Bangla Document Categorisation using Multilayer Dense Neural Network with TF-IDF

Manisha Chakraborty

Department of Computer Science and Engineering

United International University

A thesis submitted for the degree of

MSc in Computer Science & Engineering

November 2019

Abstract

Document categorisation is a quintessential example of a natural language processing quest which includes sorting documents by their content into one or more predefined classes. This thesis proposes a model which consists of multilayer Dense Neural Network with Term Frequency - Inverse Document Frequency (TF-IDF) as feature selection technique in terms of Bangla text document categorisation. This proposed system is divided into three consecutive steps: i) preprocessing raw text data and extracting feature using TF- IDF, ii) designing the model architecture and fitting the model to training set, and iii) evaluating model performance on test set by measuring accuracy and weighted average of F1-score. It is observed from experiments that the proposed method exhibits higher accuracy (85.208%) and weighted F1 score (0.85) compared to the other well-known classification algorithms (K Nearest Neighbor, Decision Tree, Support Vector Machine, Stochastic Gradient Descent, Multinomial Naïve Bayes, and Logistic Regression) for Bangla text document classification.

Published Papers

Work relating to the research presented in this thesis has been published by the author in the following peer-reviewed conference:

1. Manisha Chakraborty, and Mohammad Nurul Huda, “Bangla Document Categorization using Multilayer Dense Neural Network with TF-IDF”, *International Conference on Advances in Science, Engineering Robotics Technology (ICASERT 2019)*, May 3-5, 2019, Dhaka, Bangladesh, pp. 1-4.

Acknowledgements

I would like to convey my thankfulness toward all those who have aided me with their help for completing my MSc in Computer Science and Engineering Thesis. Firstly, I am thankful to my supervisor Dr. Mohammad Nurul Huda, Professor and MSCSE Director, Department of Computer Science and Engineering at United International University for his constant guidance and insight throughout the thesis work. I am also grateful to Head Examiner Dr. Swakkhar Shatabda, Associate Professor, Department of Computer Science and Engineering at United International University, for reviewing the thesis book and his helpful suggestion on thesis book. I would like to thank Dr. Dewan Md. Farid, Associate Professor, Department of Computer Science and Engineering at United International University, for reviewing my thesis book and his valuable remarks. I also thank Rubaiya Rahtin Khan, Assistant Professor, Department of Computer Science and Engineering at United International University for her valuable time to review my thesis book. Lastly, I would like to convey my gratitude toward those individuals who have supported me throughout the process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides a descriptive viewpoint of the introductory aspects of thesis work which includes problem statement of the work, objective and motivation behind it. It also includes a brief portrayal of the experiments carried out in this thesis. Furthermore, thesis contribution is discussed and finally this chapter concludes with book organization which gives an outline for the rest of the book.

1.1 Introduction

Document categorization, a classification problem by nature, is an interesting research aspect of natural language processing (NLP) involving a broad range of real-life applications such as sentiment analysis, spam detection, review rating prediction, e-commerce, online library management, and so on. The task of automatic document categorisation is defined by assorting documents into one or more pre-determined classes by integrating computer program on the basis of the content of those documents. This task can also be manoeuvred to analyse hierarchically structured datasets [1] and multiple label containing datasets [2]. The rapid growth of internet activity and exponentially rising social media usage has not only necessitated but also implied this task of great importance. On the subject of document classification tasks, supervised classification algorithms have been a popular choice among NLP practitioners [3], [4]. Furthermore, neural networks (NN) with word vectors have accomplished remarkable results regarding text classifications in recent times [5]. However, myriads of studies have been carried out on English in contrast to Bengali. Bangla, also known as Bengali, is one of the most widely spoken language throughout the world. Hence, with a gigantic amount of internet users who use Bangla as their native language, almost every form of digital text documents such as blogposts, newspaper articles, social media posts are noticeably escalating all over the web. Subsequently, the indispensability of automated Bangla document categorisation becomes a critical issue to address. Various methods of Bangla document classification have been proposed in recent studies. Henceforth, in this thesis, a system consisting Dense Neural Network model with Term Frequency-Inverse Document Frequency (TF-IDF) as feature selection technique is designed aiming to classify Bangla electric text documents. However, a comparative analysis among the proposed model and other well-established methods can aid to the better comprehension of model's applicability in terms of Bangla document categorization. Therefore, nine experiments are designed with two different feature selection method (TF-IDF and word embedding) for experimentation and performance evaluation in this thesis.

1.2 Problem Statement

The increased popularity of internet usage in people who exercise Bangla on a daily basis has led to the expansion of Bangla text documents on the web. Moreover, in online libraries which contain books of various genres, different kinds of books are added to the collection in large quantity. Manual sortation of these documents is laborious and expensive approach to implement. Alongside, storage capacity of computers has upgraded radically and the ease of data storing and retrieving has increased manifold over time. However, this amelioration in volume of text data can be proven beneficial to the experts in various fields if categorized according to task-specificity. Earlier approach of tackling textual information was to analyze and process text data by domain specific experts. Nonetheless, the constant increment in ever-so massive amount of text data has made it almost impossible to rely upon the earlier manual approach of categorization. Furthermore, toxic comments and contents has become an integral part of social media with the increasing use of social networking platforms such as Facebook, Twitter and so on. Continuous inspection of contents prior to posting on social media should be the course of action, however, human scrutinizing of every content is a daunting task. Document classification problem has multiple applicability and bringing automation in this task can address variety of issues revolving this problem. Thus, automatic document classification is a task of great importance and a solve for various tasks in modern days.

1.3 Motivation

Categorising documents in automated approach has become a prominent task in the field of natural language processing that achieved much popularity over the recent years due to the skyrocketing accretion of textual information. Applicability is one of the important facet to opt for the task of automatic Bangla text documents categorisation, since distinction among text data of different fields is necessitated in various domain of expertise such as education, healthcare, law, consumer goods, and the list is unending. Furthermore, text data of numerous field is also accruing due to the increasing internet usage among native Bengali people. Apart from applicability, in terms of choosing algorithms for the task, handful number of methods have been comprised of well-known supervised algorithms. However, applying neural networks in case of Bangla document categorisation is less of a common practice. There has been some recent works conducted incorporating neural network [6], [7] but there is no established comparison among all of the well-known and widely used classification algorithms. Besides, most of these experiments are conducted on large dataset but in the matter of new real-world projects making a fresh start, there might be fewer number of training examples [8]. Hence, performance evaluation on small scale dataset with existing methods should be considered as essential as on the large dataset.

1.4 Objectives of the Thesis

This thesis aims to incorporate a method that categorizes electronic copy of Bangla text documents. For this purpose, previously labeled documents are collected from two online Bangla newspaper (Prothom Alo and Bdnews24). These documents are divided intro train set and test set and train set is used in variety of experiments including the proposed model. After experimentation, comparison among the result of different experiments are conducted for model evaluation and prediction is made on test set.

1.5 Brief Methodology

Six classification algorithms namely K Nearest Neighbour, Decision Tree, Support Vector Machine, Stochastic Gradient Descent, Multinomial Naïve Bayes, Logistic Regression combined with 10-fold cross validation and grid search, and three different neural network architectures with 10-fold cross validation are experimented for Bangla text document classification. After comparing all the experiments, experimental results showed that our proposed method performed better than other experimented models.

1.6 Thesis Contributions

Contributions of this thesis work is outlined in brief as follows:

- This thesis includes an extensive literature review on text classification task. Literature review is arranged on the basis of varying languages, different data sources, manifold feature selection techniques, miscellaneous algorithm implementations and various evaluation measurement techniques.

- We have created a Bangla text dataset from two popular online newspaper known as Prothom-Alo and Bdnews24 containing 16 categories where each category consists of 100 articles.

- We have designed nine experiments with six supervised classification algorithms namely K Nearest Neighbor, Decision Tree, Support Vector Machine, Stochastic Gradient Descent, Multinomial Naïve Bayes, Logistic Regression and three different neural network architectures: one is a Dense Neural Network with TF-IDF as Feature extractor, other two incorporates embedding layer as part of architecture where one is a Convolutional Neural Network and another one is a Dense Neural Network.

- Hyper parameters are optimised for traditional classification algorithms through grid searching.

- We have included 10-fold cross validation in all the experiments conducted in this thesis for estimation purpose.

- The performance of all nine experiments are evaluated on the basis of accuracy, weighted average scores of precision, recall and F1-measure. Experimental results depicted that our proposed method outperformed the other two Neural Network architectures and other traditional classification algorithms.

1.7 Organization of the Thesis

The rest of this thesis is distributed into the following chapters:

Chapter 2 presents background study and literature review required for this thesis.

Chapter 3 provides methodology of the proposed model and other experiments conducted in this thesis.

Chapter 4 empirical analysis of the experiments is described.

Chapter 5 conclusion and future work of this thesis is depicted.

Chapter 2

Background and Literature Review

Reviewing existing literatures provide substantial knowledge on the topic of interest offering guidance to follow throughout the thesis work. This chapter discusses background and literature review of the work conducted and provides an insight into the necessary preliminaries of this thesis.

2.1 Preliminaries

Language, in general, seems difficult to learn as the tenancy of its complex characteristics and evolutionary change, yet we human beings perceive and utilise it since early childhood to communicate with our family and other persons. Language has been an essential part of our community as a tool of communication since the dawn of civilisation. However, our communication these days not only includes oral and written form of language usage, but also constitutes thousands of electronic texts that has become integral part of our daily life like online newspaper, social media, e-books, etc., and for computer to analyse the complicated structure of natural language and providing useful outcome is always a challenging task. This task requires the implementation of the preliminary knowledge of various mathematical appliances which are also experimented in this thesis. In this chapter, these introductory topics will be discussed for ease and coherence.

2.1.1 Feature Selection Methods

Feature selection method is an indispensable aspect for classifying tasks. This section describes feature selection techniques applied in this thesis work briefly.

2.1.1.1 Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency - Inverse Document Frequency, shortly known as TF-IDF is a statistical weight measurement which is widely used in text mining tasks. TF-IDF is often opted as a tool for feature extraction in a variety of Natural Language Processing (NLP) tasks or text mining tasks [9],[10]. The TF-IDF value denotes the significance of a word in a document or a collection of documents called corpus. Term Frequency(TF) indicates the frequency of a term t occurring in a document d. Inverse Document Frequency (IDF) provides information about the importance of a term t. The TF-IDF weight of a term i can be calculated by the following equation:

$$w_i = (TF_i \times \ log\,(N \div n_i)) \div \sqrt{\sum_{i=1}^{n} (TF_i \times \ log\,(N \div n_i))^2}$$

2.1.1.2 Word Embedding

Word embedding is a form of vector representation of words where similar words have similar encoding in the vector space. This method learns vector representation from a text corpus for a prior fixed-size vocabulary. For this experiment, we used embedding layer in the neural network which learns the vector representations of the words jointly with the model in the training process. Including embedding layer as a part of the network helps the network to learn the useful combination of the word vectors from the input which might play a role in prediction [11]. These kind of word vectors are also known as distributional similarity based word representations. The embedding layer, also known as lookup layer, provides a matrix E from words tokens. The matrix E can be described by the following equation:

$$E \in \mathbb{R}^{|vocab \times d|}$$

where each row is correlated with a different word from vocabulary. Afterwards, lookup operation is conducted by indexing word to the matrix. Embedding layer produces a Continuous Bag of Words (CBOW) feature representation , where matrix E is the embedding matrix [11]. In case of Continuous Bag of Words (CBOW), for a given target word, context can be represented by various words.The CBOW approach is comparable to traditional bag of words approach.

2.1.2 Algorithms

Algorithm selection is an integral part of machine learning algorithms. In this section, a brief description of all the algorithms selected for this thesis is provided as follows.

2.1.2.1 Support Vector Machine

Support vector machine algorithm is well-suited for text classification tasks [3], [12]. A Support Vector Machine (SVM) is a classifier that distinguishes between data by a separating hyperplane. In other words, when trained on labeled training dataset in supervised approach, it yields an optimal hyperplane that groups new examples on testing dataset. If considered a two dimensional feature space, the hyperplane becomes a line that divides the plane into two regions, each region signifying a class which is the essence of a binary classification task. In support vector machine, linear kernel is used for linearly separable data while polynomial and radial basis function (RBF) kernels are applied to linearly inseparable data. There are two essential parameters for implementing Support Vector Machines: parameter c handles the trade-off between the size of hyperplane and correct classification of training points, and gamma parameter regulates the curvature of decision boundary.

2.1.2.2 Stochastic Gradient descent

Stochastic gradient descent, often abbreviated as SGD, is a significant optimisation technique in terms of machine learning. It is iterative by nature, which optimises an objective function equipped with the parameters of a model and updates parameters for each training sample. From [13], it can be stated that SGD works faster than batch gradient descent since batch gradient descent does redundant computation in case of large datasets by re-calculating gradients before each parameter update and SGD escapes this redundancy by executing one update at a time. Thus, SGD optimisation technique can be particularly helpful in big data applications since it works faster with reduced computational load. In scikit-learn library used in python programming language, a model can be selected by changing loss parameter to employ the SGD optimisation technique. By default, the value of loss parameter is 'hinge' and it fits a linear support vector machine with SGD optimisation technique. Log loss provides a logistic regression with SGD training. Alpha is a regularisation parameter used in SGD approach.

2.1.2.3 Logistic Regression

Logistic Regression is a classification algorithm which is widely known for its ability to work with categorical data. Logistic regression's prediction is highly dependent on the categorical variable as it predicts probability of an outcome, $P(Y = 1)$ as a function of input data, X. It constructs a logistic curve providing values between 0 and 1. Logistic Regression is broadly exercised in case of binary response data in data modelling such as spam detection, positive/negative movie review analysis, tumour malignancy detection, etc. It is a part of a group of models called generalised linear models. Logistic regression and linear regression are almost similar except for the fact that the curve constructed by logistic regression uses the natural log-odds of target variable. There are three types of logistic regression: binomial, ordinal and multinomial. In binomial or binary logistic regression, there are only two possible types of outcome. Multinomial logistic regression is applied where three or more categories are possible that are not ordered. Ordinal logistic regression works well with ordered dependent variables. Figure-2.1 illustrates an example of a simple binary logistic regression.

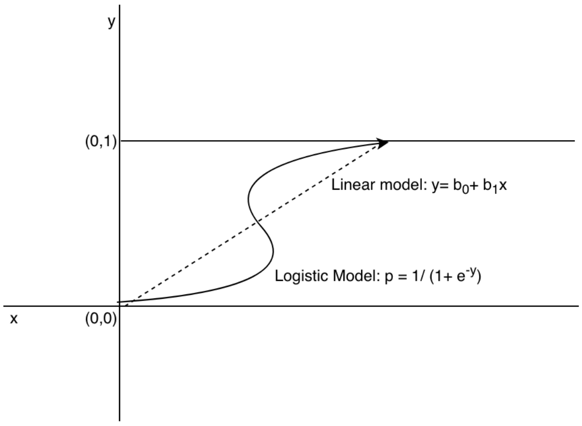

y
(0,1)
Linear model: $y= b_0+ b_1x$
Logistic Model: $p = 1/ (1+ e^{-y})$
x
(0,0)

Figure 2.1: Logistic Curve - Comparison Between Linear model and Logistic model

In scikit-learn library of python programming language, if the value of ‘multi_class’ parameter is ’multinomial’ then multinomial loss is calculated even when data is binary. The ’ovr’ value of ’multi_class’ treats the problem as binary problem by incorporating ‘One vs Rest’ scheme.

2.1.2.4 Naïve Bayes

Naïve Bayes algorithm is based on Bayes’ theorem which presumes attribute independence [14]. Although the base assumption of conditional independence rarely holds truth in terms of real world applications, its driven performance in classification tasks is quite astounding. Naïve Bayes is reviewed comprehensively by several researchers in terms of text classification tasks [15],[16].

Text classification from Naïve Bayes perspective necessitates the assumption that text data was manufactured by a parametric model and this approach uses training data to compute Bayes-optimal approximates of the model parameters. Afterwards, based on these estimates of model parameters, this method uses Bayes’ rule to calculate posterior probability of new test documents. Subsequently, it selects the most probable class for classifying new documents. The Bayes’ rule is provided in the following equation:

$$P(A|B) = \frac{P(B|A)P(A))}{P(B)}$$

where $P(A|B)$ is the posterior probability that event A occurs when given event B. $P(B|A)$ is the likelihood probability of occurring event B when event A is given. $P(B)$ is the probability of event B happening independently.

In [15], multinomial Naïve Bayes model is compare with Multi-variate Bernoulli Naïve Bayes to contrast the efficiency of different Naïve Bayes models in support of text classification. In this study, the multinomial model offered a 27% decline in error over the multi-variate Bernoulli on average across the experimented datasets. From studies [15] and [16], both indicate that Naïve Bayes classifier with multinomial event model usually outperforms the other variants of Naïve Bayes models. Parameters of Multinomial Naïve Bayes model includes alpha which is Laplace smoothing parameter.

2.1.2.5 K-Nearest Neighbor

K-Nearest Neighbor, shortly known as KNN, is a well-known statistical pattern recognition algorithm which has been widely used in various text classification tasks. This method maneuvers classification tasks by categorising objects on the basis of nearest neighbors or training samples in the feature space. In this algorithm, given a test document, it locates k nearest neighbors from the training examples and uses categories of the k neighbors to determine the category candidates. The weight of the categories of the neighbors is regulated by the similarity score of each neighbor document to the test document. If some of the neighbors share same category, then weight of each neighbor is added for that category and the resulted sum is used as the prospective score of that category for the test document. Next a sorted and ranked list of scores of different categories is created for the test document. By including a threshold value on these scores, category assignment is attained. The decision rule for this algorithm is provided in the following equation:

$$y(\vec{x}, c_j) = \sum_{\vec{d_i} \in kNN} sim(\vec{x}, \vec{d_i}) y(\vec{d_i}, c_j) - b_j$$

where, $y(\vec{d_i}, c_j) \in \{0, 1\}$ is the classification of document $\vec{d_i}$ in respect of class c_j. $sim(\vec{x}, \vec{d_i})$ is similarity function between training text $\vec{d_i}$ and testing text $\vec{x}$. b_j is the threshold value for decision making.

Parameters of KNN includes 'n_neighbors' which indicates number of neighbors and 'weights' parameter indicates weight function used for evaluation. One particular drawback in terms of applying KNN is the struggle to elect optimal values. The best pick for k value is usually varies with data. Larger k value can reduce the influence of noise on classification, however, it can formulate less distinct boundaries among classes.

2.1.2.6 Decision Tree

Decision Tree is a supervised learning algorithm commonly used for classification and regression tasks. It forecasts target value on the basis of simple decision rules from the extracted features of dataset. Being a commonly opted supervised learning algorithm, Decision Tree based methods are able to achieve higher accuracy with stability and offers simplicity in interpretation. These methods also represent non-linear associations suitably which is unlikely of linear models. In [4], Decision Tree and PropBayes algorithm is compared to measure the approximate conditional probabilities of category occurrence given feature occurrences. According to this study, PropBayes does not capture large amount of feature efficiently and also unable to achieve good precision with high recall and few number of features. The consequence of combining high recall with small feature set is putting high value on a single feature, which often results into unreliable classification. Unlike PropBayes, Decision Tree can select feature stepwise which facilitates the usage of more features than PropBayes resulting more reliable classification, nonetheless, requiring more training examples.

2.1.2.7 Neural Network

The concept of neural network is inspired from human brain and the way it processes information. Neural networks are known for their extraordinary aptitude in obtaining useful material from complicated or imprecise data and often used to detect patterns which are too intricate for human or other computing methods to perceive. A neural network usually consists of an input layer, one or more hidden layer and an output layer where each layer consists of nodes or units. It is able to map training samples from input layer to the output layer. For an input unit or node, the given inputs are multiplied by the weights of that unit and added as a sum. This summation value is known as the summed activation of the unit. Afterwards, this summed value is transformed by an activation function which is the output for the individual input node, also referred to as activation of the node. However, only the input layer and hidden layer contains activation function since output layer is commonly taken to embody the class scores in classification tasks. In our experiments, we used different types of layer to construct three different neural network. These layers and their usage is described in the following paragraphs.

Fully connected layer Fully-connected layer, also known as densely connected layer is the most common form of neural network layer. It is also known as multilayer perceptron. Networks created with only this layer are fully connected pairwise, however, neurons which reside in a single layer shares no connection.

Dropout layer Dropout layer averts overfitting and offers a way to merge various architectures of neural network competently [17]. The term "dropout" indicates temporary removal of neural network units from both hidden and visible layers alongside all of its incoming and outgoing connectivities. Dropout has proven to be a good regularizer even using a larger neural network as it steadily added 2%-4% to the comparative performance and a dropout rate of 0.5 has been demonstrated to be efficient in the experiments conducted in [5].

Convolution layer While convolutional Neural Network(CNN) has been successfully involved in various benchmark studies for image recognition tasks [18], [19], it has also in and proven to be applicable for several Natural Language Processing(NLP) tasks such as semantic parsing [20], sentence modeling [21], sentence classification [5], character-level text classification [22]. In convolution layer, convolution is used over the input where each local region of input is connected to a neuron of next layer. This spatially extended connectivity is a hyper-parameter known as filter of the convolution. Each layer applies different filters and combines their result. For NLP tasks, one dimensional convolution layers are used generally. Input is generally a matrix of vector representations of text documents, each row corresponding to a word or a character. Afterwards, filters are used which slides over the full rows of the matrix (words or characters). Thus, the width of the filter is same as the width of the input matrix. Height of the filter, also known as region size or kernel size, usually specifies the length of one dimensional convolution window. Stride refers to the step size of each shift taken by the filter. Input size is fixed with zero padding for regulating spatial size of output.

Spatial dropout layer Spatial dropout is an alternative approach of incorporating dropout layer with CNN. In SpatialDropout, value of dropout is extended across the feature space. In [18], authors initially experimented standard dropout with convolution layer and observed that applying standard dropout before each convolution raised training time, however, did not inhibit overtraining. The authors initialized spatial dropout layer which drops out values across entire feature map and observed that this modified dropout improved performance of their method.

Pooling layer Pooling layers are usually employed after the convolutional layers. Applying a max operation to each filter's outcome is a common method of executing pooling. In Natural Language Processing(NLP), generally, pooling is applied over the whole output volume, capturing a single value for each filter, which is known as 1-max pooling. [23] implies that 1-max-pooling performs better than average-pooling and k-max pooling.

Activation functions The activation functions, also known as transfer functions, regulate the output of a neural network. This function is connected to each neuron of all the layers of a neural network (except for output layer), determining the activation of the neuron. Moreover, activation function normalizes the output of a neuron between a range of 0 to 1 or -1 to 1.

a) Rectified Linear Unit (RELU) The rectified linear activation function, otherwise known as RELU, is a piecewise linear function that outputs zero if the value of input is negative, otherwise output value is the same as input value. It has become the default activation function that is widely used in varieties of neural network architectures since it equips model with ease at training and often acquires excellent performance.

b) Sigmoid Sigmoid activation function provides smooth gradient by inhibiting fluctuations in output values. It normalizes each neuron's output value that range between 0 and 1. The sigmoid function used for the activation is given below:

$$y = \frac{1}{1 + e^{-x}}$$

In the equation above, x is the input and y is the output for sigmoid function.

c) Softmax Softmax activation function is capable of handling more than two classes or categories. This is why Softmax activation function is used for multi-class classification tasks on the output layer of the neural network since it provides the probability distribution for all classes to categorise from all the possible target classes.

2.1.3 Evaluation Metrics

Performance evaluation metrics that are used in this study are described in brief as follows:

Confusion Matrix Confusion matrix depicts the performance of a classification model on test data. It is essential for computing other evaluation metrics such as accuracy, precision, recall and F1-score. In Table -2.1, structure of a confusion matrix is provided.

hdtp

Table 2.1: Confusion matrix

	Predicted class		
Actual class		Class=Yes	Class=No
	Class=Yes	True Positive(TP)	False Positive(FP)
	Class=No	True Negetive(TN)	False Negetive(FN)

- True Positives (TP) – True positives provides the number of correctly predicted positive values.
- True Negatives (TN) - True Negatives (TN) provides correctly predicted negative values.
- False Positives (FP) – False Positives occurs when actual class is negative and predicted class is positive.
- False Negatives (FN) – False Negatives happens when actual class is positive but predicted class in negative.

Precision Precision is the ratio of True Positives(TP) to the total predicted positive records (TP + FP). It provides information about the proportion of the data points are relevant according to the model, are actually relevant.

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

Recall Recall is the ratio of True Positives(TP) to the all observations in actual positive class. Recall has the ability to find all relevant instances in a dataset.

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

Accuracy Accuracy is the most commonly used performance measure and it is a ratio of correctly predicted observations to the total observations. Accuracy can provide information if a model is being trained appropriately and how will be the performance of it generally. However, it does not do well when you have a severe class imbalance.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

F1-measure F1-score is the harmonic mean of Precision and Recall. Therefore, this score considers both false positives and false negatives. Usually F1-score is more useful than accuracy, especially in case of uneven class distribution.

$$F1Score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

2.2 Literature Review

Understanding natural language in text or any other form might be an easy task for human beings; however, scrutinising the structure of a language, describing the underlying concept and applying these intricacies to address task specific solutions is a tricky endeavour for computers. Nonetheless, diverse methods have been incorporated and achieved extraordinary outcome regarding miscellaneous language processing tasks over period of time; document categorisation being one of them. Document categorisation refers to a process of classifying unlabelled documents into one or more predefined classes depending on their contents. It is a supervised classification task and entails feature extraction method to leverage different aspects of textual information. In this section, literature review is described based on various aspects such as language, data source, feature selection techniques, algorithms and results.

2.2.1 Language

Text document classification or categorization is a common natural language processing task conducted on numerous languages with varying methods. However, different languages necessitate contrasting techniques in preprocessing due to the distinction among the complex structures and characteristics of various languages. Considering the fact that different languages consist of unique sets of character, particular grammar rules and separate set of punctuation symbols; it is certain that preprocessing varies from one language to another. Apart from the differences in core structure of various languages, inadequate amount of work is another reason that makes the preprocessing much more difficult due to the lack of sufficient preprocessing knowledge on target language.

On the subject of the amount of existing works, ample research works are carried out on English language compared to any other languages. Since most of the benchmark studies are conducted on English language, preprocessing in English is well-studied and advanced preprocessing tools are centered towards English language. Generally, this task includes removing extra meaningless characters, removing words that occurs commonly in most of the documents since they have less importance in categorization – these words are known as stopwords.

For English language classification tasks, in paper [3], dataset used in the experiment is hand-labeled where some text data are categorized with more than one label. Removing stopword and disregarding the terms occurring in less than three document is also done as part of preprocessing. Similarly, manual category assignment can also be noticed in paper[4], [24]. In paper [25], the authors collected their English text data from Yahoo newsgroup in HTML form; also manually indexed by human experts. Their preprocessing includes the removal of HTML header and tags, synonymously known as document parsing and removal of stop-word and low frequency words based on term weighting and principal component analysis (PCA). Moreover, while most of the studies are conducted concerning word level categorisation in English; however, in recent years, character level English text categorisation has become an imminent interest among natural language processing(NLP) researchers. The experiments conducted in [26] confirms that character level text categorisation is a definite possibility for English text categorisation without necessitating words. Furthermore, a study of short text classification is carried out where English text is engineered as sequential data using convolutional neural network based and recurrent neural network based representations [27]. The reason to consider text as sequential data relys upon the fact that short texts are usually part of sentences in documents or dialogues [27].

Notwithstanding the fact that English texts are the most extensively studied in natural language processing fields, existing research works are present for various Asian languages. In paper [28], authors worked on Marathi text document categorization with various supervised learning algorithms such as Naïve Bayes classifier, Modified K nearest neighbor classifier and Support Vector Machine. In paper [29], task of classification is conducted on Gujarati language where authors compared the effect of Naïve Bayes Classifier on Guajarati text classification without and with using feature selection method. In paper [30], authors used combination of Naïve Bayes Classifier and ontology based classification approach to categorize Punjabi text documents. There are some text categorization research work exists for Arabic languages [31],[32].

There are also existent research works which compares more than one language for performance evaluation. For instance, in paper [33], authors used Indonesian and English twitter data for analysing character and word level text classification. Both datasets are preprocessed beforehand by removing hyperlinks and lowercasing characters in the dataset. However, Chinese and English language datasets are more commonly contrasted among various experiments [34],[26],[35].

In the matter of Bangla document categorisation, various methods have been experimented over recent years. Preprocessing task is different despite all dataset being in Bangla language due to the variation in data sources, varying task requirements. However, the objective of preprocessing is to prepare text data to be accessible by feature extraction tools. Thus, this task includes removal of stop words, punctuations, English letters, English and Bangla numeric letters since these are frequently used in Bangla text [36]. In paper[37], authors proposed a Bangla document categorisation system using Term Frequency- Inverse Document Frequency (TF-IDF) as feature selection technique and Stochastic Gradient Descent(SGD) as classifier. Pre-labeled dataset is collected for this study from an online newspaper named BDNews24 where each document is preprocessed. In paper[38], supervised learning methods are assigned and contrasted in terms of classifying Bangla text documents. Authors collected dataset from online newspapers known as Prothom Alo and Bdnews24 and preprocessed these documents by removing punctuation symbols, unnecessary words, and stemming the words into root words. Preprocessing task also includes tokenization. Text documents generally comprises of sequences containing sentences, words or letters necessitating segmentation in order to enact the usability of feature extraction methods. This segmentation can be executed on the basis on sentence, word or character. The process of segmenting documents into usable units is known as tokenization [39], [40]. For Bangla documents, ‘space’ is commonly used as delimiter for tokenization as it is noticable in [39], [41], [36], [42], [43],[40]. Moreover, word stemming and removal of insignificant words or stopwords like conjunctions, pronouns is also included in various studies [44], [6], [45], [46], [47], [40]. However, in [7], tokenization process includes fixing the size of each text data after segmentation and padding is added to retain fixed size of 2200 if input text inhibits lesser words.

2.2.2 Data Source

Text categorization has numerous real-life application ranging from sentiment analysis to spam detection. This is the incentive behind varying data resources as data can be amassed from countless sources to comply with target application. In case of English document categorization, Reuters-21578 corpus is extensively used [3], [24], [48], [35], [49],[50], [51], [52].

Some studies compared performance with dataset of two different languages as described in 2.2.1. However, in recent works, text data of various domain is collected for measuring performance of their proposed method on different setting of experiments. In paper[21], authors conducted experiment in 4 datasets: first two datasets are collected from Stanford Sentiment Treebank [53] to analysis sentiment of movie reviews, third is TREC dataset for categorising six question types in and fourth one is a large set of twitter posts for sentiment analysis. In paper [26], authors created eight large-scale text datasets that ranges from hundreds of thousands to several millions of samples. In paper [5], author tested their proposed convolutional neural network against six datasets: first one called MR includes movie review dataset for positive/negative review analysis, next two dataset named SST-1 and SST-2 is collected from Stanford Sentiment Treebank [53] for sentiment analysis and in case of SST-2 neutral reviews were eliminated, third dataset is a subjectivity dataset which is previously used in [54] to determine the subjective/objective of a sentence), fourth is TREC question dataset [55], fifth dataset CR involves customer reviews of various product to foretell positive/negative reviews and finally sixth dataset is MPQA dataset [56]. In [27], authors incorporated three different datasets to validate their Recurrent Neural Network and Convolutional Neural Network based model to classify short texts:

- DSTC 4: Dialog State Tracking Challenge [57].
- MRDA: ICSI Meeting Recorder Dialog Act Corpus [58].
- SwDA: Switchboard Dialog Act Corpus [59].

For Bangla text categorization, online newspapers are the most utilised source in various studies such as Prothom Alo, Bdnews24, Jugantor, Manabzamin, Kaler Kontho, AnandabazarPatrika, Bartaman, Ebela Tabloid, etc.[60], [38], [42], [39], [41], [36], [43], [37], [7] , [40],[61], [62].

However, in paper [63], authors constructed a readability classifier to measure the ease of grasping a text where corpus is gathered from school textbooks used in the education system of Bangladesh. Furthermore, Bangla twitter text data is collected for sentiment analysis [64]. In paper [65], dataset that is used for sense classification of Bangla sentences is developed under the TDIL (Technology Development for the Indian Languages) project of the Govt. of India which contains 84 domain of subjects.

2.2.3 Feature Selection methods

Feature selection method is a task of crucial importance in terms of text categorization. Various feature extraction tools are incorporated in terms of text categorization. In earlier studies, bag of word feature extraction method was used [25]. However, Term Frequency- Inverse Document Frequency(TF-IDF) is by far the most used and proven method for word-level feature extraction [26]. However, over the years, some other feature techniques have shown to outperform TF-IDF. In [35], authors tested Latent Semantic Indexing (LSI), TF-IDF and multi-word text representation combining with Support Vector Machine and their experimental results showed that LSI worked better in both Chinese and English datasets used in their study. In [66], term weighting scheme tf.rf (term frequency-relevance frequency) outperformed TF-IDF with Support Vector Machine(SVM). In [39], authors adopted a new feature selection technique Term Frequency- Inverse Document Frequency – Inverse Class Frequency(TF-IDF-ICF) which attained better result than both Term Frequency(TF) and Term Frequency-Inverse Document Frequency TF-IDF in case of classifying Bangla text documents. Nonetheless, these methods are tested with few algorithms, while there are other well-established algorithms which combining with TF-IDF provided strong results regardless of language distinction [24],[67],[38],[42],[36],[40],[37].

Word embedding or word vector is another feature extraction method that has gain much popularity with the increased interest in neural network. In paper [5], authors used publicly available word2vec vectors which were prepared by training 100 billion words from Google News. Moreover, Word2Vec with skip-gram algorithm is used to create Bengali word embedding model for Bangla document categorization using deep convolutional neural network [7]. In paper[68], the authors used word2vec model to create word embedding and used t-distributed Stochastic Neighbor Embedding (t-SNE) to reduce the dimension of the vector into two in order to attain less computation time while creating word clusters with k-means clustering algorithm and tested word clusters with Support Vector Machine(SVM) to categorize text documents. These Word clusters are used as features for classifying Bangla texts.

However, there is no hard and fast rule for feature selection and extraction as numerous studies has exhibited diverse feature selection systems. In [27], authors used Recurrent Neural Network (RNN) or Convolutional Neural Network (CNN) based vector representations of text as features in order to classify short text. In [21], authors used feature graph induced by Dynamic Convolutional Neural Network (DCNN) consisting convolution layer and dynamic k-max pooling layer for semantic modelling of sentences. In paper [26], authors used characters as features for character-level text classification. In this study, they included 70 characters comprising 26 English letters, 10 digits, 33 other characters and the new line character. Authors did not differentiate between upper-case and lower-case letters as semantics remains unaffected with alternate letter cases. In [22], authors used a wide convolution layer, a k-max pooling layer and a non-linear function to obtain feature maps of varying order. Furthermore, term association and term aggregation are investigated as feature extraction approaches for Bangla text classification in [41], [43]. In paper [6], authors used TF-IDF vector representation with autoencoders constructed with NN to represent high dimensional feature into lower space. Moreover, in paper [44], authors initially used unigram TF-IDF score to investigate the consequence of exploring the varieties of linear and non-linear Support Vector Machine (SVM) kernels of in case of Bangla news classification. However, they observed that linear SVM handle linearly inseparable cases by transforming data into some higher dimensions. As their dataset already contains a lot of features which translates as large amount of dimensions, they did not prefer combining TF-IDF with linear SVM. Thus, they compared feature space by using Document Frequency(DF) Thresholding and Term Frequency(TF) Thresholding, and chose Term Frequency(TF) as primary feature selection technique.

2.2.4 Algorithms

Selecting an algorithm and harnessing it toward a specific task was a common practice before the progression of machine learning. Machine learning algorithms have reduced task specific engineering in manifolds making them applicable for variety of tasks which explains the increasing popularity of machine learning algorithms in various domain of expertise. Algorithm selection is nonetheless an important task in machine learning since every algorithm does not offer decent result for all sort of application. However, for the task of text document categorization, various supervised learning algorithms has provided outstanding result.

In case of English text classification, authors Decision Tree algorithm is compared with PropBayes algorithm in [4] to evaluate the applicability in text categorization on two English datasets and discovered that Decision Tree algorithm performed better than PropBayes. However, this study outlines that PropBayes does not use large number of features efficiently and Decision Tree's stepwise feature selection enables it to use more feature but demands more training set samples. In [25], authors implemented classification algorithms such as Naïve Bayes, nearest neighbor, decision tree classifier and subspace method in text classification task and observed that Naïve Bayes and Subspace Classifier algorithm performed better than Nearest Neighbor and Decision Tree. Another remark can be noted from this study is that combining multiple classifiers for classification task does not always provide better classification accuracy. In [67], authors implemented a framework combining TF-IDF and KNN for text classification task. For the case of Bangla document categorization, In paper [37], SGD classifier algorithm provided better performance in comparison to algorithms. In most studies, performance supervised algorithms such as Support Vector Machine (SVM), K Nearest Neighbor (KNN), Decision Tree (DT), Stochastic Gradient Descent (SGD), Naïve Bayes are compared with each other while different algorithms perform better in different studies due to the contrast in fine tuning of hyper parameters, feature selection, data preprocessing [40], [38], [25] , [41]. However, some other lesser known algorithms are experiment for Bangla document categorization in recent years such as PART classifier [43], LIBLINEAR [36], Cosine Similarity and Euclidean Distance used as similarity measures on vector space with TF-IDF[42] . In paper [44], different kernel functions such as linear polynomial and radial basis kernels of Support Vector machines are experimented and evaluated for Bangla text classification. In some studies, newly proposed feature selection techniques are experimented with classification algorithms which are known to provide good results. In paper [39], Multinomial Naïve Bayes algorithm is used to evaluate their proposed feature selection technique Term Frequency- Inverse Document Frequency – Inverse Class Frequency(TF-IDF-ICF) with more commonly used Term Frequency- Inverse Document Frequency (TF-IDF) and Term Frequency

(TF). In paper [68], authors used Support Vector Machine to test the efficiency of word clusters as features in terms of Bangla text categorisation. Furthermore, in recent years, neural networks with different architectures and kinds have achieved excellent outcome in text classification task. In [5], four Convolutional Neural Network(CNN) architectures are tested with six English dataset to validate the efficiency of CNN for sentence classification task. The experimental results provided evidence that CNN is applicable in text classification task. Various studies combined word embedding model such as Word2Vec with neural networks as classifier such as [5], [7], [68]. However, there are some other studies which does not incorporate Word2Vec or word embedding algorithms even though their exmeriments consist of CNN. In paper[21], authors constructed with Convolutional Neural Network (CNN) with dynamic k-max pooling for Modelling Sentences which performed well across various datasets described in 2.2.2. In paper [27], sequential CNN or RNN based vector representations of text are forwarded to a two-layered feedforward neural network as classifier for the task of short text classification. In paper [22] , authors proposed a Long Short Term Memory(LSTM) based architecture with K-max pooling layer and one dimensional spatial dropout layer which simulates Convolutional Neural Network. In paper [26], authors designed two convolutional neural network with convolutional layer and fully connected layer for character level text classification task. In paper [6], authors used low dimensional auto encoded text representations with neural network architecture as classifier to categorise Bangla texts. The architecture of the neural network includes 50, 50 and 100 hidden units, RELU activation function at first two layers and Softmax activation in final output layer.

2.2.5 Performance Evaluation

A model's applicability in terms of its target application is evaluated by its performance. Thus, performance evaluation is an imperative task for machine learning applications. There are various evaluation metrics to assess a model's capability and different studies choose diverse performance measures to estimate their adopted methods. A brief description is provided for some of the commonly used and known evaluation criterions in 2.1 In terms of English text categorization, authors used micro average of recall and precision as evaluation measure in [4]. In this study, with parameter controlling, they increased algorithm's document assigning capabilities which results increased recall and usually decreased precision. Also, they analyzed precision-recall curve and employed linerar interpolation to observe breakeven point on the curve. On the basis of recall/precision curve, authors experimented PropBayes and Decision Tree with varying feature sets selected by information gain and observed that Decision tree method performed better even on high recall value. In [67], authors tested a framework combining Term Frequency – Inverse Document Frequency (TF-IDF) and K- Nearest Neighbor(KNN) with 500 online documents. The testing was conducted in online environment and authors observed that classification performance was depended on the variety of categories since documents of same category provided better result. They also discerned that quality of classification decreased with increased number of documents. In [5] , authors included a single channel convolutional neural network(CNN) architecture similar to [21] and compared performance. Even though both study incorporated similar structures of CNN, [5] reported better performance than [21]. In paper [22], authors tested with six different datasets with Long Short Term Memory(LSTM) based architecture by comparing with several strong baseline methods and conjectured that a single machine learning model does not work well with all kinds of datasets. In paper [26], authors made observations that character level convolutional neural networks, otherwise known as ConvNets, are applicable in terms of text classification without needing any word. They also observed that character level classification works better with larger dataset along the scale of millions whereas traditional approach like n-gram TF-IDF performs well with dataset size that expanses up to several hundreds of thousands. For Bangla document categorisation, in [69], authors included training and testing time with precision, recall and F1-score as evaluation metrics and based on this metrics, Stochastic Gradient Descent outperformed other compared algorithms with precision value 0.9386, recall value 0.9388 and F1 score 0.9385. In paper [38], authors compared three types of sampling methods namely: training set validation, percentage split validation and cross-fold validation with multiclass models of several classification algorithms distinctly on full dataset and on selected features. This study used accuracy as evaluation metric and experimental results indicated that Logistic Regression algorithm provided better accuracy than other algorithms in all sampling techniques. In [61], authors compared TF-IDF and Chi-square as feature selection technique with Stochastic Gradient Descent, Naïve Bayes, Support Vector Machine algorithms and observed that TF-IDF based models performed better than Chi-square based models.

In [42], authors used accuracy, precision, recall and F-measure as evaluation metrics. Their proposed methods for text classification, cosine similarity achieved 95.80% accuracy, 0.958 in precision, 0.958 in recall and 0.958 in F-measure whereas Euclidean distance secured 95.20% accuracy, which are higher than other compared methods. They also included statistical nonparametric Friedman rank sum test for calculating rank of all algorithms, and in this ranking system, cosine similarity and Euclidean distance achieved 1st and 2nd rank respectively. In [39], authors introduced Term Frequency - Inverse Document Frequency - Inverse Class Frequency (TF-IDF-ICF) as a feature selection technique and evaluated its performance with accuracy, precision, recall and F1 measure as performance measure. Experimental results depicted that TF-IDF-ICF based model scored 98.87% in accuracy, 0.989 in precision, 0.989 in recall and 0.989 in F1 measure which is better than other experiments conducted in this study. In paper [7], deep convolutional neural network is employed for text classification task. This approach obtained 94.96% accuracy and showed better performance than other experiments conducted in this study. In [6], authors experimented a deep feed forward network in terms of text classification and their proposed method achieved 94.05% accuracy.

Chapter 3

Proposed Method

Methodology of a thesis offers intricate specificity of the work conducted by providing information about data sources, analysis techniques, performance scrutiny, etc. This chapter provides details about our proposed method and other experiments conducted in this study.

3.1 Block Diagram

This study entails three sequential steps: A) preprocessing raw text data and distributing the total dataset into train set and test set, B) applying classifier algorithms and neural networks to test set and finally, C) prediction and model evaluation. The block diagram of the methodology of this thesis work is provided in Figure 3.1.

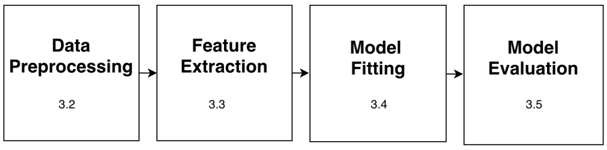


Figure 3.1: Block Diagram - Steps of Methodology.

In section 3.2, data collection and preprocessing is described. Section 3.3 provides details about feature extraction techniques. Section 3.4 delivers model fitting analogies of conducted experiments. Section 3.5 offers information about model evaluation.

3.1.1 Data Preprocessing

1600 articles of 16 distinct classes are collected from two online Bangla newspaper known as ProthomAlo (http://www.prothom-alo.com) and BDnews24 (http:// bdnews24.com). Documents were previously labeled in the mentioned newspapers. These documents are obtained by parsing Bangla texts from paper articles and further arranged into 16 classes where 100 articles belong to each class. The 16 categories are namely: Commerce, Science, Art and literature, Economy, Education, Entertainment, Immigrant, Kids, Lifestyle, Movie, National, Politics, Sports, Stock, Technology, World. The whole dataset is divided into two sets: training set containing 1120 articles and testing set composed of 480 articles. Each class has 70 training documents and 30 test documents. Stratification of dataset guarantees that training/testing set would contain documents from all possible categories which is uncertain when it comes to random sampling. Thus, dataset is stratified. Furthermore, punctuation marks, English letters, mail addresses, html tags are removed from each raw document while tokenizing.

3.2 Feature extraction

In terms of feature extraction, Term Frequency-Inverse Document Frequency (TF-IDF) is used for classification algorithms. Top 1500 features with the highest term frequency and document frequency less than 0.7 are selected where each feature appears in more than four documents during the process. In the case of neural networks, 5000 words with the highest word frequencies in training set are selected for feature extraction where each word is mapped to an integer number (known as token). These tokens represent sequential data and for Convolutional Neural Network (CNN), sequential data is essential since TF-IDF does not interpret neighbourhood among words. However, for multilayer dense neural network, both TF-IDF and word vectors are incorporated with two different model architectures. Moreover, CNN architecture includes embedding layer which creates word vectors to be used as features. These word vectors are also known as distributional similarity based word representations. These representations also provides contextual similarity between words which results into a more effective representation. The maximum number of words in a document is 2037 among all the documents of the training set. Each document requires to have equal length to be utilized by embedding layer. Hence, each document size is fixed to 2000 words by padding and truncating as needed.

3.3 Model Fitting

Model fitting denotes the ability of generalization of a machine learning algorithm. It requires a suitable algorithm for the target task. After fitting a model on training data, the performance of the model is evaluated. In 3.3.1, model fitting process of this work is described.

3.3.1 Classification Algorithms

Six well- known classification algorithms: K Nearest Neighbor, Decision Tree, Support Vector Machine, Stochastic Gradient Descent, Multinomial Naïve Bayes, and Logistic Regression are used in this study since these algorithms have been used in many text classification works as described in 2.2.1. Different models are created with these algorithms with different parameter values and these models are fitted on the training set. Subsequently, grid search is applied for each model on training set with different sets of parameters for obtaining optimal hyper parameter values. Hyper-parameter optimization is a pivotal task since a model's performance is highly influenced by it. Optimal parameters are determined based on the best score of 10-fold cross validation. During this process, training set is being fitted to each model with optimal set of parameters. This method is illustrated in Figure 3.2.

3.3.2 Neural Networks

Convolutional Neural Network In convolutional neural network model (CNN), embedding layer is used as part of the model itself to procure word embedding. This layer learns word vectors along with the model while being fitted on training data. Each input document is provided as sequence of N tokens (N=2000) to the model's embedding layer that transforms the tokens into n-dimensional word vectors (n=128). Subsequently, a one dimensional spatial dropout layer is included for regularization as suggested in paper [22]. Next, a one dimensional convolution layer is included in front of a one dimensional pooling layer, namely GlobalMaxPool layer. Finally, the output layer of this architecture is a dense layer with 16 units is where softmax is added with

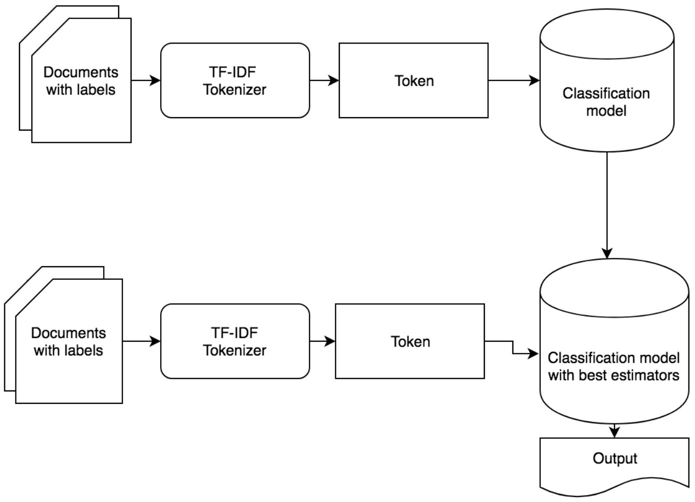


Figure 3.2: Process Diagram for Classification Algorithms - used for K Nearest Neighbor, Decision Tree, Support Vector Machine, Stochastic Gradient Descent, Multinomial Naïve Bayes, and Logistic Regression.

as activation function. The diagram of Convolutional Neural Network architecture is given in Figure 3.3

Dense neural networks For this study, two architectures of dense neural network are experimented.

- DenseNN1: One dense neural network includes embedding layer as a part of architecture to construct word vector and conduct classification task simultaneously.
- DenseNN2: other one uses TF-IDF as feature selection technique hence it does not contain embedding layer (proposed model).

In terms of dense neural network model which contains embedding layer as the first layer (DenseNN1), it takes input as sequences like convolutional neural network (CNN). However, instead of spatial dropout layer, a flatten layer is added that transforms multi-dimensional word vectors into one dimensional tensors. These tensors can be employed by the following dense layer. Afterwards, a dropout layer is added for regularization. The output layer has the same structure as CNN. The diagram for Dense Neural Network (DenseNN1) architecture is depicted in Figure 3.4.

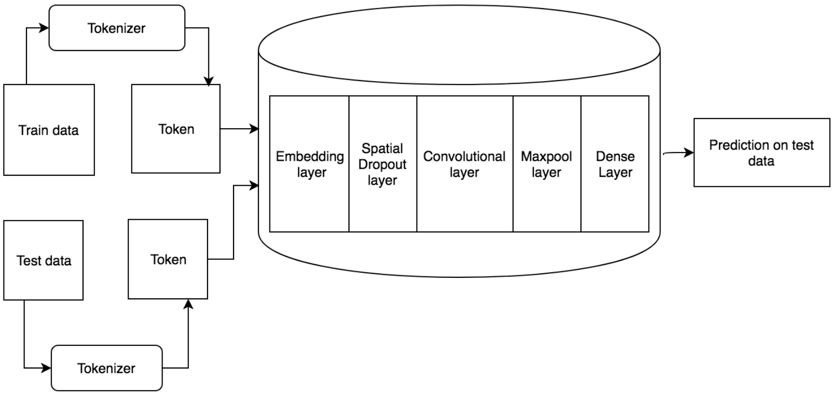


Figure 3.3: Convolutional Neural Network (CNN) architecture - with word vectors as features

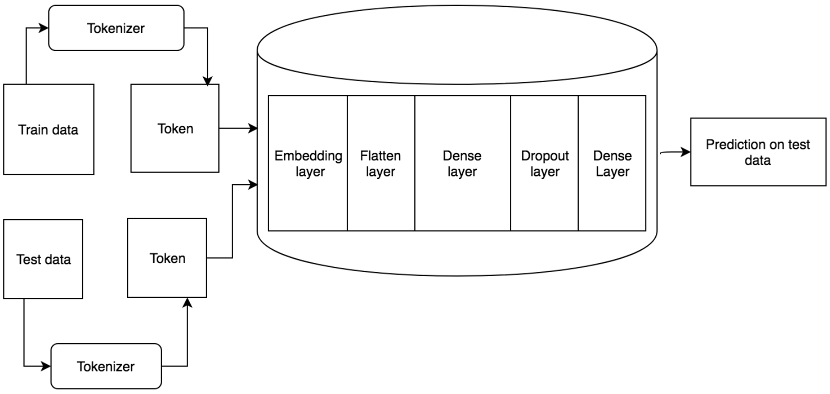


Figure 3.4: Dense Neural Network (DenseNN1) architecture - with word vectors as features.

The proposed model DenseNN2 uses TF-IDF feature selection method. The first layer of this model is a dense layer which uses TF-IDF features as input. It is worth mentioning that this model does not include any flatten layer or spatial dropout layer due to the absence of embedding layer. A dropout layer is added instead of previously mentioned layers after each of the first two dense layer to fulfil the purpose of regularizer. The objective of a dropout layer is to refine generalization in performance by inhibiting strong correlation among activation functions. Finally, the output layer is a dense layer with 16 units and uses softmax as activation function. The diagram of the Proposed Method: Dense Neural Network (DenseNN2) architecture is illustrated in Figure 3.5.

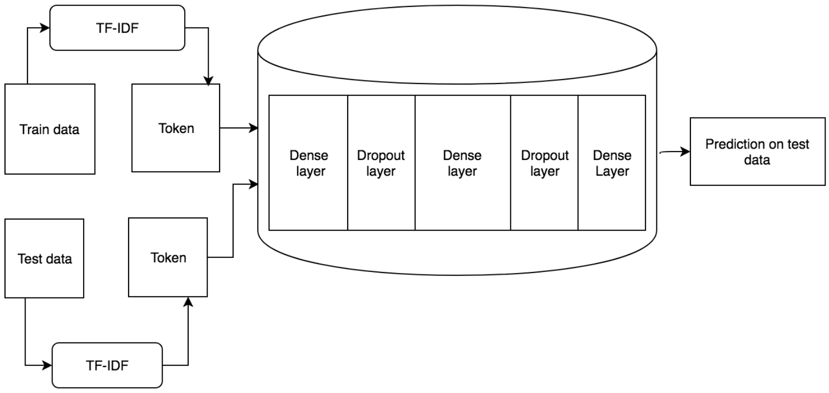


Figure 3.5: The Proposed Method: Dense Neural Network (DenseNN2) architecture - with TF-IDF features.

10 fold cross validation is used for estimating model's capacity of predicting on unseen data for all neural network architectures and classification techniques. Cross fold validation provides an estimation of the skill of a machine learning model and choosing the number 10 for cross fold validation is due to the fact that various experiments conducted on varying datasets with different learning methods have depicted that 10 folds can obtain the best estimation [70]. The reason for choosing 128 dimension for embedding layer in CNN and DenseNN1 is that paper [71] shows that 128 dimensions can capture the necessary semantic informations. In our proposed model (DenseNN2), input layer has 128 nodes as well since in [72], authors experimented over 16, 32, 64 and 128 number of units in input layer for abnormality detection in X-ray images and found that 128 nodes performed plausibly well. Uniformly distributed initialiser are

used in embedding layer and for all dense layer in these architectures, 'glorot_uniform' is used as kernel initialiser and 'zeros' is used as bias initialiser. Batch size is usually selected as number that is power of two since the number of physical processor of GPU is often a power of two and selecting batch size ensures optimal computation due to data parallelism. In our experiments, batch size is 128. Learning rate is 0.001 which is default value of adam optimiser.

3.4 Model Evaluation

To evaluate the performance of our experimental models, we used accuracy and F1 score as performance measures. Prediction on test set is conducted for each fitted model of all nine experiments. Predicted labels and true labels of test set are used to quantify accuracy and F1 score. Moreover, 10 fold cross validation is performed on both traditional classification algorithms and neural networks for model estimation.

3.5 Summary

In summary, this chapter provides the structural information of the experiments designed for the purpose of this thesis. Experiments include K-Nearest Neighbor, Decision Tree, Support Vector Machine, Stochastic Gradient Descent classifier, Multinomial Naïve Bayes, Logistic Regression algorithms and three different architectures of neural networks where word embedding and TF-IDF are used as feature selection techniques. The proposed model contains TF-IDF as feature selection technique and dense neural network as classifier.

Chapter 4

Experimental Analysis

This chapter provides implementational details of the designed experiments described in chapter 3. Moreover, this provides information about performance evaluation analysis of the experiments conducted in this study.

4.1 Data Collection and Preprocessing

For this thesis work, we collected 1600 articles from two different online Bangla newspaper namely: ProthomAlo and BDnews24. These 1600 articles are collected from 16 different categories by parsing which are labeled previously in these Bangla newspapers. At next, these articles grouped into 16 classes with 100 articles per class in the same categories as in the newspapers. Since these documents are collected from online newspapers, raw data contains html-tags, email addresses, English letters and digits and many other unnecessary literals which will add noise to the dataset. Thus, these needless characters are excluded from the dataset by occupying regular expressions in python language. Python language is used for all the tasks conducted in this thesis work. These documents are further processed while tokenizing by incorporating Bangla Unicode values and keeping only Bangla words after tokenization by using regular expression based tokenization approach. Afterwards, the dataset is apportioned into training set and testing set by using train/test split where the testing set size is 30% and training set size is 70% of the whole dataset.

4.2 Experimental Setup

In this section, experimental intricacies of our conducted experiments are provided in details. Feature selection techniques and hyper parameter optimization of feature space is described in section 4.2.1. Fine tuning of algorithms of this thesis work are described in section 4.2.2. Lastly, outcome of this thesis work is analyzed in the following section 4.2.3.

4.2.1 Feature Selection

For classification algorithm based models, Term Frequency–Inverse Document Frequency (TF-IDF) is used as feature extraction technique. First, tokenized datasets are converted into bag of words by 'CountVectorizer' function of scikit-learn's preprocessing tool. In this process, several parameters that shapes the structure of bag of word methods are optimized such as 'encoding', 'max_df', 'min_df', 'token_pattern', 'max_features', 'lowercase'. Since this task is on Bangla language, the value of 'unicode' parameter is given as UTF-8 and in 'token_pattern' parameter, Bangla word pattern is provided as a regular expression. Parameter 'lowercase' value is set to 'True' as a default value, however, this is not applicable for Bangla language so this value is changed to 'False'. The value of 'max_df' and 'min_df' are 0.7 and 4 respectively. Value of parameter 'max_df' means if a word appears in 70% documents then that term is ignored, and 'min_df' optimization eliminates a word which appears in less than 4 documents. Optimization of these two parameters eliminates the requirement of stop words removal. Parameter 'max_features' value is 1500 which signifies top 1500 features are selected for this task after applying all the other optimizations. Subsequently, these bag of word vectors are transformed into TF-IDF vectors by a transforming function of scikit-learn called 'TfidfTransformer'. For neural networks, the most common 5000 words are kept as features and transformed into sequences by a preprocessing tool for text tokenization in a neural network library known as 'keras' which is built in python programming language. For the proposed model (DenseNN2), these sequences are converted into TF-IDF vector representations. For Neural Network models with embedding layer (CNN and DenseNN1), each input documents should be of fixed size to maintain spatial integrity. Therefore, maximum document length for training set is calculated and the highest number of words in a training document is 2037. For this experiment, each article's word length is affixed to 2000 words by pre-padding or pre-truncating.

4.2.2 Algorithms

For classification models, sets of parameters are primarily selected for K Nearest Neighbor, Decision Tree, Support Vector Machine, Stochastic Gradient Descent, Multinomial Naïve Bayes and Logistic Regression algorithm to find optimal parameter by doing grid search and cross validation jointly with 'GridSearchCV'. It is an extensive search method for obtaining ideal parameter values for an estimator by combining cross validation. This method picks the most suitable parameters for the specific task based on Cross Validation (CV) scores and incorporates the user-chosen estimator. The impact of various parameters on all these algorithms are described in 2.2.1. The best set of parameters obtained from the various combination of parameter sets after performing grid search and 10-fold cross validation is given below:

- Decision tree: optimal set of parameters is 'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 1
- Support Vector Machine: optimal set of parameters is 'C': 10, 'gamma': 0.1, 'kernel': 'rbf'
- Logistic Regression: 'multi_class': 'multinomial', 'solver': 'newton-cg'
- K Nearest Neighbor: 'n_neighbors': 15, 'p': 2, 'weights': 'distance'
- Multinomial Naïve Bayes: 'alpha': 0.1
- Stochastic Gradient Descent: 'alpha': 0.0001, 'loss': 'log'

In terms of neural networks, a neural network library namely 'keras' which is written in python programming language is used. All of the neural network architectures are constructed in sequential manner which facilitates these neural networks to be assembled layer by layer. For the proposed model (DenseNN2), input layer consists of 128 nodes and Rectified linear unit (RELU) is used as activation function. Second dense layer involves 64 nodes with sigmoid activation function. Furthermore, a dropout layer is included after each dense layer with a value of 0.5 as recommended [5]. For CNN and word vector based dense neural network (CNN, DenseNN1), embedding layer takes 5000 tokens as vocabulary and each token is represented in 128 dimensions (n=128) and maximum length of each sequence is 2000. However, spatial dropout layer is incorporated with a value of 0.2 in CNN whereas word vector based dense neural network includes flatten layer after the embedding layer of both of the models. For CNN, a one dimensional convolution layer, with 256 different filters of kernel size 3 is added with RELU as activation function followed by a 1-D global max pool layer. For dense NN, after flatten layer comes a dense layer composed of 128 units and Sigmoid as activation function, followed by a dropout layer with a value of 0.5 which purpose is to regularisation. Finally, output layer contains a dense layer with 16 units and Softmax activation function for all NN models. During model compilation, Adam optimiser is used with categorical cross entropy as loss function [73]. While doing 10 cross fold validation on neural networks, 10 models must be constructed individually and evaluated to analysis the efficiency of model's structure in capturing the details. This process is computationally expensive, however provide less biased estimate for the model. Thus, each neural network architecture is tested for 25 epochs with 10 different model with same architecture while training set is split into 10 folds. In each cross validation iteration, a model with the same structure is trained on nine folds and tested on one fold and this process is repeated until all of the folds are evaluated. Afterwards, one model for each three structures are tested for performance evaluation. Cross validation estimates at each iteration are shown as a chart in Figure 4.1

From this line plot, it can be noticed that the proposed model:Dense Neural Network with TF-IDF as Feature extractor (DenseNN2) has better estimation than other neural network architectures (CNN, DenseNN1) based on 10-fold cross validation method.

4.2.3 Result

For this study, Accuracy, weighted average of precision, weighted average of recall, weighted average of F1 scores are selected as evaluation metrics. To calculate this scores, confusion matrix is necessary. Thus, prediction is made on test dataset and confusion matrix is evaluated for all models. Confusion matrix provides information on how many documents of a class in test set is correctly classified or misclassified. Consequently, accuracy, F1 score is calculated from confusion matrix. Confusion matrix

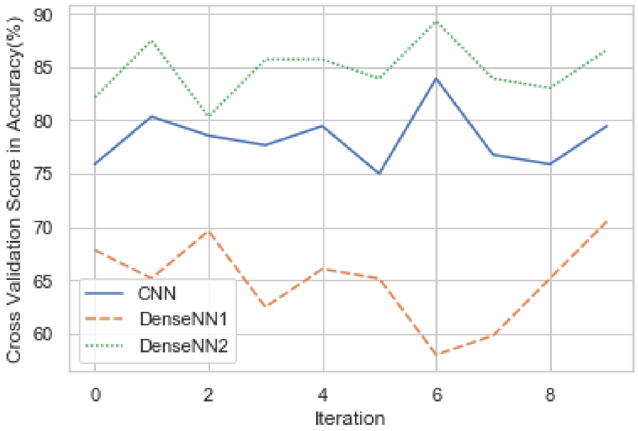


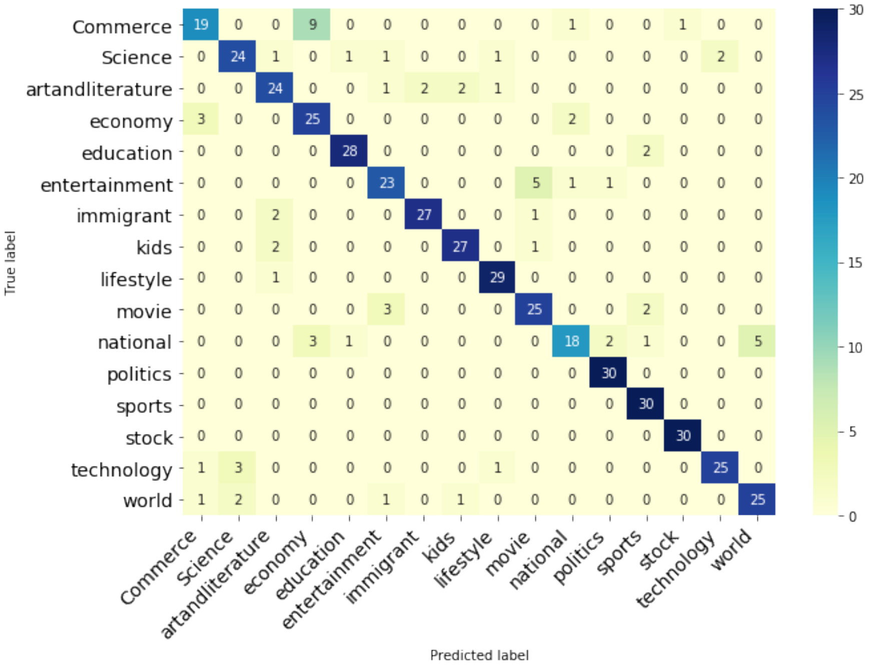
Figure 4.1: Cross Validation Scores Of Neural Network Models - 10-fold cross validation score of all three models after each iteration.

Table 4.1: Performance Evaluation Scores

Algorithms	F1 Score(weighted average)	Precision	Recall	Accuracy(%)
DenseNN2	0.85	0.85	0.85	85.208
Support Vector Machine	0.82	0.82	0.82	81.870
Stochastic Gradient Descent	0.82	0.83	0.82	82.890
Logistic Regression	0.81	0.81	0.81	80.830
Multinomial Naïve Bayes	0.80	0.81	0.81	80.620
CNN	0.80	0.79	0.79	79.166
K-Nearest Neighbor	0.76	0.82	0.74	74.375
DenseNN1	0.65	0.62	0.61	62.291
Decision Tree	0.53	0.59	0.52	51.670

of all the models are illustrated in Figure 4.2-4.10. Accuracy, precision, recall, F1 score and cross validation score of all models are depicted in Table 2.

Discussion It is noticable from all the confusion matrices from Figure (4.2- 4.10) that dataset shows interclass relations among the classes which means documents of one class can be categorize into other class as well. This can be noticed in the case of several classes such as documents in ‘movie’ category can be classified into ‘entertainment’ category, ‘commerce’ and ‘economy’ class shares similar content, and documents of ‘national’ classes also can include various kinds of informations which can be content wise similar to ’world’ ‘economy’, ‘education’,‘politics’ news. The proposed model, Dense Neural Network with TF-IDF as Feature extractor(DenseNN2) shows this mis-



	Commerce	Science	artandliterature	economy	education	entertainment	immigrant	kids	lifestyle	movie	national	politics	sports	stock	technology	world
Commerce	19	0	0	9	0	0	0	0	0	0	1	0	0	1	0	0
Science	0	24	1	0	1	1	0	0	1	0	0	0	0	0	2	0
artandliterature	0	0	24	0	0	1	2	2	1	0	0	0	0	0	0	0
economy	3	0	0	25	0	0	0	0	0	0	2	0	0	0	0	0
education	0	0	0	0	28	0	0	0	0	0	0	0	2	0	0	0
entertainment	0	0	0	0	0	23	0	0	0	5	1	1	0	0	0	0
immigrant	0	0	2	0	0	0	27	0	0	1	0	0	0	0	0	0
kids	0	0	2	0	0	0	0	27	0	1	0	0	0	0	0	0
lifestyle	0	0	1	0	0	0	0	0	29	0	0	0	0	0	0	0
movie	0	0	0	0	0	3	0	0	0	25	0	0	2	0	0	0
national	0	0	0	3	1	0	0	0	0	0	18	2	1	0	0	5
politics	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0
sports	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0
stock	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0
technology	1	3	0	0	0	0	0	0	1	0	0	0	0	0	25	0
world	1	2	0	0	0	1	0	1	0	0	0	0	0	0	0	25

Figure 4.2: Confusion Matrix of Proposed model: DenseNN2 - Dense Neural Network with TF-IDF as Feature extractor.

classification trend as well as other well performed classifier algorithms such as Support Vector Machine, Stochastic Gradient Descent. In terms of neural networks, both the proposed model DenseNN2 and CNN architecture showed almost similar pattern in confusion matrices. Both of the architectures did misclassification among categories which are content-wise similar and almost interchangeable except for DenseNN1 which architecture includes embedding layer. Multidimensional Word vectors are flattened in this architecture which caused dimensionality reduction. Dimensionality reduction can be a benefit when large number of features are involved. However, in case of small dataset, reducing dimensionality can lead to data loss. This might be the reason of low performance of this neural network architecture.

The worst performing experiment contains Decision tree algorithm. However, this phenomena can be explained. Although Decision tree algorithm is reliable in terms of text classification, it uses features stepwise requiring more data [4]. Since our dataset is small, it can be expected that performance of decision tree can be improved with employing more training instances.

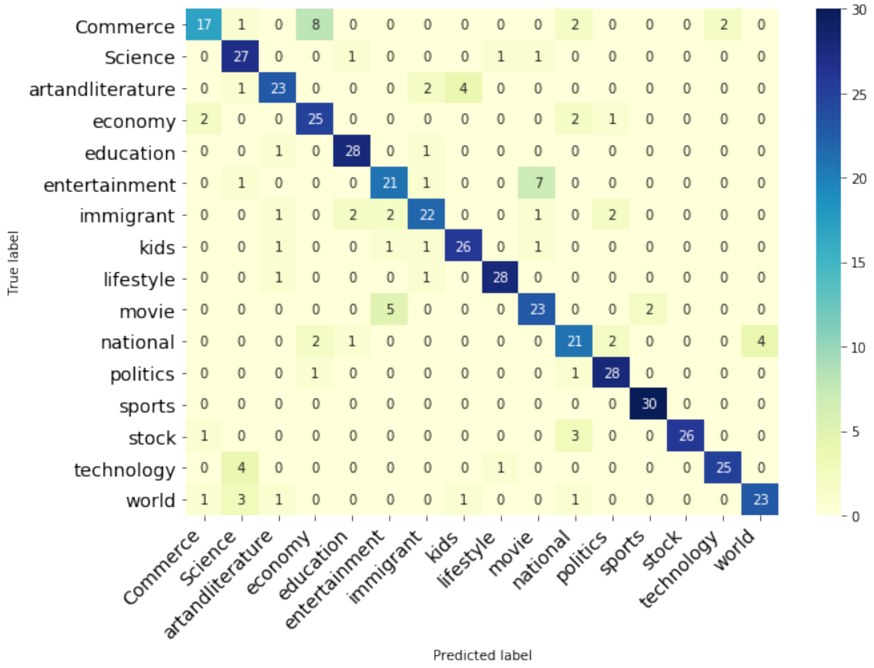
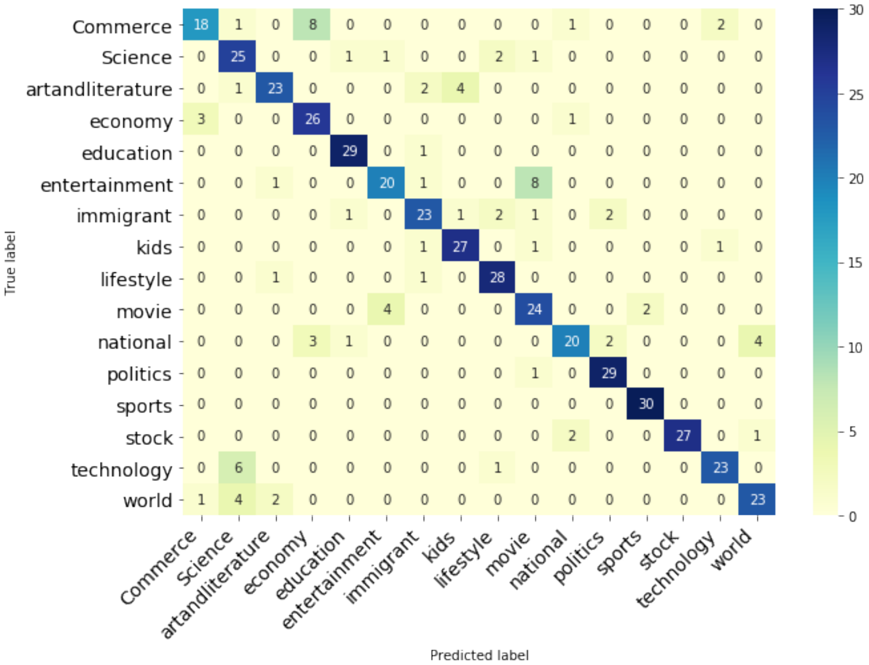


Figure 4.3: Confusion Matrix: Support Vector Machine - Support Vector Machine model with optimal parameters.

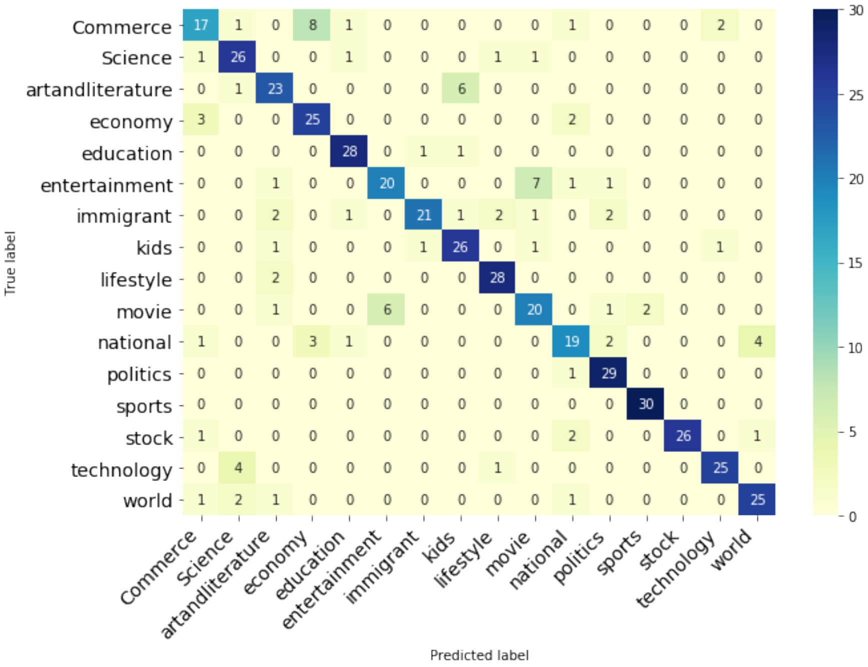
Table 4.1 shows the performance evaluation of all the experiments conducted in this thesis.From the table, it can be noticed that Our proposed method: Dense Neural Network with TF-IDF as Feature extractor (DenseNN2) outscored (85.208% in accuracy, 0.85 in precision, recall and weighted F1-score) other algorithms(Support Vector Machine, Stochastic Gradient Descent, Logistic Regression, Multinomial Naïve Bayes, Convolutional Neural Network(CNN), K-Nearest Neighbor, DenseNN: Dense Neural Network with embedding layer as part of architecture, Decision Tree) on the basis of accuracy(%), weighted average of precision, recall, and F1-measure. Support Vector Machine(SVM) and Stochastic Gradient descent(SGD) models also provided satisfactory performance in terms of evaluation metrics. Decision tree model is the worst performing one among all the experiments. In case of Neural Networks, the proposed method (DenseNN2) performed better than other architectures(DenseNN1, CNN). Since TF-IDF based methods performed better than word vector based CNN and DenseNN, it can be said that TF-IDF remains a strong candidate as feature selection technique for small-scale dataset. Furthermore, it is worth mentioning that



	Commerce	Science	artandliterature	economy	education	entertainment	immigrant	kids	lifestyle	movie	national	politics	sports	stock	technology	world
Commerce	18	1	0	8	0	0	0	0	0	0	1	0	0	0	2	0
Science	0	25	0	0	1	1	0	0	2	1	0	0	0	0	0	0
artandliterature	0	1	23	0	0	0	2	4	0	0	0	0	0	0	0	0
economy	3	0	0	26	0	0	0	0	0	0	1	0	0	0	0	0
education	0	0	0	0	29	0	1	0	0	0	0	0	0	0	0	0
entertainment	0	0	1	0	0	20	1	0	0	8	0	0	0	0	0	0
immigrant	0	0	0	0	1	0	23	1	2	1	0	2	0	0	0	0
kids	0	0	0	0	0	0	1	27	0	1	0	0	0	0	1	0
lifestyle	0	0	1	0	0	0	1	0	28	0	0	0	0	0	0	0
movie	0	0	0	0	0	4	0	0	0	24	0	0	2	0	0	0
national	0	0	0	3	1	0	0	0	0	0	20	2	0	0	0	4
politics	0	0	0	0	0	0	0	0	0	1	0	29	0	0	0	0
sports	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0
stock	0	0	0	0	0	0	0	0	0	0	2	0	0	27	0	1
technology	0	6	0	0	0	0	0	0	1	0	0	0	0	0	23	0
world	1	4	2	0	0	0	0	0	0	0	0	0	0	0	0	23

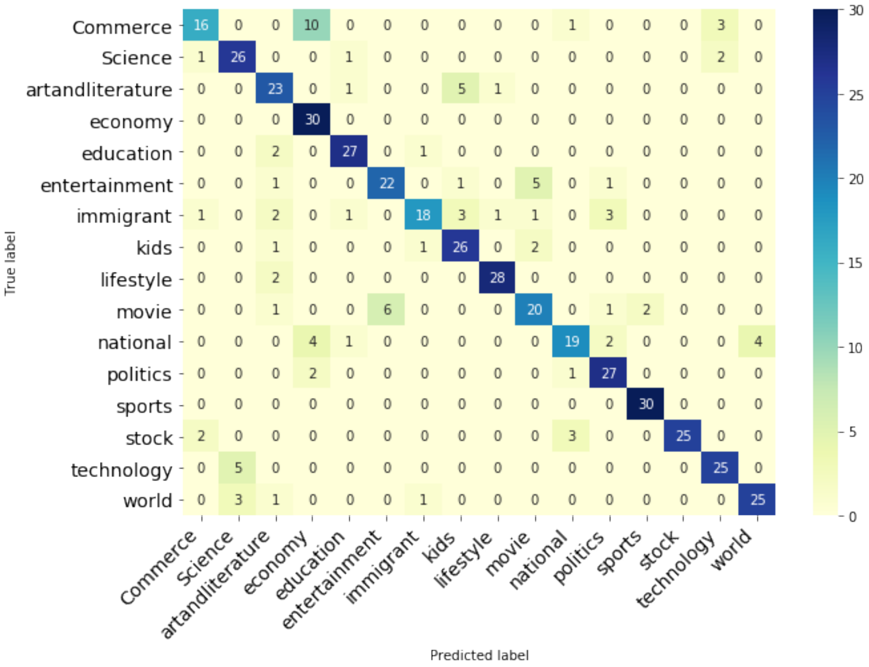
Figure 4.4: Confusion Matrix: Stochastic Gradient Descent - Stochastic Gradient Descent model with optimal parameters.

although neural networks in general requires more training data than compared to other traditional machine learning algorithms, in this study, both the proposed method: DenseNN2 and CNN performed surprisingly well with DenseNN2 surpassing other experimental model on the scale of evaluation metrics.



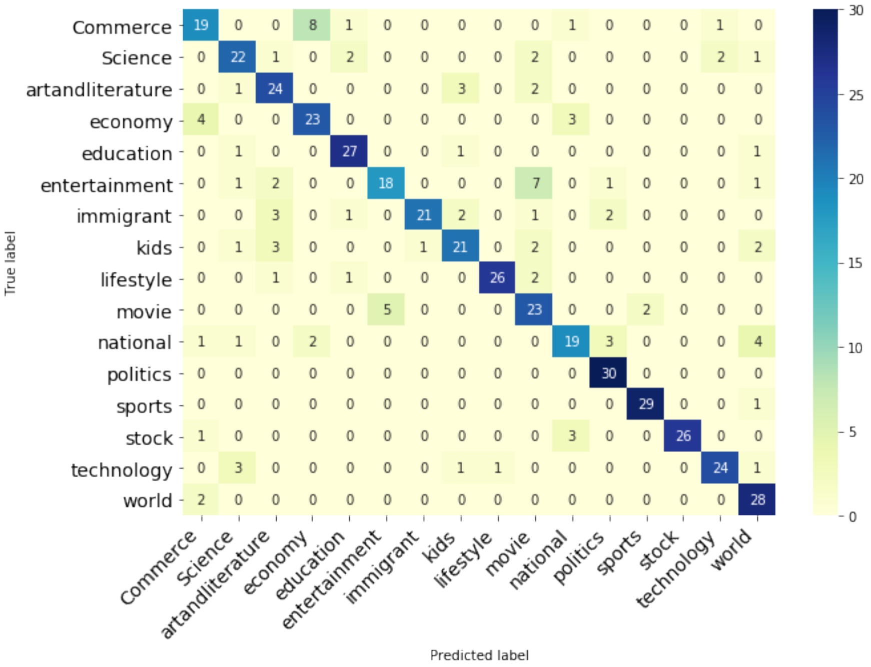
	Commerce	Science	artandliterature	economy	education	entertainment	immigrant	kids	lifestyle	movie	national	politics	sports	stock	technology	world
Commerce	17	1	0	8	1	0	0	0	0	0	1	0	0	0	2	0
Science	1	26	0	0	1	0	0	0	1	1	0	0	0	0	0	0
artandliterature	0	1	23	0	0	0	0	6	0	0	0	0	0	0	0	0
economy	3	0	0	25	0	0	0	0	0	0	2	0	0	0	0	0
education	0	0	0	0	28	0	1	1	0	0	0	0	0	0	0	0
entertainment	0	0	1	0	0	20	0	0	0	7	1	1	0	0	0	0
immigrant	0	0	2	0	1	0	21	1	2	1	0	2	0	0	0	0
kids	0	0	1	0	0	0	1	26	0	1	0	0	0	0	1	0
lifestyle	0	0	2	0	0	0	0	0	28	0	0	0	0	0	0	0
movie	0	0	1	0	0	6	0	0	0	20	0	1	2	0	0	0
national	1	0	0	3	1	0	0	0	0	0	19	2	0	0	0	4
politics	0	0	0	0	0	0	0	0	0	0	1	29	0	0	0	0
sports	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0
stock	1	0	0	0	0	0	0	0	0	0	2	0	0	26	0	1
technology	0	4	0	0	0	0	0	0	1	0	0	0	0	0	25	0
world	1	2	1	0	0	0	0	0	0	0	1	0	0	0	0	25

Figure 4.5: Confusion Matrix: Logistic Regression - Logistic Regression model with optimal parameters.



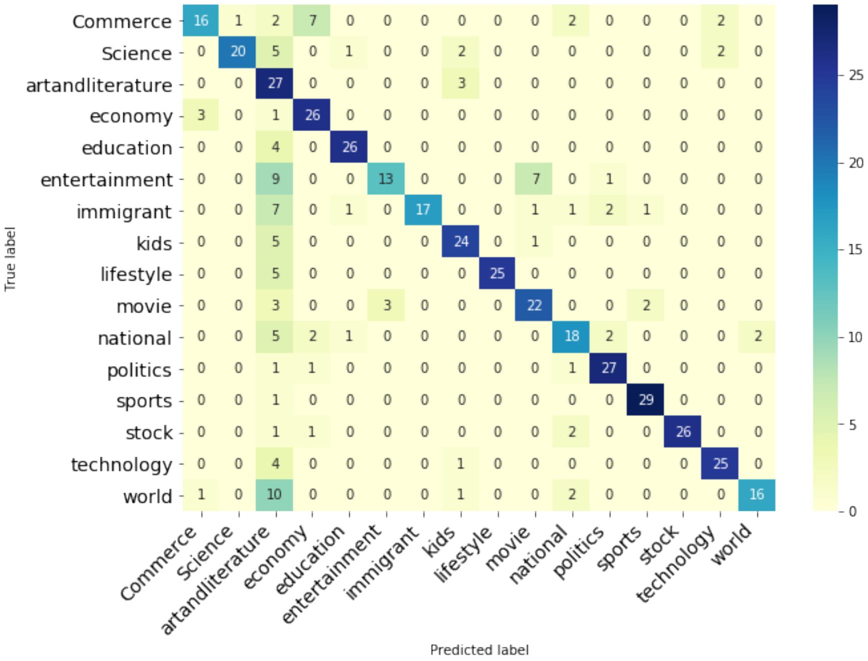
	Commerce	Science	artandliterature	economy	education	entertainment	immigrant	kids	lifestyle	movie	national	politics	sports	stock	technology	world
Commerce	16	0	0	10	0	0	0	0	0	0	1	0	0	0	3	0
Science	1	26	0	0	1	0	0	0	0	0	0	0	0	0	2	0
artandliterature	0	0	23	0	1	0	0	5	1	0	0	0	0	0	0	0
economy	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0
education	0	0	2	0	27	0	1	0	0	0	0	0	0	0	0	0
entertainment	0	0	1	0	0	22	0	1	0	5	0	1	0	0	0	0
immigrant	1	0	2	0	1	0	18	3	1	1	0	3	0	0	0	0
kids	0	0	1	0	0	0	1	26	0	2	0	0	0	0	0	0
lifestyle	0	0	2	0	0	0	0	0	28	0	0	0	0	0	0	0
movie	0	0	1	0	0	6	0	0	0	20	0	1	2	0	0	0
national	0	0	0	4	1	0	0	0	0	0	19	2	0	0	0	4
politics	0	0	0	2	0	0	0	0	0	0	1	27	0	0	0	0
sports	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0
stock	2	0	0	0	0	0	0	0	0	0	3	0	0	25	0	0
technology	0	5	0	0	0	0	0	0	0	0	0	0	0	0	25	0
world	0	3	1	0	0	0	1	0	0	0	0	0	0	0	0	25

Figure 4.6: Confusion Matrix: Multinomial Naïve Bayes - Multinomial Naïve Bayes model with optimal parameters.



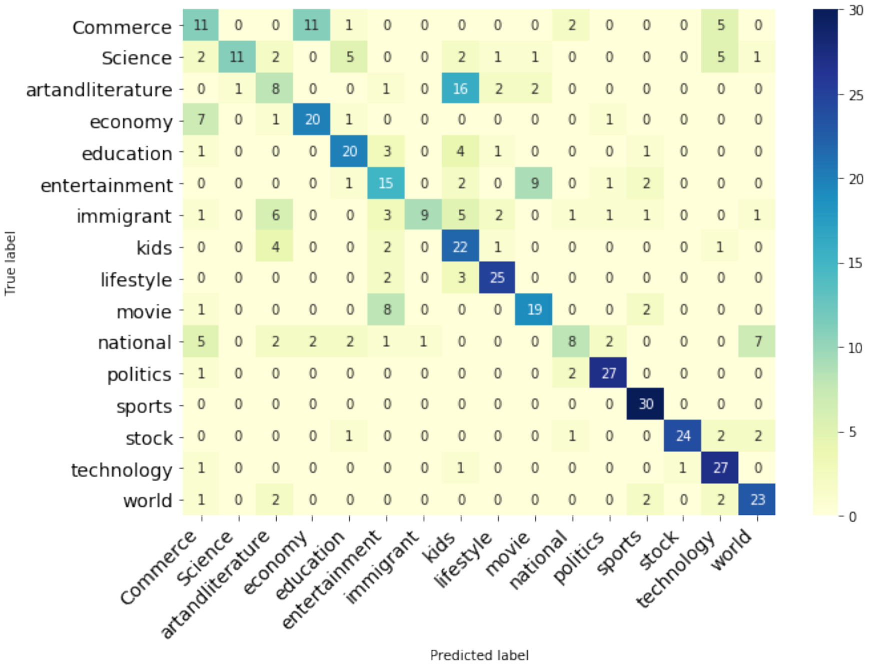
	Commerce	Science	artandliterature	economy	education	entertainment	immigrant	kids	lifestyle	movie	national	politics	sports	stock	technology	world
Commerce	19	0	0	8	1	0	0	0	0	0	1	0	0	0	1	0
Science	0	22	1	0	2	0	0	0	0	2	0	0	0	0	2	1
artandliterature	0	1	24	0	0	0	0	3	0	2	0	0	0	0	0	0
economy	4	0	0	23	0	0	0	0	0	0	3	0	0	0	0	0
education	0	1	0	0	27	0	0	1	0	0	0	0	0	0	0	1
entertainment	0	1	2	0	0	18	0	0	0	7	0	1	0	0	0	1
immigrant	0	0	3	0	1	0	21	2	0	1	0	2	0	0	0	0
kids	0	1	3	0	0	0	1	21	0	2	0	0	0	0	0	2
lifestyle	0	0	1	0	1	0	0	0	26	2	0	0	0	0	0	0
movie	0	0	0	0	0	5	0	0	0	23	0	0	2	0	0	0
national	1	1	0	2	0	0	0	0	0	0	19	3	0	0	0	4
politics	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0
sports	0	0	0	0	0	0	0	0	0	0	0	0	29	0	0	1
stock	1	0	0	0	0	0	0	0	0	0	3	0	0	26	0	0
technology	0	3	0	0	0	0	0	1	1	0	0	0	0	0	24	1
world	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28

Figure 4.7: Confusion Matrix: Convolutional Neural Network - Convolutional Neural Network with word embedding (CNN).



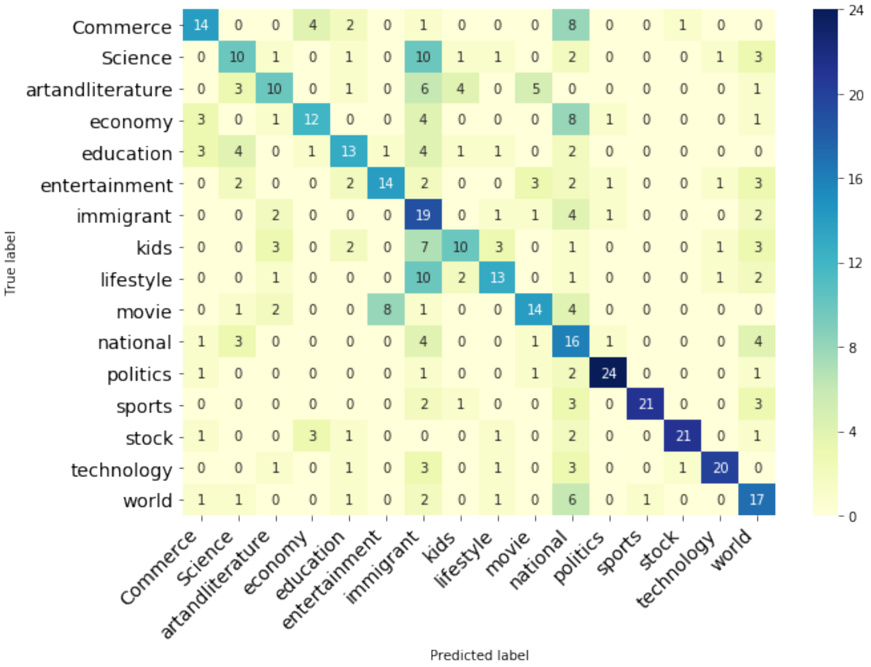
	Commerce	Science	artandliterature	economy	education	entertainment	immigrant	kids	lifestyle	movie	national	politics	sports	stock	technology	world
Commerce	16	1	2	7	0	0	0	0	0	0	2	0	0	0	2	0
Science	0	20	5	0	1	0	0	2	0	0	0	0	0	0	2	0
artandliterature	0	0	27	0	0	0	0	3	0	0	0	0	0	0	0	0
economy	3	0	1	26	0	0	0	0	0	0	0	0	0	0	0	0
education	0	0	4	0	26	0	0	0	0	0	0	0	0	0	0	0
entertainment	0	0	9	0	0	13	0	0	0	7	0	1	0	0	0	0
immigrant	0	0	7	0	1	0	17	0	0	1	1	2	1	0	0	0
kids	0	0	5	0	0	0	0	24	0	1	0	0	0	0	0	0
lifestyle	0	0	5	0	0	0	0	0	25	0	0	0	0	0	0	0
movie	0	0	3	0	0	3	0	0	0	22	0	0	2	0	0	0
national	0	0	5	2	1	0	0	0	0	0	18	2	0	0	0	2
politics	0	0	1	1	0	0	0	0	0	0	1	27	0	0	0	0
sports	0	0	1	0	0	0	0	0	0	0	0	0	29	0	0	0
stock	0	0	1	1	0	0	0	0	0	0	2	0	0	26	0	0
technology	0	0	4	0	0	0	0	1	0	0	0	0	0	0	25	0
world	1	0	10	0	0	0	0	1	0	0	2	0	0	0	0	16

Figure 4.8: Confusion Matrix: K Nearest Neighbor - K Nearest Neighbor model with optimal parameters.



	Commerce	Science	artandliterature	economy	education	entertainment	immigrant	kids	lifestyle	movie	national	politics	sports	stock	technology	world
Commerce	11	0	0	11	1	0	0	0	0	0	2	0	0	0	5	0
Science	2	11	2	0	5	0	0	2	1	1	0	0	0	0	5	1
artandliterature	0	1	8	0	0	1	0	16	2	2	0	0	0	0	0	0
economy	7	0	1	20	1	0	0	0	0	0	0	1	0	0	0	0
education	1	0	0	0	20	3	0	4	1	0	0	0	1	0	0	0
entertainment	0	0	0	0	1	15	0	2	0	9	0	1	2	0	0	0
immigrant	1	0	6	0	0	3	9	5	2	0	1	1	1	0	0	1
kids	0	0	4	0	0	2	0	22	1	0	0	0	0	0	1	0
lifestyle	0	0	0	0	0	2	0	3	25	0	0	0	0	0	0	0
movie	1	0	0	0	0	8	0	0	0	19	0	0	2	0	0	0
national	5	0	2	2	2	1	1	0	0	0	8	2	0	0	0	7
politics	1	0	0	0	0	0	0	0	0	0	2	27	0	0	0	0
sports	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0
stock	0	0	0	0	1	0	0	0	0	0	1	0	0	24	2	2
technology	1	0	0	0	0	0	0	1	0	0	0	0	0	1	27	0
world	1	0	2	0	0	0	0	0	0	0	0	0	2	0	2	23

Figure 4.9: Confusion Matrix: Dense Neural Network - Dense Neural Network with embedding layer as part of architecture(DenseNN1).



	Commerce	Science	artandliterature	economy	education	entertainment	immigrant	kids	lifestyle	movie	national	politics	sports	stock	technology	world
Commerce	14	0	0	4	2	0	1	0	0	0	8	0	0	1	0	0
Science	0	10	1	0	1	0	10	1	1	0	2	0	0	0	1	3
artandliterature	0	3	10	0	1	0	6	4	0	5	0	0	0	0	0	1
economy	3	0	1	12	0	0	4	0	0	0	8	1	0	0	0	1
education	3	4	0	1	13	1	4	1	1	0	2	0	0	0	0	0
entertainment	0	2	0	0	2	14	2	0	0	3	2	1	0	0	1	3
immigrant	0	0	2	0	0	0	19	0	1	1	4	1	0	0	0	2
kids	0	0	3	0	2	0	7	10	3	0	1	0	0	0	1	3
lifestyle	0	0	1	0	0	0	10	2	13	0	1	0	0	0	1	2
movie	0	1	2	0	0	8	1	0	0	14	4	0	0	0	0	0
national	1	3	0	0	0	0	4	0	0	1	16	1	0	0	0	4
politics	1	0	0	0	0	0	1	0	0	1	2	24	0	0	0	1
sports	0	0	0	0	0	0	2	1	0	0	3	0	21	0	0	3
stock	1	0	0	3	1	0	0	0	1	0	2	0	0	21	0	1
technology	0	0	1	0	1	0	3	0	1	0	3	0	0	1	20	0
world	1	1	0	0	1	0	2	0	1	0	6	0	1	0	0	17

Figure 4.10: Confusion Matrix: Decision Tree - Decision Tree model with optimal parameters.

Chapter 5

Conclusion and Future works

This chapter provides the concluding remark of this thesis work, the limitations it has and the future direction of this thesis. In section 5.1, conclusion of this thesis work provided. In section 5.2, limitations are discussed and lastly, in section 5.3, future direction of this thesis work is outlined.

5.1 Conclusion

In conclusion, this thesis presents a system to categorize Bangla text document by integrating TF-IDF as feature selection technique with dense neural network. The main verdicts are depicted as follows:

- Proposed system has accomplished higher accuracy (85.208%) and F1-score (0.85) then rest of the experiments.
- Proposed method which incorporates Term Frequency-Inverse Document Frequency based Neural Network has superior performance than word embedding based Neural Network models.
- Neural Networks perform well on relatively small size dataset in terms of Bangla text classification contrary to the popular belief of neural network requiring large-scale dataset for better performance.
- Among the classification models, Support Vector Machine is a close competitor to the proposed model based on performance.

5.2 Limitations

The limitation is of this work is that when two class have similarity in the contents of their articles, classifier may predict incorrectly since some article can be part of both of the class. For instance, accuracy discrimination is noticed between class movie and class entertainment, since both can hold similar kind of articles which raises the question to multi-label classification problem. Considering the fact that an article can belong to more than one classes, implementing multi-label categorization can improve this limitation.

5.3 Future Work

The future direction of this work is given as follows:

- Implementing multi-label classification to improve the classification of articles which belong to different class with similar contents.
- Incorporating hybrid approach with deep neural network on larger dataset regarding Bangla text document categorization.
- Reducing dimensionality of features by including PCA to improve model performance.

Bibliography

[1] M. Krendzelak and F. Jakab, "Text categorization with machine learning and hierarchical structures," in *2015 13th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, Nov 2015, pp. 1–5. 1

[2] A. McCallum, "Multi-label text classification with a mixture model trained by em," in *AAAI workshop on Text Learning*, 1999, pp. 1–7. 1

[3] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001. 1, 8, 17, 19

[4] D. Lewis, C. Info, L. Studies, and M. Ringuette, "A comparison of two learning algorithms for text categorization," *Third Annual Symposium on Document Analysis and Information Retrieval*, 10 1996. 1, 11, 17, 22, 23, 38

[5] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014. 1, 13, 19, 20, 23, 24, 35

[6] B. Purkaystha, T. Datta, M. S. Islam *et al.*, "Layered representation of bengali texts in reduced dimension using deep feedforward neural network for categorization," in *2018 21st International Conference of Computer and Information Technology (ICCIT)*. IEEE, 2018, pp. 1–5. 3, 18, 21, 23, 25

[7] M. R. Hossain and M. M. Hoque, "Automatic bengali document categorization based on deep convolution nets," in *Emerging Research in Computing, Information, Communication and Applications*. Springer, 2019, pp. 513–525. 3, 18, 19, 20, 23, 25

[8] G. Forman and I. Cohen, "Learning from little: Comparison of classifiers given little training," in *European Conference on Principles of Data Mining and Knowledge Discovery.* Springer, 2004, pp. 161–172. 3

[9] K. Masuda, T. Matsuzaki, and J. Tsujii, "Semantic search based on the online integration of nlp techniques," *Procedia-Social and Behavioral Sciences*, vol. 27, pp. 281–290, 2011. 7

[10] C. Friedman, T. C. Rindflesch, and M. Corn, "Natural language processing: state of the art and prospects for significant progress, a workshop sponsored by the national library of medicine," *Journal of biomedical informatics*, vol. 46, no. 5, pp. 765–773, 2013. 7

[11] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 1, pp. 1–309, 2017. 7, 8

[12] L. H. Lee, C. H. Wan, R. Rajkumar, and D. Isa, "An enhanced support vector machine classification framework by using euclidean distance function for text document categorization," *Applied Intelligence*, vol. 37, no. 1, pp. 80–99, 2012. 8

[13] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016. 8

[14] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *European conference on machine learning.* Springer, 1998, pp. 4–15. 10

[15] 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*, vol. 752, no. 1. Citeseer, 1998, pp. 41–48. 10

[16] S. Xu, "Bayesian naïve bayes classifiers to text classification," *Journal of Information Science*, vol. 44, no. 1, pp. 48–59, 2018. 10

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014. 12

[18] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656. 13

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf 13

[20] W.-t. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 643–648. 13

[21] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014. 13, 19, 21, 23, 24

[22] B. Shu, F. Ren, and Y. Bao, "Investigating lstm with k-max pooling for text classification," in *2018 11th International Conference on Intelligent Computation Technology and Automation (ICICTA)*. IEEE, 2018, pp. 31–34. 13, 21, 23, 24, 28

[23] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015. 13

[24] 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. 17, 19, 20

[25] Y. H. Li and A. K. Jain, "Classification of text documents," *The Computer Journal*, vol. 41, no. 8, pp. 537–546, 1998. 17, 20, 22

[26] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657. 17, 18, 19, 20, 21, 23, 24

[27] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 515–520. 17, 19, 21, 23

[28] P. Bolaj and S. Govilkar, "Text classification for marathi documents using supervised learning methods," *International Journal of Computer Applications*, vol. 155, no. 8, pp. 6–10, 2016. 17

[29] R. M. Rakholia and J. R. Saini, "Classification of gujarati documents using naïve bayes classifier," *Indian Journal of Science and Technology*, vol. 5, pp. 1–9, 2017. 17

[30] N. Krail and V. Gupta, "Domain based classification of punjabi text documents using ontology and hybrid based approach," in *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing*, 2012, pp. 109–122. 17

[31] H. M. Noaman, S. Elmougy, A. Ghoneim, and T. Hamza, "Naive bayes classifier based arabic document categorization," in *2010 The 7th International Conference on Informatics and Systems (INFOS)*, March 2010, pp. 1–5. 17

[32] A. H. Mohammad, O. Al-Momani, and T. Alwada'n, "Arabic text categorization using k-nearest neighbour, decision trees (c4. 5) and rocchio classifier: a comparative study," *International Journal of Current Engineering and Technology*, vol. 6, no. 2, pp. 477–482, 2016. 17

[33] M. Gumilang and A. Purwarianti, "Experiments on character and word level features for text classification using deep neural network," in *2018 Third International Conference on Informatics and Computing (ICIC)*, Oct 2018, pp. 1–6. 18

[34] Y. Jin, C. Luo, W. Guo, J. Xie, D. Wu, and R. Wang, "Text classification based on conditional reflection," *IEEE Access*, vol. 7, pp. 76 712–76 719, 2019. 18

[35] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of tf* idf, lsi and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, 2011. 18, 19, 20

[36] A. Dhar, N. S. Dash, and K. Roy, "Application of tf-idf feature for categorizing documents of online bangla web text corpus," in *Intelligent Engineering Informatics*. Springer, 2018, pp. 51–59. 18, 19, 20, 22

[37] 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. 18, 19, 20, 22

[38] 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. 18, 19, 20, 22, 24

[39] A. Dhar, N. S. Dash, and K. Roy, "Classification of bangla text documents based on inverse class frequency," in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*. IEEE, 2018, pp. 1–6. 18, 19, 20, 22, 25

[40] A. K. Mandal and R. Sen, "Supervised learning methods for bangla web document categorization," *arXiv preprint arXiv:1410.2045*, 2014. 18, 19, 20, 22

[41] 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. 18, 19, 21, 22

[42] A. Dhar, N. Dash, and K. Roy, "Classification of text documents through distance measurement: an experiment with multi-domain bangla text documents," in *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)*. IEEE, 2017, pp. 1–6. 18, 19, 20, 22, 25

[43] A. Dhar, N. S. Dash, and K. Roy, “An innovative method of feature extraction for text classification using part classifier,” in *International Conference on Information, Communication and Computing Technology.* Springer, 2018, pp. 131–138. 18, 19, 21, 22

[44] Q. I. Mahmud, N. I. Chowdhury, and M. Masum, “Reducing feature space and analyzing effects of using non linear kernels in svm for bangla news categorization,” in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP).* IEEE, 2018, pp. 1–6. 18, 21, 22

[45] M. S. Islam, F. E. M. Jubayer, and S. I. Ahmed, “A support vector machine mixed with tf-idf algorithm to categorize bengali document,” in *2017 international conference on electrical, computer and communication engineering (ECCE).* IEEE, 2017, pp. 191–196. 18

[46] A. DHAR, N. S. DASH, and K. ROY, “A fuzzy logic-based bangla text classification for web text documents.” 18

[47] T. Dash Roy, S. Khatun, R. Begum, and A. M. Saadat Chowdhury, “Vector space model based topic retrieval from bengali documents,” in *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, Oct 2018, pp. 60–63. 18

[48] Q. Li, L. He, and X. Lin, “Dimension reduction based on categorical fuzzy correlation degree for document categorization,” in *2013 IEEE International Conference on Granular Computing (GrC)*, Dec 2013, pp. 186–190. 19

[49] Y. Saad and K. Shaker, “Support vector machine and back propagation neural network approach for text classification.” 19

[50] V. Tam, A. Santoso, and R. Setiono, “A comparative study of centroid-based, neighborhood-based and statistical approaches for effective document categorization,” in *Object recognition supported by user interaction for service robots*, vol. 4, Aug 2002, pp. 235–238 vol.4. 19

[51] Z. Zhen, H. Wang, L. Han, and Z. Shi, "Categorical document frequency based feature selection for text categorization," in *2011 International Conference of Information Technology, Computer Engineering and Management Sciences*, vol. 2, Sep. 2011, pp. 65–68. 19

[52] Ziqiang Wang, Xia Sun, and Qingzhou Zhang, "Document categorization algorithm based on kernel npe," in *2009 Chinese Control and Decision Conference*, June 2009, pp. 2958–2961. 19

[53] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642. 19

[54] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics.* Association for Computational Linguistics, 2004, p. 271. 19

[55] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1.* Association for Computational Linguistics, 2002, pp. 1–7. 19

[56] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan, "Opinionfinder: A system for subjectivity analysis," in *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, 2005, pp. 34–35. 19

[57] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams, M. Henderson, and K. Yoshino, "The fifth dialog state tracking challenge," in *2016 IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2016, pp. 511–517. 19

[58] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The icsi meeting corpus," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 1. IEEE, 2003, pp. I–I. 19

[59] D. Jurafsky, R. Bates, N. Coccaro, R. Martin, M. Meteer, K. Ries, E. Shriberg, A. Stolcke, P. Taylor, and C. Van Ess-Dykema, "Automatic detection of discourse structure for speech recognition and understanding," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings.* IEEE, 1997, pp. 88–95. 19

[60] M. N. Hasan, S. Bhowmik, and M. M. Rahaman, "Multi-label sentence classification using bengali word embedding model," in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, Dec 2017, pp. 1–6. 19

[61] M. Islam, F. E. M. Jubayer, S. I. Ahmed *et al.*, "A comparative study on different types of approaches to bengali document categorization," *arXiv preprint arXiv:1701.08694*, 2017. 19, 24

[62] M. R. Hossain and M. M. Hoque, "Automatic bengali document categorization based on word embedding and statistical learning approaches," in *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2).* IEEE, 2018, pp. 1–6. 19

[63] Z. Islam, M. R. Rahman, and A. Mehler, "Readability classification of bangla texts," in *International conference on intelligent text processing and computational linguistics.* Springer, 2014, pp. 507–518. 20

[64] S. Chowdhury and W. Chowdhury, "Performing sentiment analysis in bangla microblog posts," in *2014 International Conference on Informatics, Electronics & Vision (ICIEV).* IEEE, 2014, pp. 1–6. 20

[65] A. R. Pal, D. Saha, and N. S. Dash, "Automatic classification of bengali sentences based on sense definitions present in bengali wordnet," *arXiv preprint arXiv:1508.01349*, 2015. 20

[66] M. Lan, C.-L. Tan, H.-B. Low, and S.-Y. Sung, "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines," in *Special interest tracks and posters of the 14th international conference on World Wide Web.* ACM, 2005, pp. 1032–1033. 20

[67] B. Trstenjak, S. Mikac, and D. Donko, “Knn with tf-idf based framework for text categorization,” *Procedia Engineering*, vol. 69, pp. 1356–1364, 2014. 20, 22, 24

[68] A. Ahmad and M. R. Amin, “Bengali word embeddings and it’s application in solving document classification problem,” in *2016 19th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2016, pp. 425–430. 20, 23

[69] J. Kaur and J. R. Saini, “A study of text classification natural language processing algorithms for indian languages,” *The VNSGU Journal of Science, Technology*, pp. 162–167, 2015. 24

[70] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., ser. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 2005. 31

[71] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, “Massive exploration of neural machine translation architectures,” *arXiv preprint arXiv:1703.03906*, 2017. 31

[72] T. Sasaki, K. Kinoshita, S. Kishida, Y. Hirata, and S. Yamada, “Effect of number of input layer units on performance of neural network systems for detection of abnormal areas from x-ray images of chest,” in *2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS)*, Sep. 2011, pp. 374–379. 31

[73] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014. 36