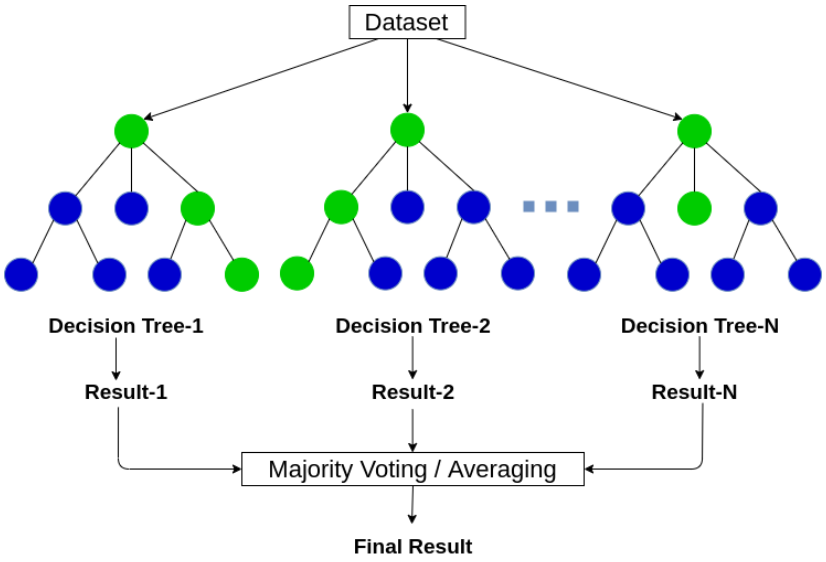


Figure 3.3: Example of a random forest [34]

Random forest classifier follows some steps to perform. They are:
1st step: Selecting K random data points from the training set.
2nd step: Using these K data points, developing decision tree.
3rd step: choosing the N number of tree which we want to construct.
4th step: Predicting the y value by building every N Tree trees for a fresh point of data and provide a data-point mean that is new over every anticipated values of y.
Figure (3.3) shows a random forest example.

3.5 XGBoost

XGBoost (eXtreme Gradient Boosting) is a gradient boosted decision tree extension that has been optimized for speed and performance. It is a great method for classifying text and is a greedy algorithm that qualifies as quick dataset training. It is based on the assumption that when the best possible next model is joined with prior models, the prediction of overall error is minimized as low as possible. As a result, each tree grows and learns from the one before it. We have used XGBoost for its reputed performance on big dataset [22].
If we consider a function, firstly it creates a series based on function gradients [33]. The equation below (3.5) represents a specific type of gradient descent. This represents the loss function to minimize. So, it gives the direction in which the function decreases. It's the loss function's fitted rate of change which is it's equal to the gradient descent learning rate.

$$F_{x_t+1} = F_{x_t} + \epsilon_{x_t} \frac{\partial F}{\partial x}(x_t) \tag{3.5}$$

The function (3.6) will act as an error measure, which will allow to reduce loss and maintain performance. The following series (3.6) will converge to the function's minimum.

$$f(x, \theta) = l(F((X_i, \theta), y_i)) \tag{3.6}$$

3.6 MLP

Multi-layer perceptron is one of the common and widely used neural network model in deep learning field [11]. A deep neural network model consists of three types of layers. The initial layer of the network is input layer which takes all the inputs. The final stage of the network is output layer. In our case, there are 12 nodes for 12 class of the output layer. Inside output and input layer, there is hidden layer which is a chain network of perceptron. There may be several hidden layers in MLP. In our case we have used 1000 hidden layer as default. We have used maximum iteration to 200. In the fully connected network of the perceptron, each node is a linear combination of weighted terms from the previous layers all nodes with an activation function. In other way, we can say that each node's value is the summation of all of its connected values dot product with its weight. By this process, every layers calculation has been done to get the last output layer and get the desired. MLP architecture has been shown in figure (3.4) [23].

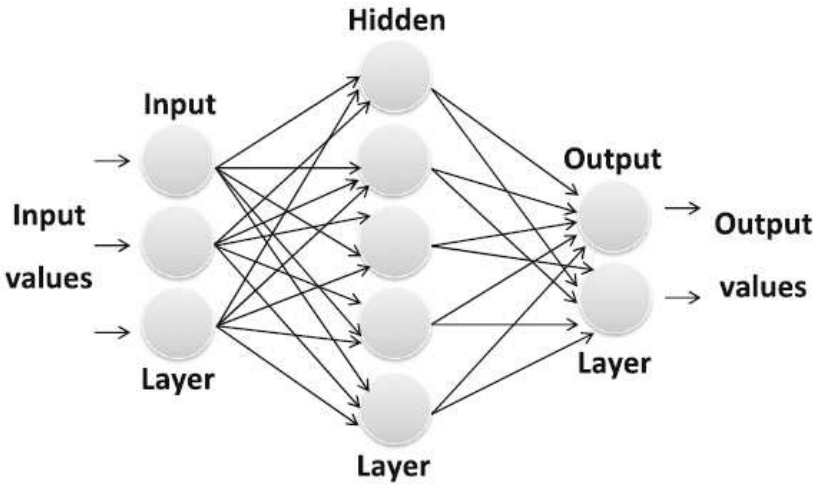


Figure 3.4: MLP architecture

3.7 LSTM

LSTM is a type of Recurrent neural network (RNN) which stands for long short term memory. The main difference of traditional RNN and LSTM is traditional RNN does not have persistent memory while LSTM has it [20]. In our dataset, we have used LSTM for classifying the data into category, as we need to capture or process the whole details of the specific element of the dataset. The decision will be an outcome of the whole together. So, we need such a system to capture the whole. This is called long term dependencies which is only classify most accurately with LSTM in maximum cases. This is because it solves the vanishing gradient problem of the RNN. For vanishing gradient problems, RNN except LSTM cannot be used in long term dependencies [20]. LSTM has a unique feature. With the help of activation function and three gates, it can decide which information should go to next level, which information should be forgotten and which is important to remember. The gates are input gate, forget gate and output gate. Figure (3.5) [21] shows LSTM architecture.

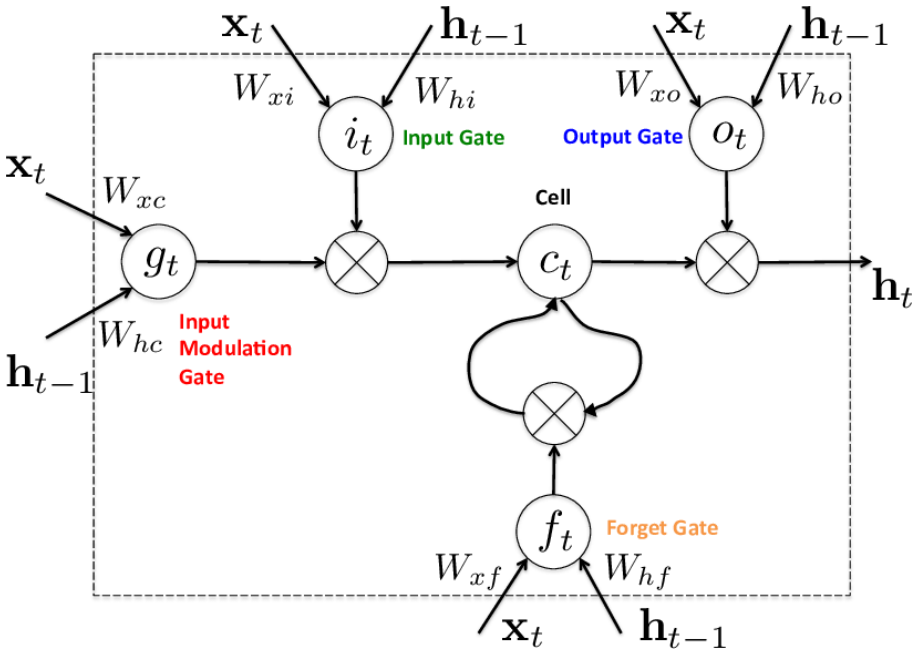


Figure 3.5: LSTM architecture

We set the vocabulary size 75000 and maximum length to 475. So, the model will go through 475 words for giving the category to us. The vector feature is 60 for each word. These parameters are given in the embedding layer. The dimension of the outer space of our LSTM layer is given 128. Our dataset has 12 classes, so we have used 12 as a dense layer units parameter for the model and batch size of 64 with 50 epochs. We have used the relu activation along with the softmax activation function because our dataset consists of more than 2 classes. RMSprop optimizer has been used with learning rate at 0.01. We have used accuracy as metrics and sparse categorical cross-entropy as loss function.

Chapter 4

Dataset Description

In order to classify Bangla text data correctly, we have taken various steps and process to achieve most accurate result.In this chapter, we try to bring the details of our data collection process, description of our collected dataset and data pre-processing steps. As to get the most outcome from machine learning and deep learning techniques we need a large dataset to train well enough, our first task is to collect the news data as we have implemented the classification on news data only.

4.1 Data Collection

Table 4.1: Data distribution of the dataset

Class	Samples	Training Samples	Testing Samples
Accident	5036	4029	1007
Art	2117	1694	423
Crime	6858	5486	1372
Economics	2731	2185	546
Education	9646	7717	1929
Entertainment	7913	6330	1583
Environment	3439	2751	688
International	4121	3297	824
Opinion	6430	5144	1286
Politics	15867	12693	3174
Science and Technology	2315	1852	463
Sports	9478	7582	1896

Other works that was done before consists of small data. So in our case we try for a large dataset and collected it. We have taken the dataset from here [30]. No other previous work has been done before with this type of large dataset for Bangla text classification. But we have seen that many other languages had used this type of large dataset and also diversified dataset for their work. Our total number of articles is 75951 which is divided into 12 classes. After collecting and preprocessing the dataset, we split them into training set and testing set separately. For training purpose we have used 80% of each class data and the rest 20% is used for test purpose. The table (4.1) overviews our collected dataset properties and how we

split them for our training and test purpose. Figure (4.1) represents the percentage of the class distribution of our dataset.

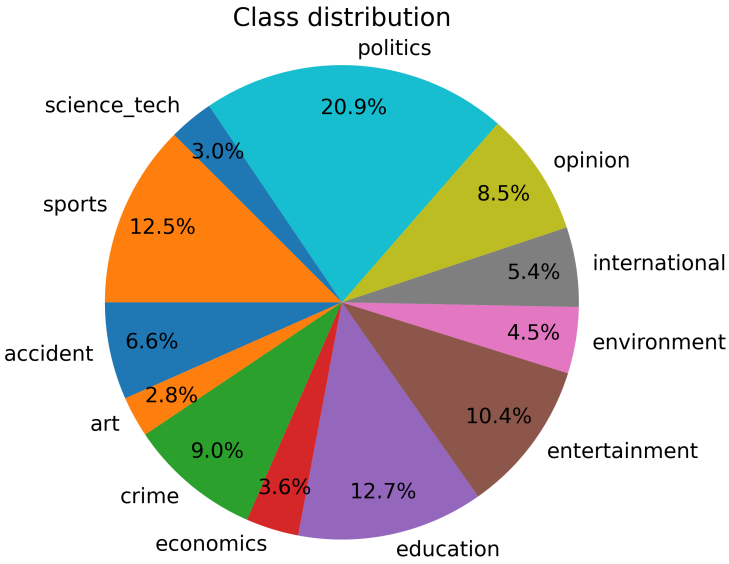


Figure 4.1: Class distribution

4.2 Data preprocessing

The data is not suitable to run on classification algorithm. To make it fit for the algorithms we need to go through several processing steps.
The dataset has 3 columns. Among them, we first eliminated unnecessary column. The class name and the article is remaining. Next, we find the duplicate article and remove them so that our dataset remains in unique. We have found 767 duplicate articles. All of them are removed.
Our dataset has to be in totally Bangla. But it has some punctuation, English words, digits etc. In order to remove these we used Unicode values to keep only Bangla words. After that we use bnlp toolkit [35] for stop-words removal because it is not useful to be in the data for classification. Our next task is to tokenize the entire data. We used tokenizer function for tokenize and mapped the class dataset to numeric value manually. For fitting into algorithm we need to select and extract features among the dataset. We have used each tokenized word as a feature and use TFiDF for feature selection.

The full form of TFiDF is term frequency inverse document frequency. Term frequency calculates the frequency of a term in a document.

$$TF(b) = \frac{\text{Amount of times term b appears in a specific document}}{\text{Total amount of terms in that document}} \tag{4.1}$$

Inverse document frequency calculates the importance of a term.

$$iDF(b) = \log\left(\frac{\text{Total amount of documents}}{\text{Amount of documents with term b in it}}\right) \tag{4.2}$$

$$TFiDF = TF * iDF \tag{4.3}$$

TFiDF gives us weighted values about which feature is important and which is not. Finally, we split with stratify into 80% for training purpose and rest 20% for test purpose so that each class is divided according to it. After all this processes, we get the data with weight and ready to run on classification algorithm.

Chapter 5

Result Analysis

This chapter will discuss about the performances of our applied classifiers. We have summarized the performances of our applied classifications by using confusion matrices. The output of a confusion matrix can be understood better with four parameters which are accuracy, precision, recall and f1-score. Equations of these parameters have some common variables. These are, TP=True Positives, TN=True Negatives, FP=False Positives, and FN=False Negatives.
Accuracy can be calculated by dividing the number of accurate predictions with the amount of predictions made in total.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Precision is the proportion of the accurate positive predictions to all positive predictions.

$$precision = \frac{TP}{TP + FP} \tag{5.2}$$

Recall is the fraction of the actual positive predictions among total actual positive instances.

$$recall = \frac{TP}{TP + FN} \tag{5.3}$$

F1-Score represents the harmonic average value of precision and recall.

$$F1Score = 2 * \frac{precision * recall}{precision + recall} \tag{5.4}$$

5.1 Logistic Regression

Table (5.1) contains the logistic regression classification report. Figure (5.1) has shown logistic regression confusion matrix.

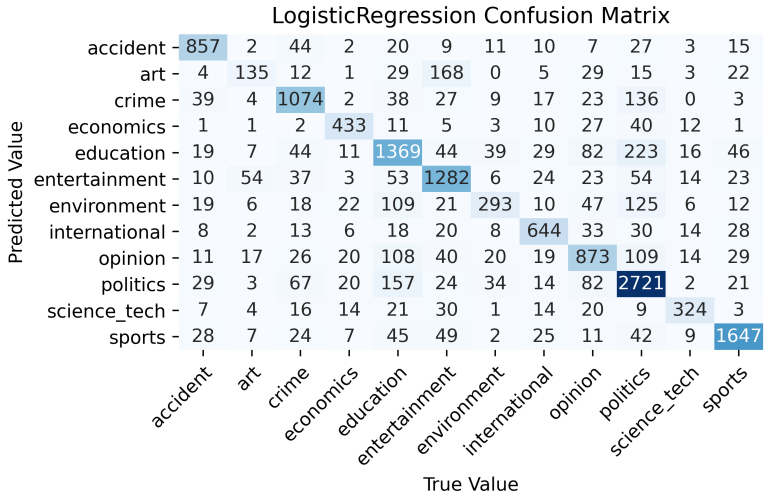


Figure 5.1: Logistic regression confusion matrix

Table 5.1: Logistic Regression classification report

Class	**precision**	**recall**	**f1 score**	**support**
Accident	0.83	0.85	0.84	1007
Art	0.56	0.32	0.41	423
Crime	0.78	0.78	0.78	1372
Economics	0.80	0.79	0.80	546
Education	0.69	0.71	0.70	1929
Entertainment	0.75	0.81	0.78	1583
Environment	0.69	0.43	0.53	688
International	0.78	0.78	0.78	824
Opinion	0.69	0.68	0.69	1286
Politics	0.77	0.86	0.81	3174
Science and Technology	0.78	0.70	0.74	463
Sports	0.89	0.87	0.88	1896
Accuracy	0.77	0.77	0.77	15191
Weighted average	0.76	0.77	0.76	15191

5.2 Multinomial Naïve Bayes

In this classifier we have noticed that changing the alpha value has improved the performance significantly. Alpha is a smoothing parameter in scikit-learn API for multinomial Naïve Bayes classifier. The default value for alpha is 1.0. We have changed the value to 0.01 and it improves the classification result. Figure (5.2) is showing the effect of changing the alpha value. Multinomial naive Bias confusion matrix is represented in figure (5.3). Table (5.2) indicates the multinomial Naïve Bayes classification report.

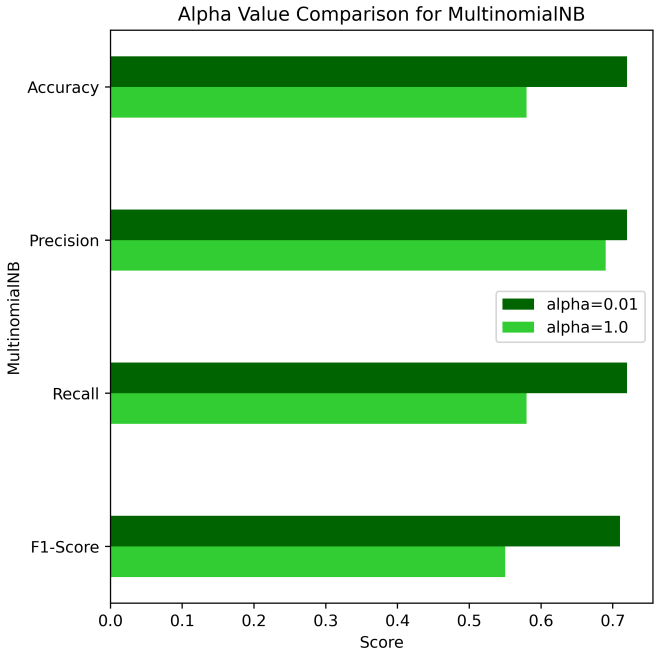


Figure 5.2: Effect of changing the alpha value

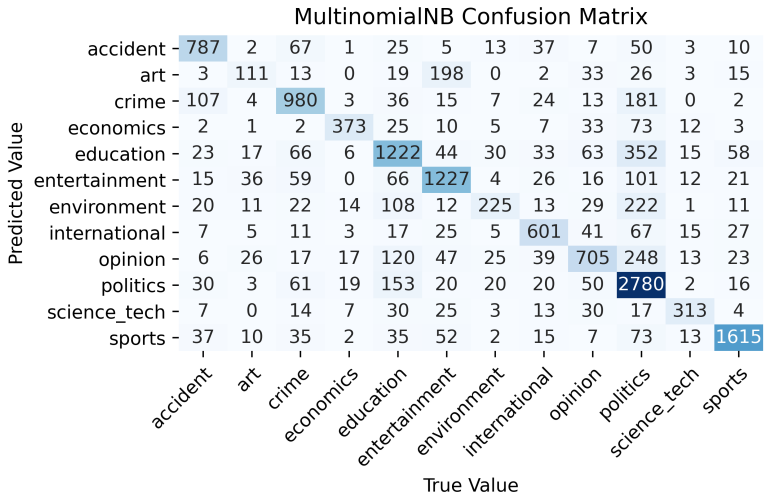


Figure 5.3: Multinomial naive Bias confusion matrix

Table 5.2: Multinomial naive Bias classification report

Class	precision	recall	f1 score	support
Accident	0.75	0.78	0.77	1007
Art	0.49	0.26	0.34	423
Crime	0.73	0.71	0.72	1372
Economics	0.84	0.68	0.75	546
Education	0.66	0.63	0.65	1929
Entertainment	0.73	0.78	0.75	1583
Environment	0.66	0.33	0.44	688
International	0.72	0.73	0.73	824
Opinion	0.69	0.55	0.62	1286
Politics	0.66	0.88	0.76	3174
Science and Technology	0.78	0.68	0.72	463
Sports	0.89	0.85	0.87	1896
Accuracy	0.72	0.72	0.72	15191
Weighted average	0.72	0.72	0.71	15191

5.3 SVM

Table (5.3) is representing the SVM classification report, SVM confusion matrix is in figure (5.4).

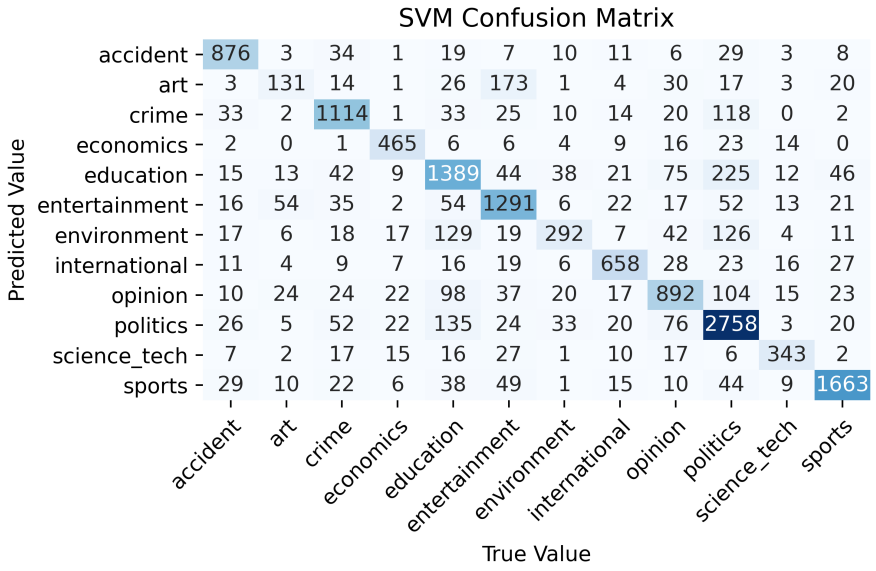


Figure 5.4: SVM confusion matrix

Table 5.3: SVM classification report

Class	precision	recall	f1 score	support
Accident	0.84	0.87	0.85	1007
Art	0.52	0.31	0.39	423
Crime	0.81	0.81	0.81	1372
Economics	0.82	0.85	0.83	546
Education	0.71	0.72	0.71	1929
Entertainment	0.75	0.82	0.78	1583
Environment	0.69	0.42	0.53	688
International	0.81	0.80	0.81	824
Opinion	0.73	0.69	0.71	1286
Politics	0.78	0.87	0.82	3174
Science and Technology	0.79	0.74	0.76	463
Sports	0.90	0.88	0.89	1896
Accuracy	0.78	0.78	0.78	15191
Weighted average	0.78	0.78	0.78	15191

5.4 Random Forest

For this classifier, if we make bootstrap parameter false, the performance slightly increases. Bootstrap parameter decides if a single tree can use some samples several times. False means every tree will be built by using the whole dataset. By making the change we can observe that, accuracy, precision and recall increase from 0.67 to 0.68, f1-score increases from 0.65 to 0.67. In the below table (5.4), classification report of random forest is generated and figure (5.5) represents the random forest classifier confusion matrix.

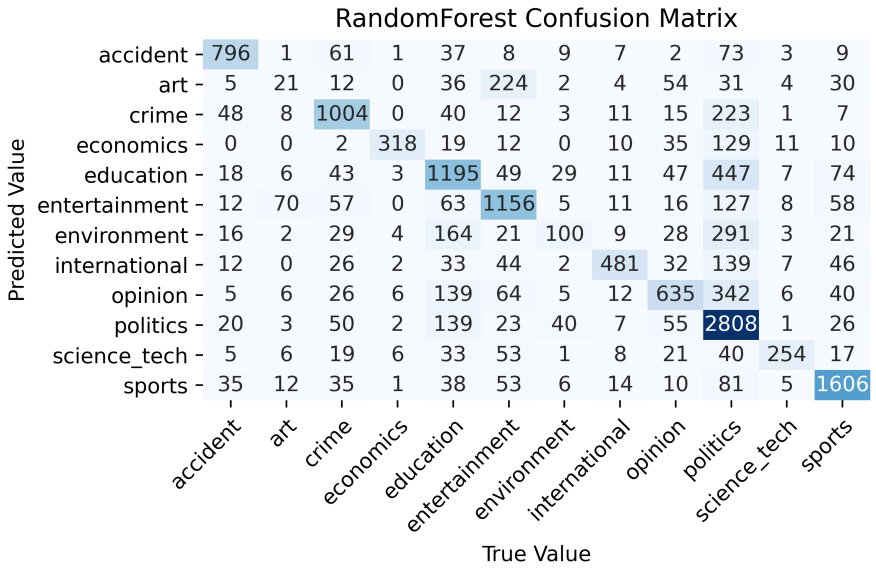


Figure 5.5: Random Forest confusion matrix

Table 5.4: Random Forest classification report

Class	precision	recall	f1 score	support
Accident	0.82	0.79	0.80	1007
Art	0.16	0.05	0.08	423
Crime	0.74	0.73	0.73	1372
Economics	0.93	0.58	0.72	546
Education	0.62	0.62	0.62	1929
Entertainment	0.67	0.73	0.70	1583
Environment	0.50	0.15	0.22	688
International	0.82	0.58	0.68	824
Opinion	0.67	0.49	0.57	1286
Politics	0.59	0.88	0.71	3174
Science and Technology	0.82	0.55	0.66	463
Sports	0.83	0.85	0.84	1896
Accuracy	0.68	0.68	0.68	15191
Weighted average	0.68	0.68	0.67	15191

5.5 XGBoost

XGBoost classification report is in table (5.5). The confusion matrix of XGBoost is in figure (5.6).

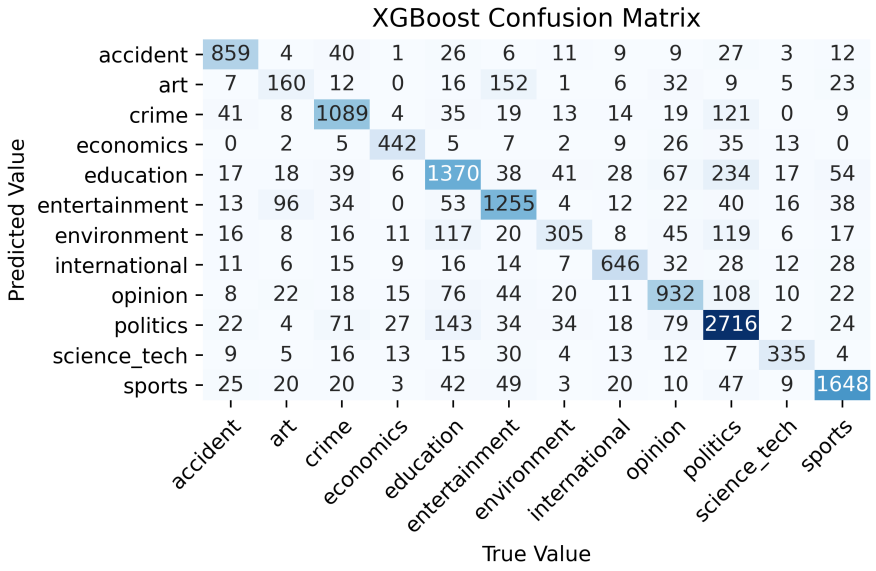


Figure 5.6: XGBoost confusion matrix

Table 5.5: XGBoost classification report

Class	**precision**	**recall**	**f1 score**	**support**
Accident	0.84	0.85	0.84	1007
Art	0.45	0.38	0.41	423
Crime	0.79	0.79	0.79	1372
Economics	0.83	0.81	0.82	546
Education	0.72	0.71	0.71	1929
Entertainment	0.75	0.79	0.77	1583
Environment	0.69	0.44	0.54	688
International	0.81	0.78	0.80	824
Opinion	0.73	0.72	0.73	1286
Politics	0.78	0.86	0.82	3174
Science and Technology	0.78	0.72	0.75	463
Sports	0.88	0.87	0.87	1896
Accuracy	0.77	0.77	0.77	15191
Weighted average	0.77	0.77	0.77	15191

5.6 MLP

We have found that using early stopping increases the performance of MLP classifier. But it also increases the loss (from 0.0992 to 0.6777). As we have prioritized performance over optimization, we are going with early stopping. In figure (5.7), the classification comparison between MLP and MLP without early stopping is shown. Figure (5.8) shows MLP confusion matrix. The classification report of MLP is in table (5.6).

Table 5.6: MLP classification report

Class	**precision**	**recall**	**f1 score**	**support**
Accident	0.84	0.85	0.84	1007
Art	0.55	0.30	0.39	423
Crime	0.79	0.78	0.79	1372
Economics	0.81	0.80	0.80	546
Education	0.69	0.73	0.71	1929
Entertainment	0.74	0.81	0.77	1583
Environment	0.68	0.42	0.52	688
International	0.76	0.81	0.79	824
Opinion	0.73	0.65	0.69	1286
Politics	0.77	0.86	0.81	3174
Science and Technology	0.77	0.71	0.74	463
Sports	0.89	0.87	0.88	1896
Accuracy	0.77	0.77	0.77	15191
Weighted average	0.77	0.77	0.76	15191

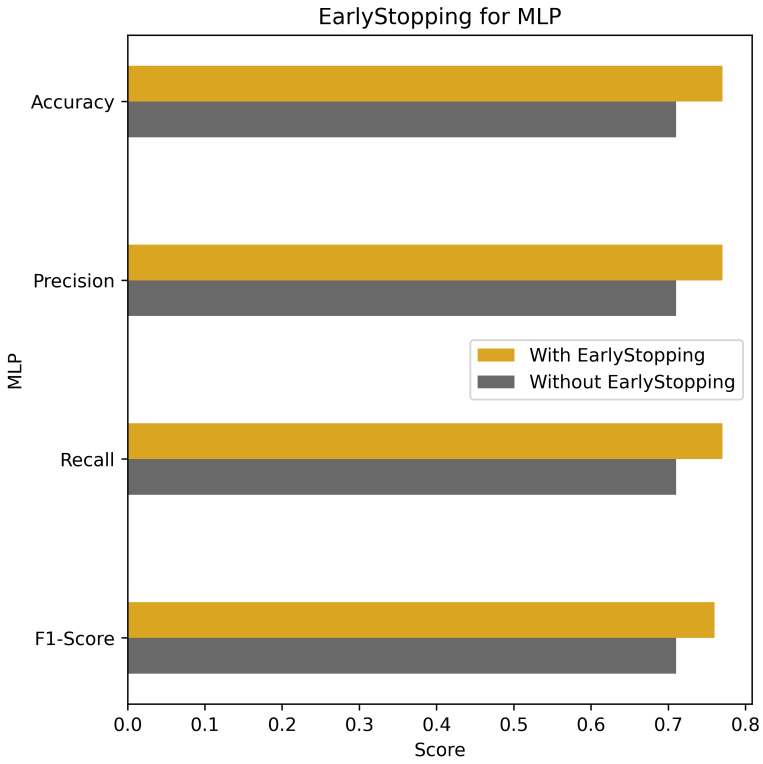


Figure 5.7: MLP early stopping

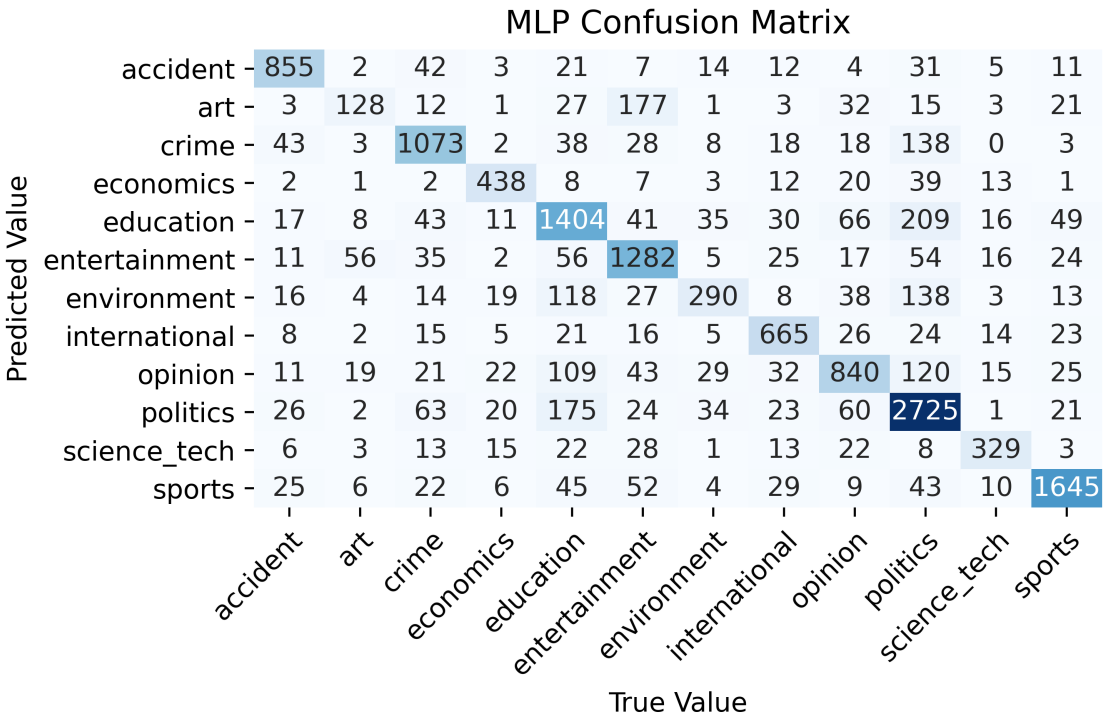


Figure 5.8: MLP confusion matrix

5.7 LSTM

Initially, we had run 50 epochs in our LSTM model. But the model was overfitted. The figure (5.9) shows the overfitting.

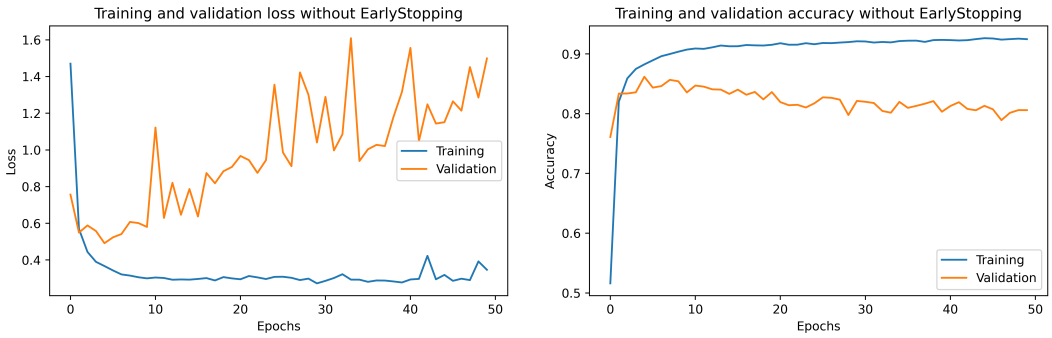


Figure 5.9: LSTM with 50 epochs

For reducing the overfitting, we have used early stopping. It is a callbacks API [37]. We have monitored the validation accuracy. If there is no improvement of validation accuracy over two epochs, the training will be stopped. Weights have been restored from the epoch with the highest validation accuracy. In the below figure (5.10), accuracy and loss of our LSTM model is shown.

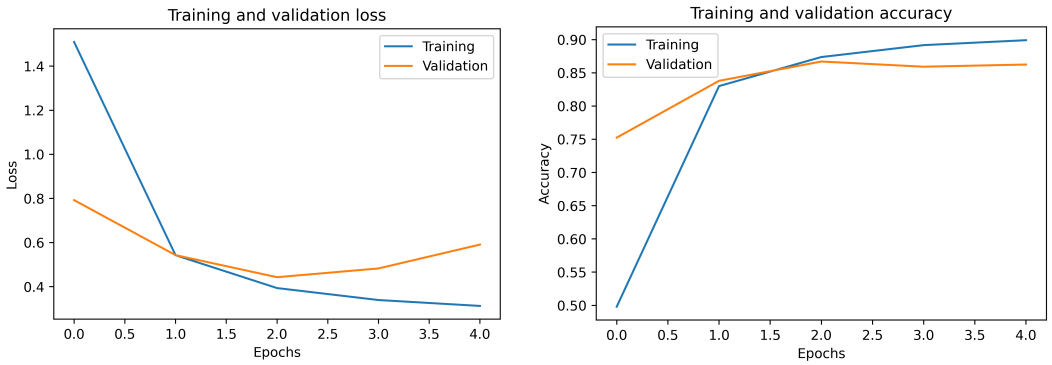


Figure 5.10: LSTM model loss (left) and accuracy (right)

In figure (5.11), we have shown the comparison of the performance and loss between our lstm model and the previous 50 epochs LSTM model. Our model outperforms the old model in every aspect. Table (5.7) shows LSTM classification report and in figure (5.12), LSTM confusion matrix is displayed.

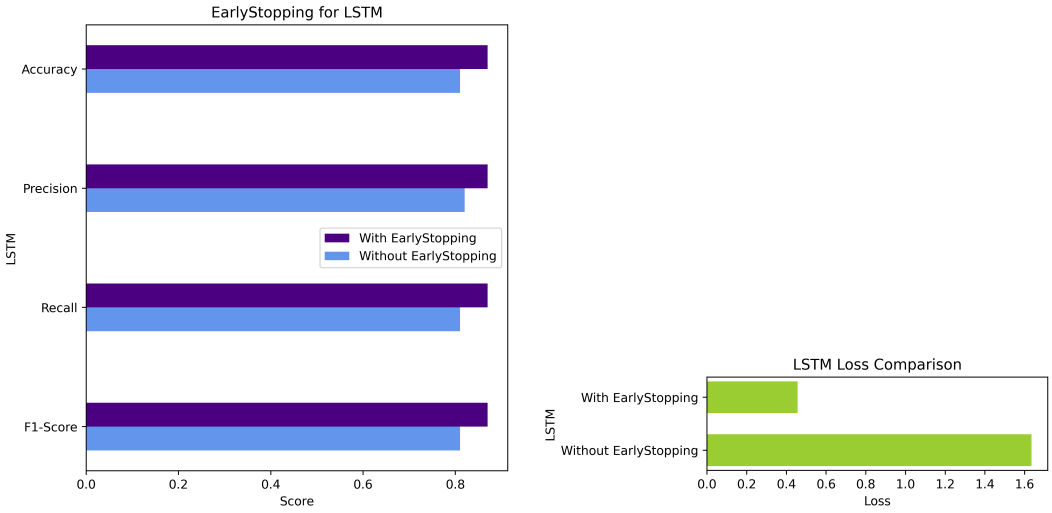


Figure 5.11: LSTM model performance comparison

Table 5.7: LSTM classification report

Class	**precision**	**recall**	**f1 score**	**support**
Accident	0.85	0.92	0.93	1007
Art	0.62	0.61	0.61	423
Crime	0.91	0.86	0.88	1372
Economics	0.89	0.78	0.83	546
Education	0.82	0.90	0.86	1929
Entertainment	0.80	0.93	0.86	1583
Environment	0.74	0.80	0.77	688
International	0.90	0.88	0.89	824
Opinion	0.91	0.75	0.82	1286
Politics	0.88	0.87	0.88	3174
Science and Technology	0.84	0.78	0.81	463
Sports	0.93	0.95	0.94	1896
Accuracy	0.87	0.87	0.87	15191
Weighted average	0.87	0.87	0.87	15191

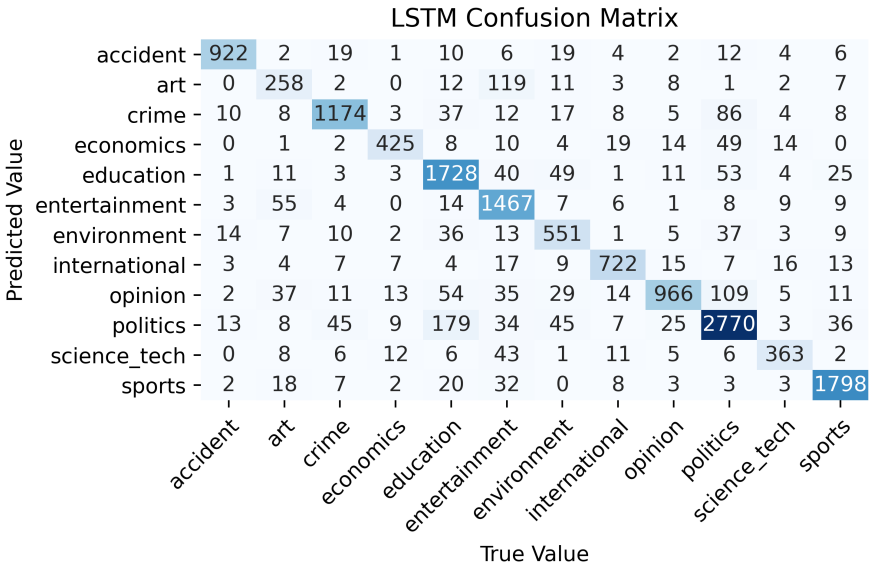


Figure 5.12: LSTM confusion matrix

5.8 Result Summary

In summary, LSTM has given the most accuracy, precision, recall and f1-score (0.87) and Random Forest has given the least accuracy, precision, recall (0.68) and f1-score (0.67). The performance comparison between our used classifiers is shown in below figure (5.13) and in table (5.8).

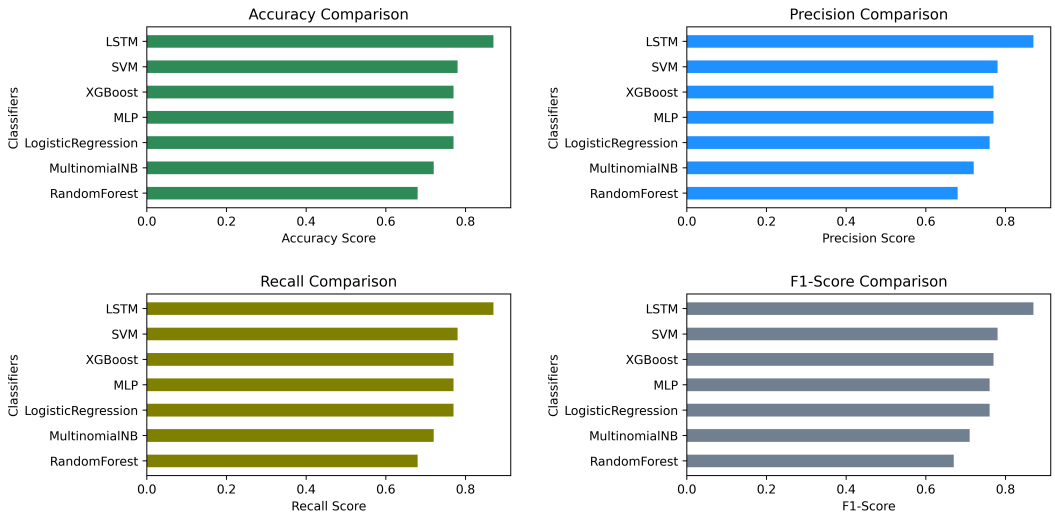


Figure 5.13: Accuracy (upper left), precision (upper right), recall (lower left) and f1-score (lower right) comparison

Table 5.8: Performance comparison

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-score**
LSTM	0.87	0.87	0.87	0.87
SVM	0.78	0.78	0.78	0.78
XGBoost	0.77	0.77	0.77	0.77
MLP	0.77	0.77	0.77	0.76
Logistic Regression	0.77	0.76	0.77	0.76
Multinomial NB	0.72	0.72	0.72	0.71
Random Forest	0.68	0.68	0.68	0.67

5.9 Comparing with similar papers

Table 5.9: Comparing with previous works

Paper Name	**Dataset class**	**Samples**	**Performance**
[18]	5	1000	89.14% (F1-score)
[25]	5	5870	97.3% (Accuracy)
[31]	5	44001	85% (Accuracy)
[19]	9	9127	93.85% (F1-score)
[28]	16	16000	85.0% (F1-score)
In our work	12	75951	87.0% (Accuracy, F1-score)

Table (5.9) shows the comparison of our work with some similar works which are previously done. By comparing with some previous works, we can see that our dataset is quite large from other works and we have more dataset classes than those 3 papers which have better performance than ours. For these reasons, we are considering our research will be well accepted.

Chapter 6

Conclusion

In our work we tried different techniques to get the best outcome from the dataset. Our preprocessing steps includes unnecessary column, duplicate, non Bengali words, punctuation and stop-words removal and then we have done class mapping to numeric values and use TFiDF for feature selection before splitting the dataset. We take words as a feature and use TFiDF for feature selection in all algorithms except LSTM. Instead of it, we use a tokenizer. Among all the algorithms that we used, LSTM gives the best result in all ways. LSTM gives accuracy of 87%. SVM and XGBoost gives most accurate outcomes with compare to other algorithms after LSTM. Both SVM and XGBoost have given 77% accuracy. This scenario is repeated in all other performance matrix as well. The lowest accuracy is given by Random forest, multinomial naive Bias, logistic regression and multilayer perceptron neural network sequentially. We have also found that for making Alpha value 0.01 from default value 1.0, we get around 14% more accurate result in multinomial naive Bias. We also get more accuracy in random forest algorithm while making Bootstrap parameter false. In case of neural network, we get more performance after giving early stopping.

We get outcomes where no others get as much like us. Moreover, there were not much work done before on Bangla text categorization with using both machine learning and deep learning techniques. However, there are many other feature selection methods that were used in other languages that we did not use. Our dataset consists of only news data. There are many other types of data in Bangla such as magazines, literature, nobel, religious and so on. We do not use diversified dataset in our work. For the shortage of having good quality Bangla stemming and lemmatization library we can not work as much as we expect to. But the day is not so far when we get our good Bangla stemming and lemmatization. We hope that we can work on this using several feature selection techniques, diversified dataset with good quality Bangla stemming and lemmatization library for our use.

Bibliography

[1] C. Apté, F. Damerau, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Transactions on Information Systems (TOIS)*, vol. 12, no. 3, pp. 233–251, 1994.

[2] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, Springer, 1998, pp. 137–142.

[3] A. McCallum, K. Nigam, *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, Citeseer, vol. 752, 1998, pp. 41–48.

[4] J. He, A.-H. Tan, and C. L. Tan, "A comparative study on chinese text categorization methods.," in *PRICAI workshop on text and web mining*, Citeseer, vol. 35, 2000.

[5] R. Karim, M. S. Rahman, and M. Z. Iqbal, "Recognition of spoken letters in bangla," in *Proc. 5th international conference on computer and information technology (ICCIT02)*, 2002.

[6] J. Zhang, R. Jin, Y. Yang, and A. Hauptmann, "Modified logistic regression: An approximation to svm and its applications in large-scale text categorization," 2003.

[7] S. C. Hoi, R. Jin, and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 633–642.

[8] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. Khorsheed, and A. Al-Rajeh, "Automatic arabic text classification," 2008.

[9] Q. Ni, Z.-Z. Wang, Q. Han, G. Li, X. Wang, and G. Wang, "Using logistic regression method to predict protein function from protein-protein interaction data," in *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, IEEE, 2009, pp. 1–4.

[10] S. Alsaleem *et al.*, "Automated arabic text categorization using svm and nb.," *Int. Arab. J. e Technol.*, vol. 2, no. 2, pp. 124–128, 2011.

[11] M. Ghiassi, M. Olschimke, B. Moon, and P. Arnaudo, "Automated text classification using a dynamic artificial neural network model," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10 967–10 976, 2012.

[12] A. S. Patil and B. Pawar, "Automated classification of web sites using naive bayesian algorithm," in *Proceedings of the international multiconference of engineers and computer scientists*, Citeseer, vol. 1, 2012, pp. 519–523.

[13] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: Experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[14] M. Hadni, S. A. Ouatik, and A. Lachkar, "Effective arabic stemmer based hybrid approach for arabic text categorization," *International Journal of Data Mining & Knowledge Management Process*, vol. 3, no. 4, p. 1, 2013.

[15] V. Bijalwan, V. Kumar, P. Kumari, and J. Pascual, "Knn based machine learning approach for text and document mining," *International Journal of Database Theory and Application*, vol. 7, no. 1, pp. 61–70, 2014.

[16] A. Bouaziz, C. Dartigues-Pallez, C. da Costa Pereira, F. Precioso, and P. Lloret, "Short text classification using semantic random forest," in *International Conference on Data Warehousing and Knowledge Discovery*, Springer, 2014, pp. 288–299.

[17] A. Defazio, F. Bach, and S. Lacoste-Julien, "Saga: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in neural information processing systems*, 2014, pp. 1646–1654.

[18] A. K. Mandal and R. Sen, "Supervised learning methods for bangla web document categorization," *arXiv preprint arXiv:1410.2045*, 2014.

[19] F. Kabir, S. Siddique, M. R. A. Kotwal, and M. N. Huda, "Bangla text document categorization using stochastic gradient descent (sgd) classifier," in *2015 International Conference on Cognitive Computing and Information Processing (CCIP)*, IEEE, 2015, pp. 1–4.

[20] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[21] G. Chen, "A gentle tutorial of recurrent neural network with error backpropagation," *arXiv preprint arXiv:1610.02583*, 2016.

[22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[23] G. Al-Naymat, M. Al-Kasassbeh, N. Abu-Samhadanh, and S. Sakr, "Classification of voip and non-voip traffic using machine learning approaches.," *Journal of Theoretical & Applied Information Technology*, 2016.

[24] C. Chavaltada, K. Pasupa, and D. R. Hardoon, "A comparative study of machine learning techniques for automatic product categorisation," in *International Symposium on Neural Networks*, Springer, 2017, pp. 10–17.

[25] S. Al Mostakim, F. Ehsan, S. M. Hasan, S. Islam, and S. Shatabda, "Bangla content categorization using text based supervised learning methods," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, IEEE, 2018, pp. 1–6.

[26] A. Dhar, H. Mukherjee, N. S. Dash, and K. Roy, "Performance of classifiers in bangla text categorization," in *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, IEEE, 2018, pp. 168–173.

[27] I. Rasheed, V. Gupta, H. Banka, and C. Kumar, "Urdu text classification: A comparative study using machine learning techniques," in *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, IEEE, 2018, pp. 274–278.

[28] M. Chakraborty and M. N. Huda, "Bangla document categorisation using multilayer dense neural network with tf-idf," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, IEEE, 2019, pp. 1–4.

[29] M. Kabir, M. M. J. Kabir, S. Xu, and B. Badhon, "An empirical research on sentiment analysis using machine learning approaches," *International Journal of Computers and Applications*, pp. 1–9, 2019.

[30] A. Khatun, A. Rahman, and M. S. Islam, *Bangla news dataset*, Dec. 2019. [Online]. Available: https://data.mendeley.com/datasets/xp92jxr8wn/2?fbclid=IwAR3MlZ6zDTVE0_YHp29tyO3XlEhjENowrDnxVZNVGJUAejRW9XnTqP3FRzY.

[31] M. F. Ahmed, Z. Mahmud, Z. T. Biash, A. A. N. Ryen, A. Hossain, and F. B. Ashraf, "Cyberbullying detection using deep neural network from social media comments in bangla language," *arXiv preprint arXiv:2106.04506*, 2021.

[32] Baeldung, *Multiclass classification using support vector machines*, Aug. 2021. [Online]. Available: https://www.baeldung.com/cs/svm-multiclass-classification.

[33] A. Hachcham, *Xgboost: Everything you need to know*, Aug. 2021. [Online]. Available: https://neptune.ai/blog/xgboost-everything-you-need-to-know.

[34] H. Ampadu, *Random forests understanding*. [Online]. Available: https://ai-pool.com/a/s/random-forests-understanding.

[35] *Bnlp-toolkit*. [Online]. Available: https://pypi.org/project/bnlp-toolkit/.

[36] *Logistic function*. [Online]. Available: https://andymath.com/logistic-function/.

[37] *Tf.keras.callbacks.earlystopping nbsp;: nbsp; tensorflow core v2.6.0*. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping.