2020 7th International Conference on Networking, Systems and Security (NSysS)

22-24 December, 2020, Dhaka, Bangladesh

Text Analysis

A Benchmark Study on Machine Learning Methods using Several Feature Extraction Techniques for News Genre Detection from Bangla News Articles & Titles

Rifat Rahman
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology
1405007.rr@ugrad.cse.buet.ac.bd

ABSTRACT

Genre detection from news articles or news titles is one kind of text classification procedures where news articles or titles are categorized among different families. Nowadays, text classification has become a key research field in text mining and natural language understanding because of it's several applications, such as search engine, document filtering, keywords extraction, text summarizing, etc. Several studies have been conducted for detecting news genres in different languages, but little research has been done in Bangla language due to the lack of Bangla resources. Moreover, in the few works of Bangla language, a little number of news genres have been used for classifications and dataset has been created from a small number of news sources. In this study, we have shown comparative analysis of different traditional machine learning, advanced neural networks, attention-based supervised learning models by extracting several informative features including TF-IDF, category-based word frequency, word embedding, etc. We have implemented these categorization methods for both news article and news title classifications against ten mutually exclusive news genres. From our analysis, We have found that 'Bidirectional LSTM' with 'Word2vec' feature extraction technique using 'Skip-gram' method is the most robust method for both classifications, compared to other models.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**; **Machine learning**.

KEYWORDS

Text classification, News genre detection, Bangla language, Machine learning, Neural networks, Feature extraction

ACM Reference Format:
Rifat Rahman. 2020. A Benchmark Study on Machine Learning Methods using Several Feature Extraction Techniques for News Genre Detection from Bangla News Articles & Titles. In *7th International Conference on Networking, Systems and Security (7th NSysS 2020), December 22–24, 2020, Dhaka, Bangladesh.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3428363.3428373

7th NSysS 2020, December 22–24, 2020, Dhaka, Bangladesh
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8905-1/20/12…$15.00
https://doi.org/10.1145/3428363.3428373

1 INTRODUCTION

As the number of electronic or online data is increasing nowadays, text classification has become a significant part of natural language understanding. Text classification is the procedure of grouping texts into a label or family. By categorizing text, one can extract information from a large unstructured document. There are several practical applications like searching, filtering, summarizing where we can use text classification.

Several researches have been conducted in different languages regarding genre detection from news articles. But very few studies have been done in Bangla which is the 7th most spoken language and nearly 265 million people from Bangladesh and India speak in Bangla as their mother tongue[1]. It is also a culturally rich language. Hence it becomes necessary to classify and organize Bangla texts, so that one can acquire related information easily.

Document or text classification in Bangla language is very challenging because of the scarcity of large corpus. Besides, Bangla is a complex language. The sentence and grammatical structure of Bangla language are very complicated. Furthermore, there is a lack of effective resources for basic pre-processing like stemmer, data cleaner, part-of-speech tagger, etc. In python language, nltk[2] library provides several services for many other languages including English, Spanish, etc but these are not fully applicable to Bangla language. So any natural language processing related tasks with Bangla language are so much challenging.

In our approach, we collect data of six years (2015–early 2020) from topmost read five online news portals of Bangladesh. We gather almost 6,40,000 news articles and titles with their corresponding genre. We categorize both news articles and news titles against ten mutually exclusive genres. Then we implement all well-known traditional machine learning, advanced neural networks, and attention-based models applying several feature extraction mechanisms and perform comparative performance analysis among these models.

Most of the works in Bangla news article categorization [1, 4, 12, 20] have used a tiny dataset or collected the data from one or two online news portals. Again some have taken a fewer amount of classes for categorization. Furthermore, various state-of-the-art works have taken only 'Bag of words' or 'TF-IDF' as features and

[1] https://www.dhakatribune.com/world/2020/02/17/bengali-ranked-at-7th-among-100-most-spoken-languages-worldwide
[2] https://www.nltk.org

applied some traditional machine learning approaches like Logistic Regression (LR), Naive Bayes (NB), Support Vector Machines (SVM), etc or simplest form of neural networks. We have applied both traditional machine learning and deep learning models for classification and collected our large corpus from diverse online news sources.

The rest of the paper is organized as follows. In Section 2 we have reviewed several related works on different types of text classification both for Bangla and other languages. Section 3 presents our dataset and describes our proposed methods. Section 4 depicts our experimental analysis. Finally, we conclude this work and provide future direction in Section 5.

2 RELATED WORKS

In English language, many research works have been performed for classifying text or document. The work done by T Joachims [14] on text classification used Support Vector Machines (SVMs) with TF-IDF term feature for learning text classifiers. He identified the appropriateness of SVMs by analyzed the consistency between empirical results and theoretical findings.

According to the study performed by Jingnian Chen et al. [3], good feature selection can increase scalability, efficiency & accuracy of text classifier. They chose ‘Multi-class Odds Ratio’ and ‘Class Discriminating Measure’ (CDM) as feature evaluation metrics for Naive Bayes classifier. Ajay S. Patil et al. [26] also used Naive Bayesian algorithm and got 89.05% accuracy against ten categories.

Comparative analysis that has been performed among ‘K-Nearest Neighbors’ (KNN), centroid-based classifier, and HASRD algorithm is the work by V. Tam et al. [32]. They found KNN as the best for document categorization. The research of Bijalwan et al. [2] found better accuracy using KNN, compared to Naive Bayes classifier and Term-graph method.

According to the survey on different classifiers, such as Decision Tree, KNN, Rocchio’s Algorithm, Backpropagation Network, NB, SVM for text classification carried out by Pratiksha Y Pawar et al. [27] showed that the performance of SVM is better than all other approaches using twenty Newsgroups dataset. The comparative experiments, done by Zhijie Liu et al. [19] also found that SVM is better than KNN and NB classifier. Joseph Lilleberg et al. [18] used word embedding (Word2Vec) as a feature for support vector machines classifier.

A Neural network based experiment for text classification is done by Yoon Kim [16]. He used Convolutional Neural Networks (CNN) with pre-trained word vectors for sentence-level classification. A comprehensive review performed by Shervin Minaee et al. [24] examined more than 150 deep learning based models for text classification.

In addition to English language, there are several research works in other languages. Wongso et al. [33] used different distributions for Naive Bayes classifier like Multinomial NB, Multivariate Bernoulli NB, etc, and SVM using TF-IDF and SVD algorithm for feature extraction in Indonesian language. They got Multinomial Naïve Bayes+TF-IDF as the best of all. K.Rajan et al. [28] used neural network based model for Tamil document categorization and achieved 93.33% accuracy. For Arabic text classification, Mohamed El Kourdi el al. [6] used Naive Bayes classifier to classify Arabic web documents against five pre-defined categories.

Very little works have been done for categorizing documents in the context of Bangla language, compared to other languages.

The study of Ankita Dhar et al. [5] on Bangla text categorization addresses features extraction techniques, such as ‘term association’ and ‘term aggregation’. They applied traditional machine learning models like ‘Random forest’ (RF), SVM, Multinomial NB, K*, and multi-layer Perceptron as the classifier. They trained classifiers against 8000 Bangla articles with eight categories.

By conducting a comparative experiment among some traditional machine learning methods that include Support Vector Machine (SVM), Naive Bayes (NB), and Stochastic Gradient Descent (SGD) using TF-IDF for feature extraction & Chi square distribution for feature selection, M Islam et al. [13] proposed SVM+TF-IDF as the best classifier. They performed their experiment on 31,908 Bangla news articles. Besides, a similar study is done in [20] where they examined among four supervised learning Methods like KNN, NB, SVM, and Decision Tree (DT) with TF-IDF for feature extraction. They performed this investigation by training 1000 web documents against five categories.

Recently a research [1] claimed their dataset (BARD) as comprehensive and large corpus of Bangla documents that includes 3,76,226 Bangla news articles. They applied Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF), Adaboost, and simple feed-forward neural network as classifier with TF-IDF and word2vec for feature extraction. They found that neural network based apporach with word2vec features, having low dimension, performed fine among other classifiers. But they took five categories in their experiments.

By reviewing different literature, we have observed that many of the works on Bangla document categorization used a small corpus for training supervised learning models. Again these data have been gathered from a tiny amount of news sources. Most of the researches have used a little amount of classes or categories for classification. Moreover, only feed-forward neural networks have been applied for Bangla text classification. Several advanced deep learning methods, such as convolutional neural networks (CNN), variants of recurrent neural networks (RNN) including LSTMs, GRUs, bidirectional RNNs, etc, and attention mechanisms have not been experimented in building models for Bangla text classification.

3 METHODOLOGY

In this section, we provide an overview of our implementations. First of all, we collect news articles & titles from diverse online news sources of Bangladesh and label them with their corresponding genres from respective sources. As the fetched data is unstructured, we apply several pre-processing mechanisms and remove noise. Then we extract features from the data. After that, we implement our different models and train the data based on features.

3.1 Dataset Creation

We fetch data from five top used dailies and online news portals from Bangladesh. The five online news portals include daily ‘prothom alo’, ‘Bangladesh Pratidin’, ‘kaler kantho’, ‘BBC Bangla’, ‘bd-news24’. We extract almost 6,40,000 news articles dated from 2015

Table 1: News Sources Distribution of Dataset of almost six years

News sources	State	International	Economy	Opinion	Sports	Entertainment	Science & Tech	Life-style & Career	Education	Art & Lit
Prothom alo[3]	2,7955	14,691	12,513	13,357	21,047	15,117	4,357	8,967	6,361	2,532
Bangladesh Pratidin[4]	26,351	17,327	13,230	13,704	20,930	16,732	5,233	8,749	6,726	2,605
Kaler kantho[5]	25,322	15,509	12,906	14,027	21,305	14,529	4,624	7,870	5,909	2,471
BBC Bangla[6]	26,741	16,917	14,357	15,100	20,204	15,622	5,150	8,536	6,492	2,397
bdnews24[7]	24,969	14,931	13,115	14,319	21,279	16,413	3,955	8,365	5,743	2,583
Total	1,29,178	79,375	66,121	70,507	1,04,765	78,413	23,319	42,487	31,231	12,588

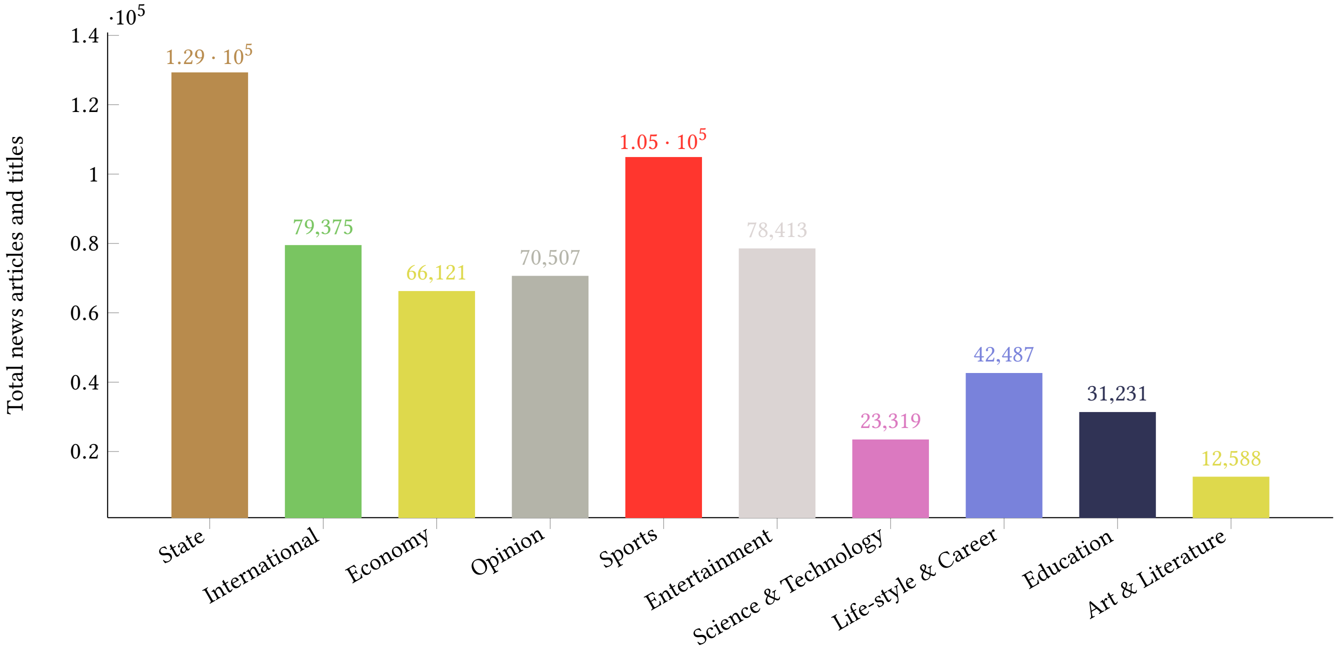


Figure 1: News Genre Distribution of Dataset

to early 2020 with their respective titles. We use urllib package and BeautifulSoup library of python language for extracting data. We crawl the text from the HTML page of the corresponding news web page.

We annotate each news content and title in the database with ten mutually exclusive genres. Table 1 provides a distribution over the news sources and news genres. These ten genres are state, international, economy, opinion, sports, entertainment, science & technology, life-style & career, education, and art & literature. We perform these annotations according to that corresponding news portals. Some news portals keep their news articles in many families. We narrow down these correlated families to our mutually independent ten categories. For example, the news related to youth, travel, jobs, health, etc can be included in the 'life-style & career' category. Again, many features related to media, music, comedy, etc can be subsumed into the 'entertainment' genre. We also include national and politics related articles in the 'state' class. However, few groups are conflicting, so we ignore those groups. The distribution of different news genre is depicted in Figure 1.

3.2 Pre-processing

The data that we extract from the web, are unstructured and noisy. Data often contains characters of other languages, duplication, error, etc. Again the syntactical structure of the Bangla language is complicated to understand for a model. The accuracy of a model depends on the integrity of the dataset.

We tokenize each sentence of every article. Then we remove stopwords and connecting words or conjunction from them. There are several resources for Bangla stopwords[8] and we take the list of connecting word from Bangla grammar book[9] approved by NCTB[10] of Bangladesh. After that, we clean the data from characters of other languages, punctuation, URL links, IP addresses, etc. Then we use a stemmer[11] for Bangla language to get root words. Sometimes stemming creates meaningless words, but we can ignore that because the stemming process is identical for all same words.

3.3 Feature Extraction

After the pre-processing step, we extract features applying several feature extracting techniques. We train our supervised models by feeding these feature vectors.

3.3.1 TF-IDF.

TF-IDF [30] refers the multiplication of TF (Term Frequency) and IDF (Inverse Document Frequency). TF-IDF is a popular technique for keyword and information retrieval. Only term frequency can not fetch the relevant words that carry the key information of a document. The equation of TF-IDF is:

$$TF(t,d) = \frac{count\ of\ term,\ t\ in\ document,\ d}{total\ number\ of\ words\ in\ document,\ d}$$

$$IDF(t,D) = log(\frac{total\ elements\ of\ document\ set,\ D}{number\ of\ documents\ in\ D\ that\ contain\ t} + 1)$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) * IDF(t,D)$$

We calculate TF-IDF of each uni-gram and bi-gram for every documents and consider the most informative uni-gram or bi-gram or both for feature vector.

3.3.2 Word Embedding.

For learning word embedding from a text corpus, Word2vec [23] is the most effective and popular technique. It is a neural networks based technique. It gives a vector representation of a predefined dimension to every word in the corpus, based on syntactic and semantic structure. There are two implementations of Word2vec.

1. Continuous Bag of Words (CBOW): 'CBOW' predicts a single center word with the highest probability from all the vocabulary words, taking context words as input.

2. Skip-gram (SG): The mechanism of skip-gram is totally opposite to 'CBOW'. 'SG' predicts some words with high probabilities that can fit with the given input word.

We implement both of these mechanisms by feeding our created corpus. For implementing 'CBOW' and 'SG', we use 1,00,000 words as vocabulary size. We take the sliding window size as 2. At first, we prepare the one hot encoding vector of context and center words of size $100,000 \times 1$. So, the number of neurons in the input layer is 100,000. We take the dimension of word embedding as 300. Hence, the number of hidden units of the only one hidden layer is 300 with $relu$ activation function. Again the output layer contains 1,00,000 nodes and the activation function is $softmax$. We feed the average of one hot encoding vectors of context words to the input layer in 'CBOW'. On the other hand, we feed the one hot vectors of center word to the input layer in 'SG'. Then we get two weight matrices of dimension $300 \times 100,000$ and $100,000 \times 300$ after applying backpropagation. We transpose the first weight matrix and take the average with the second weight matrix. Finally, we get the vector representations of 1,00,000 relevant words of our corpus with 300 dimensions.

3.3.3 Category based word frequency.

We measure the frequencies of each word of our corpus separately for ten categories. These ten separate frequencies of a word act as the weights of that particular word for different genres or categories. At the time of feature extraction, we take a 10 dimensional vector and initialize all the values by zero. We are taking 10 dimensional vector because of ten categories. We add all the ten different weights of a word including a particular document to all the corresponding ten entries of that initialized vector. This procedure is applicable for each and every word of that particular document. Finally, we extract the features, which are ten dimensional feature vectors, for all the news articles and titles.

3.3.4 Linguistic Inquiry & Word Count (LIWC).

There is no built-in LIWC [7] function developed for Bangla language. We explore all the news contents and titles for each category from our labeled dataset. Then we select maximum five relevant, informative, and most frequent words from the corpus of separate categories. We select total 40 words for news articles and 15 for news titles from ten categories. All these selected words are mutually exclusive and semantically independent. Then we evaluate the probabilities of these words for all the documents. We also use 'Bangla Wordnet'[12] for synonym mapping for effective probability measure.

3.4 Classification Model Architecture

We implement several traditional machine learning and advanced deep learning based supervised methods for both news article and title classification.

3.4.1 Traditional Machine learning Models.

We apply 10 fold cross validation for each of these classifiers and take the average scores.

Multi-class Logistic Regression: Logistic Regression [22] groups the dataset into categories by evaluating the probabilities for every category. We tune the value of C by taking values 1, 5, and 10. C is a hyper-parameter which is inversely proportional to regularization.

K Nearest Neighbor (KNN): When the training data points do not follow any mathematical data distribution, 'KNN' [35] is mostly applicable. This classifier does not create any model and all training data points are used for testing. 'KNN' classifies an input data point by analyzing the nearest K training data points. We tune K by taking odd values from 3 to 21.

Decision Tree: This model is a flowchart-like tree structure [29]. Internal nodes, branches, and leaves of this tree refer to features, decision rules, and labels respectively. We tune the maximum depth

[8] https://github.com/stopwords-iso/stopwords-bn

[9] https://drive.google.com/file/d/1Lz44Rw9btpgvBONTWYtDiagkkBwj_QNw/view

[10] http://nctb.gov.bd

[11] https://pypi.org/project/bangla-stemmer/

[12] https://raw.githubusercontent.com/soumenganguly/Bangla-Wordnet/master/load/synsets.yaml

and the minimum number of samples required to be a leaf node of the tree.

Random Forest (RF): Random forest [17] contains multiple decision trees and predicts the best output by analyzing all the trees. We tune the total number of decision trees in the random forest.

Multinomial Naive Bayes (NB): Naive Bayes [21] is a probabilistic classifier. We tune the value of smoothing parameter separately for both classifications.

Support Vector Machines (SVM): We tune the value of C and 'kernel' where C is inversely proportional to regularization [31].

Ada-Boosting: Adaptive Boosting (AdaBoosting) [11] is one kind of ensemble boosting classifiers that combines multiple weak classifiers to increase the overall accuracy. Thus Ada-boosting classifier creates a strong classifier. We use 'Decision Tree' classifier as the base estimator because from our investigation, we find 'Decision Tree' model as the weakest classifier. We apply total 100 iterations and tune the learning rate.

3.4.2 Neural Network based Models.

For neural network based approaches, we build an embedding layer at first. We use tokenizer to create one hot encoding vector from the processed sentences. We take the length of news articles of 800 words long and the length of news titles of 10 words long. We set these word lengths by applying statistical analysis on our corpus. We pad with zeros for shorter word lengthen news articles and titles. we take the most relevant 1,00,000 words as vocabulary. The dimension of the word embedding is 300. That is all of our embedding layer.

After the embedding layer, we add different layers like dense layers, 1D convolutional layers, recurrent neural layers, etc. Now we discuss the deep learning based models that we use for our classifications.

Multi-layer Perceptron (MLP): We implement one hidden dense layer with $relu$ activation function [25]. We tune the number of hidden units for the hidden layer. Then we add an output layer that contains 10 units because the number of classes is 10 and the activation function is $softmax$.

Convolutional Neural Network (CNN): We add 1D convolutional layer [15] with tuned number of filters after the embedding layer. We apply the kernel size as 3. Then we add a global max-pooling layer of pool size 2. To reduce overfitting, we take dropout probability by hyper-parameter tuning. Then we add a hidden dense layer with $relu$ activation function and an output layer of 10 units with $softmax$ activation function.

Simple Recurrent Neural Network (RNN): 'RNN' [10] performs the same task for every element of a sequence. This model contains a small memory that captures few information on the previous event. We tune the number of hidden units. Then we add an output dense layer.

Long Short Term Memory (LSTM): We use LSTM [8] for capturing long term dependencies of past information. We set the number of units for LSTM after performing hyper-parameter tuning. Then an output layer with $softmax$ activation & 10 neurons are included in that layer.

C-LSTM: C-LSTM [36] is the combination of an 1D convolutional layer and a LSTM layer. 1D convolutional layer extracts informative features and 'LSTM' layer learns long-term dependencies. We tune the hyper-parameters of the convolutional layer. Then we add the LSTM layer with tuned number of hidden units and dropout probability. We also add an output dense layer.

Bidirectional LSTM (Bi-LSTM): Document is the sequence of sentences where sentence is the sequence of words. The model needs to extract information both from previous and future events of action. So we implement 'Bidirectional LSTM' [9] which learns the long term dependencies. We tune the number of hidden units for LSTM. Then we add a hidden & a output dense layer like previous models.

Hierarchical Attention Network(HAN): Documents and sentences have hierarchical properties. A Document is formed by sentences and a sentence is formed by words. HAN [34] can capture the internal structure of a document and it has attention mechanisms both for word-level and sentence-level encoding. We set the maximum number of sentences per article as 80 and maximum words per sentence as 20 by performing statistical analysis on our corpus. For word encoder, we include bidirectional 'Gated Recurrent Unit' (GRU) with tuned number of units to the attention layer and add the layer to the network. Then we use this word encoder as the input of time distributed layer of sentence encoder.

Convolutional HAN (ConvHAN): In this model, we add an 1D convolutional layer before the hierarchical attention network. This reason behind adding a convolutional layer is to feed more relevant and informative features to HAN.

We choose 'adam' optimizer with the learning rate of 0.001 and 'sparse categorical cross-entropy' as the loss function for compiling all the models. We apply total 10 epochs and take the batch size as 256. We also use callbacks methods to reduce learning rate and for early stopping the training when needed. For reducing learning rate, we choose the reducing factor as 0.1 and patience as 5 epochs. We select 'validation accuracy' for monitoring each epoch of the training and set patience as 5 for early stopping tasks.

4 EXPERIMENTAL RESULT

In this section, we present the comparative performance analysis of our proposed models both for news article and title classification.

4.1 Experimental Setup

We use python programming language for all experiments. We do program on 'Google Colab'[13] platform. This platform provides both GPU and TPU services. For building and training traditional machine learning models, we use Scikit-learn[14] library. We also use Keras library with Tensorflow background for the implementation of deep learning based models. For data visualization and manipulation, we take support from some Python libraries including numpy, pandas, matplotlib, etc.

For machine learning models, we use 95% data for training. Besides, for the deep neural network based models, we reserve 5% of data for testing and from the rest data, we use 90% of them for training and 10% for validation.

[13] https://colab.research.google.com/notebooks/intro.ipynb
[14] https://scikit-learn.org/stable/

4.2 Performance Metrics

We calculated Precision, Recall and F1-score to measure the performances of our implemented supervised models.The formulas of these performance metrics are:

$$Precision = \frac{True\ Positive}{True\ Positive\ +\ False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive\ +\ False\ Negative}$$

$$F1\text{-}score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

Based on these performance metrics, we do the comparative analysis among all the models.

4.3 Hyper-parameter Tuning Result

For tuning hyper-parameters of traditional machine learning models, we use python library 'GridSearchCV'. We apply 10 fold cross-validation. We observe that 'Multinomial Naive Bayes' performs best when the smoothing parameter, α is set to 0.01 for both classifications. For logistic regression, we find that with the value of $C = 1$, the model performs the best. After tuning, we discover that taking 19 nearest neighbors for the 'KNN' model carries out the best result. This is because of our numerous number of genre classes. From our tuning experiment, we invent that the maximum depth of the decision tree of 12 works excellent, and the number of decision trees of 100 performs fine for the random forest model. We discover that for the SVM model, the value of C is 1 and the 'rbf kernel' is giving great output. The reason behind this finding is the versatility of Bangla text. From the learning rate tuning experiment for 'Ada-Boosting' model, we come to a decision that the learning rate value of 0.001 performs excellent with the 'Decision Tree' as base estimator.

We utilize 'Keras Tuner' for tuning the hyper-parameters for our neural network based models. For the hidden dense layer, we tune the number of hidden nodes by taking values 32, 64, 128, and 256. We find the value of 64 provides the best output. We tune the kernel size and dropout rate for the 1D convolutional layer and find the values 128 and 0.8 respectively. Performing tuning experiments on the number of hidden units for SimpleRNN, LSTM, and Bi-LSTM layers, we discover these values 300, 100, and 100 respectively. We find that the dropout rate of 0.4 works fine for SimpleRNN and LSTM layers. For HAN and ConvHAN, we get the number of hidden units for GRU as 100 after tuning.

For comparative performance analysis, we use the best tuned hyper-parameters. We present these result analysis in Subsection 4.4.

4.4 Result Analysis

4.4.1 Performance among different supervised models.

Table 2 and 3 are presenting the performance for deep learning methods and traditional machine learning models respectively with different features. We compare their performances based on performance metrics.

From Table 2, we find that LSTM based models are performing well. Because LSTM can capture long term dependencies. Among all the models that are using LSTM, 'Bidirectional LSTM' is doing the best of all for both Word2vec techniques. Again 'Bidirectional LSTM' is the best for both news article based classification and news title based classification. The F1-score of this model is 0.97 for news article classification and 0.91 for news title classification. We also find that both attention-based models are giving nearly the same result. Among all the models, multi-layer perceptron is giving low precision, recall, and F-1 score.

From Table 3, we observe that different models are giving fine results depending on feature extraction techniques and classification domains. For example, SVM is giving the best output for both news article and title classification when the feature extraction technique is TF-IDF. Again 'Logistic Regression' is giving the highest precision, recall, and F1-score with Word2vec (SG) feature for news article classification. With Word2vec (SG) feature 'Random Forest' is performing as the best when the classification is based on news title. Furthermore, 'Random Forest' is providing the best results for both classifications with 'category based word frequency' feature extraction. We observe that all the models with the 'Manual LIWC' feature extraction technique are giving poor results. This is because the resources and wordnet for Bangla language are not so much developed. Among all the machine learning models, 'Logistic Regression' with word2vec (SG) is giving the best result for news article classification. This model's precision, recall, and F1-score are 0.95, 0.94, and 0.95 respectively. Besides, SVM with word2vec (SG) is giving the highest accuracy for new title classification among all traditional machine learning models.

Between tradition machine learning based models and neural network based models, we find that neural networks are performing well. The best model among all these models is 'Bidirectional LSTM' with word2vec (SG).

4.4.2 Comparison among different feature extraction techniques.

We discover that the 'Skip-gram' mechanism of word2vec is better than the 'Continuous Bag of words' (CBOW) mechanism.

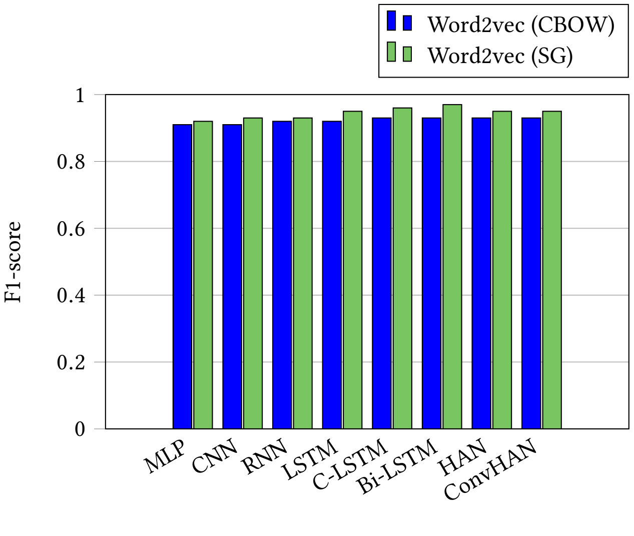


Figure 2: F1-score of different neural network based models with two Word2vec techniques for news article classification

Table 2: Performance Measure of Different deep learning approaches

Feature	Model	News Article Classification			News Title Classification		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Word2vec (CBOW)	Multi-layer Perceptron	0.91	0.90	0.91	0.82	0.8	0.81
	CNN	0.91	0.92	0.92	0.84	0.81	0.82
	Simple RNN	0.92	0.92	0.92	0.84	0.85	0.84
	LSTM	0.93	0.92	0.92	0.85	0.86	0.85
	C-LSTM	0.94	0.92	0.93	0.87	0.88	0.87
	Bi-LSTM	0.93	0.94	0.93	0.87	0.87	0.87
	HAN	0.93	0.94	0.93	0.86	0.85	0.85
	Convolutional HAN	0.93	0.93	0.93	0.86	0.86	0.86
Word2vec (SG)	Multi-layer Perceptron	0.92	0.93	0.92	0.84	0.85	0.84
	CNN	0.93	0.93	0.93	0.86	0.85	0.85
	Simple RNN	0.94	0.93	0.93	0.85	0.86	0.85
	LSTM	0.95	0.95	0.95	0.87	0.88	0.87
	C-LSTM	0.96	0.97	0.96	0.89	0.91	0.90
	Bi-LSTM	**0.97**	**0.98**	**0.97**	**0.91**	**0.91**	**0.91**
	HAN	0.95	0.95	0.95	0.88	0.87	0.87
	Convolutional HAN	0.95	0.96	0.95	0.88	0.86	0.87

Table 3: Performance Measure of Different traditional machine learning approaches

Feature	Model	News Article Classification			News Title Classification		
		Precision	Recall	F1-score	Precision	Recall	F1-score
TF-IDF	Logistic Regression	0.94	0.93	0.93	0.73	0.79	0.74
	KNN	0.85	0.83	0.84	0.76	0.74	0.75
	Decision Tree	0.77	0.78	0.77	0.43	0.66	0.52
	Random Forest	0.91	0.89	0.90	0.70	0.73	0.72
	Naive Bayes	0.88	0.86	0.87	0.79	0.78	0.78
	SVM	0.94	0.94	0.94	0.80	0.79	0.80
	Ada-Boosting	0.89	0.89	0.9	0.66	0.68	0.67
Word2vec (SG)	Logistic Regression	**0.95**	**0.94**	**0.95**	0.81	0.8	0.8
	KNN	0.87	0.87	0.87	0.77	0.77	0.77
	Decision Tree	0.79	0.78	0.78	0.65	0.63	0.64
	Random Forest	0.91	0.92	0.91	0.83	0.82	0.82
	Naive Bayes	0.75	0.76	0.75	0.65	0.66	0.65
	SVM	0.94	0.93	0.94	**0.82**	**0.81**	**0.82**
	Ada-Boosting	0.91	0.91	0.91	0.81	0.80	0.80
Category based Word Frequency	Logistic Regression	0.89	0.89	0.89	0.69	0.74	0.72
	KNN	0.91	0.91	0.91	0.73	0.75	0.74
	Decision Tree	0.9	0.91	0.9	0.71	0.74	0.72
	Random Forest	0.92	0.92	0.92	0.75	0.76	0.76
	Naive Bayes	0.85	0.87	0.86	0.73	0.62	0.67
	SVM	0.83	0.85	0.84	0.67	0.72	0.69
	Ada-Boosting	0.90	0.90	0.90	0.73	0.75	0.74
Manual LIWC	Logistic Regression	0.76	0.74	0.75	0.65	0.67	0.66
	KNN	0.74	0.74	0.74	0.57	0.59	0.58
	Decision Tree	0.61	0.65	0.63	0.50	0.47	0.48
	Random Forest	0.73	0.73	0.73	0.63	0.63	0.63
	Naive Bayes	0.73	0.72	0.73	0.61	0.61	0.61
	SVM	0.63	0.65	0.64	0.53	0.49	0.51
	Ada-Boosting	0.73	0.72	0.72	0.62	0.62	0.61

Figure 2 presents the bar chart of F1-score of different neural network based models with 'SG' and 'CBOW' techniques of word2vec for news article classification.

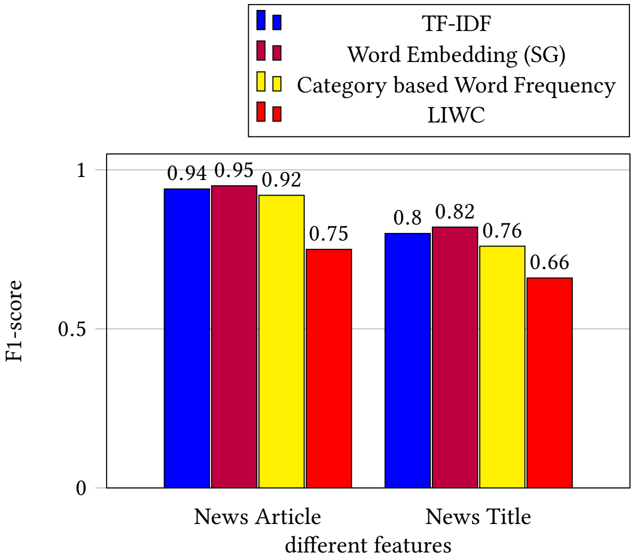


Figure 3: Maximum F1-score of different features applied for machine learning models for both news article and title classifications

Figure 3 depicts the maximum f1-score of four features for different traditional machine learning approaches. Also, we find that word embedding or word2vec is giving maximum f1-score, although different feature extraction techniques work well for different machine learning based models.

Table 4: Comparative performance analysis with other state-of-the-art models for news article classification

Feature	Model	Precision	Recall	F1-score
Word2Vec (SG)	**Bi-LSTM**	**0.97**	**0.98**	**0.97**
Word2Vec [1]	NN	0.96	0.96	0.96
TF-IDF [13]	SVM	0.9256	0.9258	0.9257

4.4.3 *Comparison with other state-of-the-art models.*

Table 4 is showing the comparative performance analysis among other state-of-the-art works for news article classification. A recent work [1] gets 96% precision from neural network based model with word2vec. They utilize 3,76,226 news articles of five classes for their model. Another work [13] gets 92.56% precision from SVM with TF-IDF by utilizing 31,908 new article of twelve categories.

We get 97% precision using 'Bi-LSTM' with Word2vec (SG) that outperforms the state-of-the-art works related to Bangla news article classification.

4.4.4 *News article and News title specific performance.*

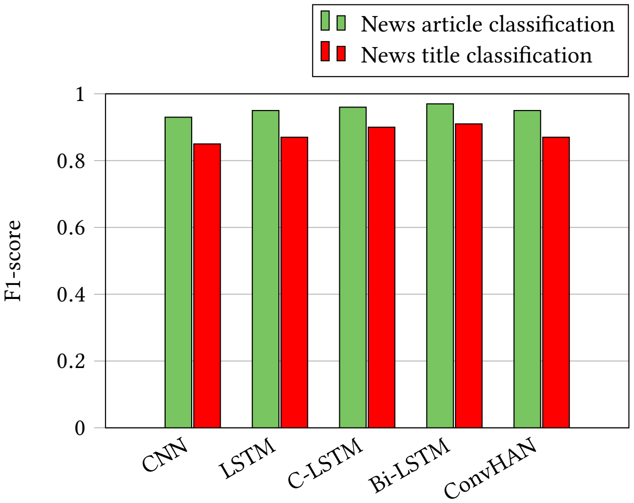
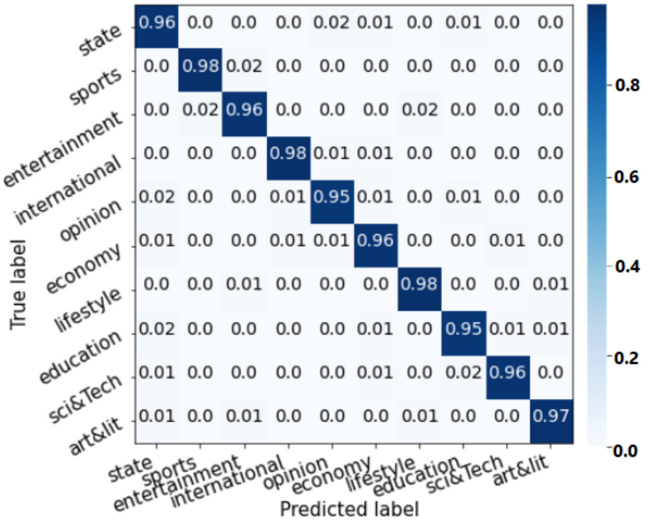


Figure 4: F1-score of different neural network based models with Word2vec (SG) techniques for both news article and title classification

We can observe from Figure 4 that for detecting news genre, news article based classification is more effective than news title based classification. This is because the length of the news articles is too much bigger than that of news titles. Figure 4 presents five neural network based models with topmost F1-scores. From all these models, we find the same scenario. News articles are so much elaborated. So, one should not judge a piece of news by depending on the news title.

4.5 Result Summary



	state	sports	entertainment	international	opinion	economy	lifestyle	education	sci&Tech	art&lit
state	0.96	0.0	0.0	0.0	0.02	0.01	0.0	0.01	0.0	0.0
sports	0.0	0.98	0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
entertainment	0.0	0.02	0.96	0.0	0.0	0.0	0.02	0.0	0.0	0.0
international	0.0	0.0	0.0	0.98	0.01	0.01	0.0	0.0	0.0	0.0
opinion	0.02	0.0	0.0	0.01	0.95	0.01	0.0	0.01	0.0	0.0
economy	0.01	0.0	0.0	0.01	0.01	0.96	0.0	0.0	0.01	0.0
lifestyle	0.0	0.0	0.01	0.0	0.0	0.0	0.98	0.0	0.0	0.01
education	0.02	0.0	0.0	0.0	0.0	0.01	0.0	0.95	0.01	0.01
sci&Tech	0.01	0.0	0.0	0.0	0.0	0.01	0.0	0.02	0.96	0.0
art&lit	0.01	0.0	0.01	0.0	0.0	0.0	0.01	0.0	0.0	0.97

Figure 5: Confusion matrix for news article classification using Bidirectional LSTM with word2vec using Skip-gram method

In our research, we find that 'Bidirectional LSTM' with word2vec (SG) is giving the best results for both news article classification and news title classification. We can observe the confusion matrix for news article classification using 'Bidirectional LSTM' model with word2vec (SG) in Figure 5. 'Bidirectional LSTM' keeps track of both previous and next events. It also uses LSTM which includes long term memory. So, it is showing it's effectiveness for text classification. Besides, the Skip-gram technique for word2vec predicts the best context words from a given center word. The confusion matrix presents the robustness of this model based on test data.

From this confusion matrix at Figure 5, we can find the relationship of news articles among different categories or classes. For example, there is a relationship among 'state', 'opinion' & 'economy' news. We can observe that 'sports' and 'entertainment' news are correlated. That is because sports give us entertainment, although they are fully different categories. Again there exists a relationship between 'education' and 'science & technology' news.

5 CONCLUSION

In this work, we have applied traditional machine learning, deep learning, and attention-based models to identify multi-class genre not only from news contents or articles but also from news titles for Bangla language. We have redacted extensive experiments against ten categories and compared the performances of all classifiers. We create our labeled Bangla news article & title dataset of almost six years from five popular online news portals of Bangladesh. From our experiment, we find that 'Bidirectional LSTM' with word2vec feature, using the Skip-gram method, performs better than other supervised models with other lexical features. We get 98% accuracy for detecting news genres from news articles and 91% accuracy from news titles. We also find that detecting news genre from news articles is more effective than news titles. We think that it can be a benchmark study, compared to state-of-the-art works in this field. In the future, we have a scheme to enlarge our dataset which can be used to increase accuracy more and more. We also have a plan to implement a more improved 'Linguistic Inquiry & Word Count' function for text analysis for Bangla language. Again our created corpus can be used in different areas of natural language processing.

REFERENCES

[1] Md Tanvir Alam and Md Mofijul Islam. 2018. BARD: Bangla Article Classification Using a New Comprehensive Dataset. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 1–5.

[2] Vishwanath Bijalwan, Vinay Kumar, Pinki Kumari, and Jordan Pascual. 2014. KNN based machine learning approach for text and document mining. *International Journal of Database Theory and Application* 7, 1 (2014), 61–70.

[3] Jingnian Chen, Houkuan Huang, Shengfeng Tian, and Youli Qu. 2009. Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications* 36, 3 (2009), 5432–5435.

[4] Ankita Dhar, Niladri Sekhar Dash, and Kaushik Roy. 2018. Application of tf-idf feature for categorizing documents of online bangla web text corpus. In *Intelligent Engineering Informatics*. Springer, 51–59.

[5] Ankita Dhar, Himadri Mukherjee, Niladri Sekhar Dash, and Kaushik Roy. 2018. Performance of classifiers in bangla text categorization. In *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*. IEEE, 168–173.

[6] Mohamed El Kourdi, Amine Bensaid, and Tajje-eddine Rachidi. 2004. Automatic Arabic document categorization based on the Naïve Bayes algorithm. In *proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*. 51–58.

[7] Ethan Fast, Binbin Chen, and Michael S Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 4647–4657.

[8] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM. (1999).

[9] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.

[10] Lalit Gupta, Mark McAvoy, and James Phegley. 2000. Classification of temporal sequences via prediction using the simple recurrent neural network. *Pattern Recognition* 33, 10 (2000), 1759–1770.

[11] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. 2009. Multi-class adaboost. *Statistics and its Interface* 2, 3 (2009), 349–360.

[12] MS Islam. 2009. Research on Bangla language processing in Bangladesh: progress and challenges. In *8th international language & development conference*. 23–25.

[13] Md Islam, Fazla Elahi Md Jubayer, Syed Ikhtiar Ahmed, et al. 2017. A comparative study on different types of approaches to Bengali document categorization. *arXiv preprint arXiv:1701.08694* (2017).

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

[15] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).

[16] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[17] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.

[18] Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. 2015. Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE, 136–140.

[19] Zhijie Liu, Xueqiang Lv, Kun Liu, and Shuicai Shi. 2010. Study on SVM compared with the other text classification methods. In *2010 Second international workshop on education technology and computer science*, Vol. 1. IEEE, 219–222.

[20] Ashis Kumar Mandal and Rikta Sen. 2014. Supervised learning methods for bangla web document categorization. *arXiv preprint arXiv:1410.2045* (2014).

[21] Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, Vol. 752. Citeseer, 41–48.

[22] Scott Menard. 2002. *Applied logistic regression analysis*. Vol. 106. Sage.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[24] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2020. Deep learning based text classification: A comprehensive review. *arXiv preprint arXiv:2004.03705* (2020).

[25] Sankar K Pal and Sushmita Mitra. 1992. Multilayer perceptron, fuzzy sets, classifiaction. (1992).

[26] Ajay S Patil and BV Pawar. 2012. Automated classification of web sites using Naive Bayesian algorithm. In *Proceedings of the international multiconference of engineers and computer scientists*, Vol. 1. Citeseer, 519–523.

[27] Pratiksha Y Pawar and SH Gawande. 2012. A comparative study on different types of approaches to text categorization. *International Journal of Machine Learning and Computing* 2, 4 (2012), 423.

[28] K Rajan, Vennila Ramalingam, M Ganesan, S Palanivel, and B Palaniappan. 2009. Automatic classification of Tamil documents using vector space model and artificial neural network. *Expert Systems with Applications* 36, 8 (2009), 10914–10918.

[29] S Rasoul Safavian and David Landgrebe. 1991. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics* 21, 3 (1991), 660–674.

[30] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[31] Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9, 3 (1999), 293–300.

[32] Vincent Tam, Ardi Santoso, and Rudy Setiono. 2002. 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. IEEE, 235–238.

[33] Rini Wongso, Ferdinand Ariandy Luwinda, Brandon Christian Trisnajaya, Olivia Rusli, et al. 2017. News article text classification in indonesian language. *Procedia Computer Science* 116 (2017), 137–143.

[34] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. 1480–1489.

[35] Min-Ling Zhang and Zhi-Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition* 40, 7 (2007), 2038–2048.

[36] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630* (2015).