A Benchmark of Machine Learning and Deep Learning Algorithms for Detecting Fake News in Bangla Language

Nirjas Mohammad Jakilim, Sm Mahamudul Hasan and Enamul Hassan

October 29, 2022

A Benchmark of Machine Learning and Deep Learning Algorithms for Detecting Fake News in Bangla Language

***Abstract*—Due to the ease with which information may be obtained and the exponential growth in the amount of information available on the internet, it has become more challenging to differentiate between false and genuine information. Any of these fake news websites may easily infect people due to their fabricated claims. This situation has a significant impact on the offline community in general. As a result, interest in this subject has grown. A critical study has been conducted on identifying fake news in English and other languages, save for a few in Bangla. Our research shows an experimental benchmark investigation into identifying false news on a Bengali news website since there is less work in this domain. This research analyses 11434 fake news and true news in the Bengali language and evaluates the performance of machine learning and deep learning algorithms to create a benchmark for detecting Bangla fake news. This research compares the model's performance to a variety of linguistic characteristics and word vectorizers. The best accuracy obtained for lemmatized text is 95.45% for TF-IDF with the SGD classifier and 95.10% for the count vectorizer with the MLP classifier. For stemmed text, we received the best accuracy of 94.9% for the Count Vectorizer with MLP classifier and 94.83% for TF-IDF with MLP classifier. Among deep learning models, RNN gave the best performance with 96.55% accuracy where the f1 score is 0.96. The pre-trained Bangla BERT model gave an F1-Score of 0.96 and showed an accuracy of 93.35%.**

***Index Terms*—Benchmark Analysis, Fake News Detection, Bangla Fake News, Machine Learning Algorithms, Deep Learning Algorithms, TF-IDF, RNN, Bangla BERT Model, CountVectorizer, NLP, Word2vec**

I. Introduction

The term "fake news" is a misnomer that undermines verifiable and public-interest reporting, or, in other words, produces news, which is vital in modern society.[1] Recently, a great number of deceptive news stories with varied political and economic goals have emerged. It has spread all over the internet because more and more people are using social media sites. [2] The uncontrolled flow of information through social media platforms such as Twitter, Facebook, YouTube, and microblogging is one of the greatest barriers to preventing the spread of misleading information or fake news. According to a recent survey, nearly one in three individuals in Spain, the United States, Germany, South Africa, South Korea, the United Kingdom, and Argentina claim to have witnessed COVID-19.[3] Some opponents have recently referenced the most recent flood of inaccurate and unreliable information concerning the COVID-19 epidemic as a result of the widespread of faulty and unreliable information. Even the dissemination of erroneous information is democratic. During the six-week 2016 US presidential election campaign, 25% of Americans visited a fake news website. [4] This circumstance has been recognized as a factor that influences the result. Ramu, the Bangladeshi tragedy of 2012, is a textbook example of a near-fatal disaster. 25,000 individuals engaged in the destruction of the Buddhist temple in 2012, according to a fictitious Facebook post by an unknown person. The area is home to around 12 Buddhist temples and monasteries, and the angry mob set fire to fifty residences [5]. Fake news articles, especially ones including blasphemy, frequently repeat similar patterns. Due to disinformation, numerous horrific occurrences have occurred in Bangladesh. July 2019 saw five deaths and ten injuries [6]. Throughout the construction of the Padma Bridge, people were sacrificed [7]. Since fake news is spread so often, it confuses people and makes it harder for them to figure out what is real news. When information is falsified, it becomes crucial to provide a mechanism for verifying its veracity. The literary style is as varied as the subject matter. False news has been detected by a variety of methods to date. The vast majority of strategies proposed in the literature for recognizing false news are classification-based [8]. In the internet media, fake news stories with logical conclusions and factual reasons for incorrect information are generated. On the other hand, these websites aren't good enough because they can't respond quickly enough to situations where fake news is being spread.

In this study, we evaluated and extended a dataset for identifying fake news, attempted to identify false news features, and built an API and Android application based on our model. This Android app can tell if a Bangla story is fake news or not and if it is true.

The rest of the content is organized in the following way. In Section II, we summarise related work. Our overall approach to this research is discussed in Section III. The dataset's characteristics are discussed in Section IV. The methodology of the overall process is discussed in Section V. In Section VI, we describe the experiment section of our research. The results and analyses are discussed in Section

VII, and the conclusion of our research is shown in Section VIII.

II. Related Works

Fake news is a misnomer that discredits verifiable and public interest reporting or, to put it another way, produced news, which is critical in today's society. [9] Deceptive news has lately emerged in huge quantities for different political and economic goals. [10] Academics are looking at the problems that internet users are encountering as a consequence of the increasing frequency and amount of fake news. Kai Shu and colleagues [11] developed the Social Article Fusion (SAF) approach to detect false news by integrating linguistic characteristics of news content with social context data. In order to detect false news and train various types of machine learning and deep learning NLP models, Zobaer et el.[12] developed a comprehensive Bangla dataset. Approximately 1K fake news datasets and 48K real news annotated datasets are available on their website. By combining the techniques of TF-IDF and Word2Vec, Sharma et el.[13] have created a hybrid text document extraction approach that is capable of accurately determining whether or not a text document is written in Bangla is a satire or a fake using traditional CNN architecture. In order to detect false news in Bangla, Gulzar et el.[14] utilized MNB and SVM classifiers. Their judgment is based on information gathered from social media sites such as Facebook and Twitter. Precision is improved by using a linear kernel in SVM, which is superior to MNB's accuracy of 93.32 percent. To collect data from social networking sites, Islam et el.[15] employ comment extractors, which remove punctuation marks as well as numerical meaning and emoticons from the data. This results in an error-free text corpus. By employing the TF-IDF vectorizer, they were able to gather characteristics from the text corpus. The Naive Bayes Classifier, which is commonly used for spam detection, was utilized to train the processing results. In order to determine the sources of fake news, several academics have turned to graph analysis. Using network dispersion models, Shu et al.[16] demonstrated how to trace the provenance of nodes and the sources of erroneous information. One of the participants, Della Vedova et al,[17] showed an innovative machine learning approach for combining news and social data. They tested the code in another open app before deploying it on the Facebook Messenger chatbot. Their own Twitter and Facebook data collectors, on the other hand, allow them to achieve high accuracy.

III. Our Approach

The proposed study makes use of nine machine learning algorithms and deep learning methods to evaluate the dataset. Before evaluation, the dataset has been preprocessed using a stemmer and a lemmatizer. We used both techniques with the TFIDF and Counvectorizer feature extractors to bring out the best result among them. Figure 1 depicts a total visual representation of our approach.

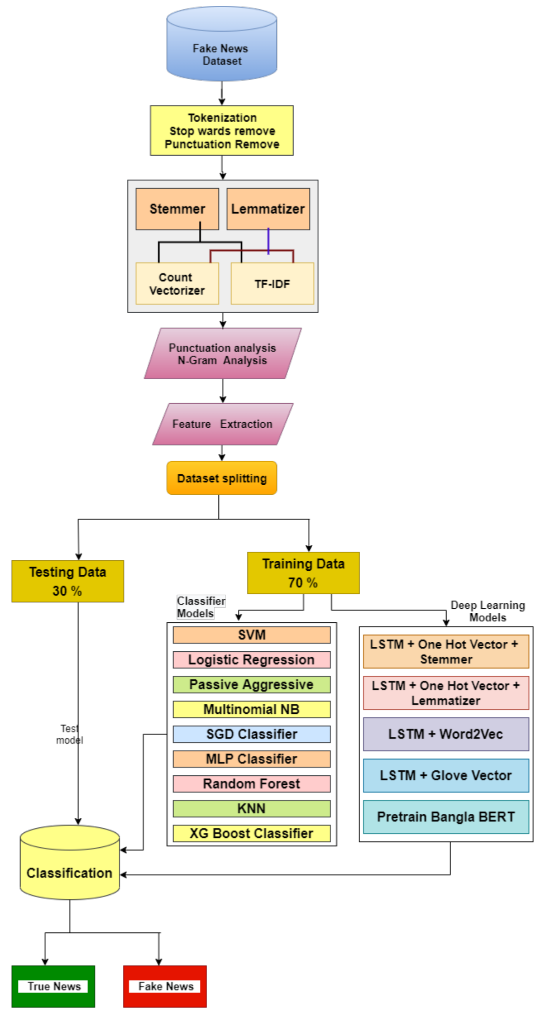

Fig. 1: Our Approach

IV. Dataset Description

The dataset was collected from a previously published research work [18] and it was hugely imbalanced. It mostly consists of 50K true news stories and 1299 fake news stories. Also, it contained a huge amount of satirical news. So, we have added an additional 137 real-world fake news and also decreased the true news to 10K to make it a little more balanced. The dataset contains three types of Fake news they are Propaganda News, Clickbait, and Satire. The final dataset contains 10000 true news and 1434 fake news. Our dataset has the following features.

- **articleID**: Unique id of the news.
- **domain**: Domain of the site that provides this news.
- **date**: News publication date.

- **category**: Category of the news.
- **source**: The person or source who has verified or published the news.
- **relation**: News is related to the headline or not.
- **headline**: Headline of the news.
- **content**: Body text of the news.
- **label**: Target of the news. The value is 0 for fake news and 1 for true.
- **F-type**: Fake news type whether it is clickbait, satire or misleading fake news.

TABLE I: Dataset Content

Name of categories	Fake News	True News
Crime	43.0	144.0
Editorial	0.0	673.0
Education	30.0	181.0
Entertainment	112.0	551.0
Finance	3.0	236.0
International	134.0	1392.0
Lifestyle	108.0	176.0
Miscellaneous	663.0	444.0
National	152.0	3814.0
Politics	100.0	644.0
Sports	60.0	1600.0
Technology	29.0	145.0

The dataset was then analyzed further to establish the prevalence of fake and accurate news within each category. Table 1 displays the number of news stories in each category. The data shows that the majority of fake news falls into the ”Miscellaneous” category. That is to say, they are not focused on a specific topic. They are more diverse and cover a variety of themes. We've also discovered that the entertainment industry is a major source of false news. This is due to the rise of satirical fake news.

V. Methodology

A. Data Preprocessing

To avoid any biased results and get a better classification, text preprocessing is a must. We have preprocessed our dataset in various approaches. We have used our custom-made stopwords list to remove the stopwords from the dataset. As the stopwords can't bear any significance to identify the fake news. Our data preprocessing includes these steps:

- **Removing Stopwards**: We utilized our custom-made built-in stopwords list to remove the stopwords from the dataset because NLTK and other major NLP tools do not offer a built-in Bengali stopword corpus, so we had to create one ourselves.
- **Cleaning Dataset**: We went on to scan the entire dataset for punctuation marks and other invalid characters, which we found and deleted.
- **Stemming**: We use BNLTK stemmer to utilize semantic word embedding and pre-trained word vectors in our analysis. [19]
- **Lemmatizing** : We utilized the BNLTK POS tagger to POS tag the words in our document and lemmatized the words using a Banglakit lemmatizer. [20]

B. Exploratory Analysis

After preprocessing, we evaluated the dataset extensively to find out any particular pattern in the case of fake news and true news. We have used bi-gram and tr-gram analysis to find out the correlation between words in fake and true news. Table 2 shows a comprehensive report of our bi-gram and tri-gram analysis.

TABLE II: Bi-gram and Tri-gram model Output

Bi-gram Fake world	Count	Trig-ram Fake world	Count
(আওয়ামী, লীগ)	2714	(প্রধানমন্ত্রী, শেখ, হাসিনা)	365
(এশিয়া, কাপ)	1007	(ভারপ্রাপ্ত, কর্মকর্তা, (ওসি))	365
(প্রধানমন্ত্রী, শেখ)	733	(থান, ভারপ্রাপ্ত, কর্মকর্তা)	339
(খালেদা, জিয়)	726	(আওয়ামী, লীগ, সম্পাদক)	321
(শেখ, হাসিনা)	725	(বৃহস্পতিব, (, সেপ্টেম্বর))	315
(জাতী, ঐক্য)	658	(জাতীয়, সংসদ, নির্বাচন)	285
(সংসদ, সদস্য)	520	(মেডিকেল, কলেজ, হাসপাতাল)	273
(সংবাদ, সম্মেলন)	516	(ডিজিটাল, নিরাপত্তা, আইন)	269
(গত, সেপ্টেম্বর)	486	(জেলা, আওয়ামী, লীগ)	265
(ড., কামাল)	480	(আওয়ামী, লীগ, সভাপতি)	240
(নাম, এক)	475	(বাংলাদেশ, সময়:, ঘণ্টা,)	219
(ঘটনা, ঘটে।)	471	(এসব, কথা, বলেন।)	208
(লাখ, টাকা)	470	(সময়:, ঘণ্টা,, সেপ্টেম্বর)	206
(ভারপ্রাপ্ত, কর্মকর্তা)	450	(ড, কামাল, হোসেন)	191
(গত, বছর)	419	(উপজেলা, আওয়ামী, লীগ)	189
(আফগানিস্তান, বিপক্ষ)	418	(প্রধানমন্ত্রী, শেখ, হাসিনা।)	180
(সংসদ, নির্বাচন)	412	(সাবেক, প্রধান, বিচারপতি)	178
(ডিজিটাল, নিরাপত্তা)	408	(উপজেলা, স্বাস্থ্য, কমপ্লেক্স)	175
(সাকিব, আল)	402	(আওয়ামী, লীগ, নেতা)	170
(প্রধান, বিচারপতি)	400	(মির্জা, ফখরুল, ইসলাম)	165

We have then further evaluated the dataset and compared the average length of words for both fake and true news and found the following.

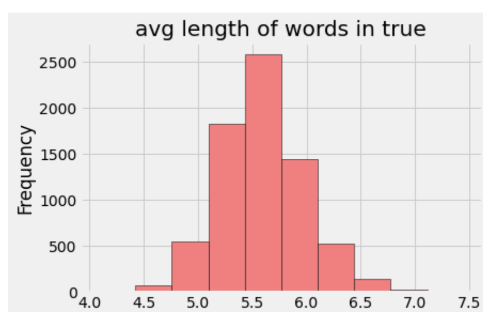

Fig. 2: Average length of words in True News

If we closely look at this then we can see that true news has more word frequency than fake news. That means fake news is not too long. They are typically shorter. Also if we look at the average length of the words then we can see true news has a more dynamic length of words than fake news. That means fake news is usually restricted to short and almost static lengths of words. That's why there are almost 400 words in the same length range.

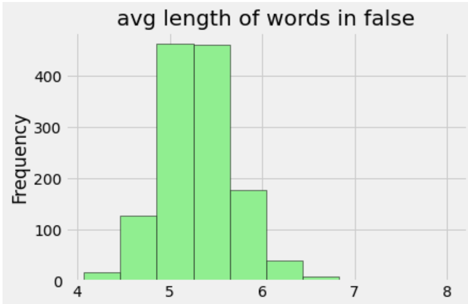

Fig. 3: Average length of words in False News

as growing satire news contains a lot of punctuation marks and punctuation mistakes, we have also evaluated the dataset to find any pattern in the frequency of punctuation marks in true and fake news and the results are as follows.

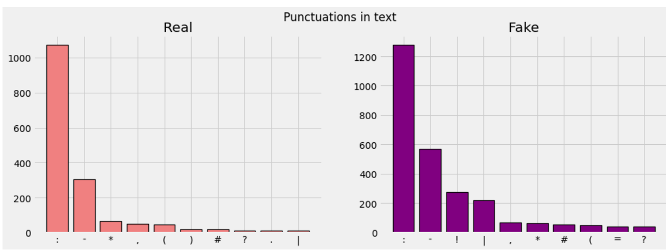

Fig. 4: Distribution of punctuation marks

Figure 4 clearly shows the increasing usage of punctuation marks in fake news. They are mostly used in satirical news, so they can be used as a common factor to distinguish satirical fake news.

C. Feature Extraction

The method of extracting features from a dataset is known as feature extraction.[21] We must convert our textual data to word vectors in natural language processing, and the task is handled in the feature extraction process. Feature extraction is essential because extracting useful features plays a significant role in machine learning classification. There are various feature extraction approaches available. However, we chose the following feature extraction models to extract features from the dataset.

- **TF-IDF:** The TF-IDF evaluates the significance and relevance of a word in a document. The IDF stands for "Inverse-Document Frequency," while TF stands for "Term Frequency." It doesn't only look at how often a term appears in a document. It also considers how many times the word appears in the whole corpus. In this case, TF is used to score the frequency of a word in an article, whereas IDF, is used to achieve the frequency of the entire dataset. The following formula is used to determine the TF-IDF score:

$$\text{tfidf} = tf(t,d) * idf(t,D) \tag{1}$$

- **CountVectorizer:** The complete information is converted to a vector using CountVectorizer. It saves the word's frequency as a vector rather than the TF-IDF score. The vector's dimension is comprised of unique words. The CountVectorizer then counts every word in the dataset and puts the result in the vector's corresponding field.
  For example: Let's consider the following lines in a dataset.
  - রহিম করিমের বন্ধু (text1)
  - কুদ্দুস রহিমের বাবা (text2)

  Then the vectors will be as follows

TABLE III: CountVectorizer

	কুদ্দুস	রহিম	করিম	এর	বাবা	বন্ধু
text1	0	1	1	1	0	1
text2	1	1	0	1	1	0

- **Word2vec:** While the TF-IDF vectors also create text vectors, they cannot adequately capture the context, while the word2vec vector attempts to extract the document's semantic connections. The cosine of the angle between two vectors was applied to determine similarity, referred to as "Cosine Similarity." And the word2vec algorithm is implemented using the CBOW (Continuous Bag Of Words) and Skip-gram models. Additionally, we experimented with a second set of pre-trained 100-dimensional word vectors trained using Word2Vec on 20K Bengali news. Also, later, we utilized the Bengali Glove Vector, trained on 20 million Wikipedia tokens. [22]

VI. Experiments

A. Machine Learning Models

We did a lot of experiments on traditional machine learning models to observe their performances. Among those models, some notable models are Support Vector Machine, Logistic Regression, Multinomial Naive Bayes, Random Forests, and MLP Classifier, SVM, Different tweaking parameters are used to get the best possible performance of a model. We have evaluated all those models against all possible scenarios like word-based embeddings, character-based embeddings, stemmed text, and lemmatized text to record the best performances. We examined the performance of several tuning parameters C in the support vector machine model and discovered that the linear SVM kernel with C=100 performs the best in classifying false news. We selected entropy as our criterion for enhancing Random Forest classification since it provides the greatest results. Following that, we increased the tree depth until we obtained the optimal result, which occurred at max depth=400. We utilized Multinomial Naive Bayes with several alpha settings ranging from 0 to 1 and obtained the best result at *alpha*=0.01.

We may see the model's performance versus the adjusting parameter "alpha" in the following image. As our Neural Network classifier, we utilized the built-in MLP classifier from the sklearn package. To fine-tune the model and assess it across all scenarios, we utilized the textbf hidden layer sizes= (33) and max iter=500 parameters.

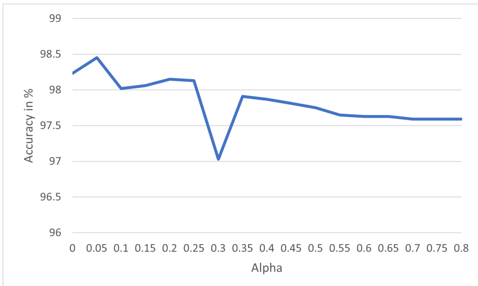

Fig. 5: Change of Alpha value in MNB

To construct the classification model, we combine a vectorizer and a classifier in a pipeline. For CountVectorizer and TFIDF Vectorizer with stemmed text, we use a test size of 0.3 and a random state of 0. For KNN, we choose a K value of 7, and for the passive aggressive classifier, we use a maximum iteration value of 10. For the MLP classifier, we utilize a hidden layer of 33 and a maximum iteration value of 500.

B. Deep Learning Models

- **LSTM with One Hot Vector:** Due to its capacity to effectively collect sequential information, LSTM models are used in text classification and generation tasks. That is why we conducted our evaluation using a sequential LSTM model. We used the 600 embedding vector feature in conjunction with the "sigmoid" as our activation function. We assessed the model using stemmed and lemmatized text with 20 epochs and 256 batches.
- **LSTM with word2vec:** Following our experiments with lexical-based features, we evaluated the LSTM model in conjunction with a pre-trained word2vec model. The model was trained using a 22K-item dataset of Bengali news. Additionally, we utilized an embedded vector dimension of 100 and activated functions such as "Relu" and "sigmoid." We maintained a 70:30 ratio in our dataset. There were 12 epochs.
- **LSTM with Glove Vector:** We have now used a pre-trained Bengali glove vector model that was trained on more than 20 million rows of Wikipedia data.[22] We utilized the identical parameters as previously stated, but with an embedding dimension of 100.
- **Recurrent Neural Network:** Additionally, we employed recurrent neural networks utilizing TensorFlow's built-in functions. As previously, we utilised the activation functions "sequence model" and "relu." Additionally, we tokenized the terms using Bidirectional LSTM and our own self-implemented tokenizer. There were twelve epochs.

C. Pre-Trained Language Models

- **Bengali BERT:** BERT and its variant models, in particular, have outperformed the GLUE benchmark for Natural Language Understanding (NLU). That's why we have decided to use the pre-trained Bangla BERT model[23] for our evaluation. We have used the BERT wrapper over the sklearn library as it slightly performs better than the original hubbingface hosted BERT[24]. We kept 70% of our dataset as a train dataset and 30% as our test dataset and used epoch=10 to train the BERT model. We have also used our custom-made Bengali vocabulary for the BERT. Furthermore, we have used learning_rate=2e-05, num_mlp_hiddens=500, random_state=42, max_seq_length=64 as our model tweaking parameters.

VII. Result and Analysis

We used 70% of the dataset for training and 30% of the dataset for testing the models. After stemming and lemmatizing the news data, we tested all of the models on the stemmed text first, then we further evaluated all the models on the lemmatized text.

A. Machine Learning Models

We discovered that the SGD Classifier has the best accuracy for TF-IDF and lemmatized text. It was 95.45% accurate. As previously said, it's an optimization approach that may optimize the loss for other classifiers like SVMs and Logistic Regression. So it outperforms all others. The MLP classifier came in second with a TF-IDF accuracy of 95.1%. It's for the lemmatized text. MLP is a neural network classifier, which requires a lot of data to train effectively. Maybe MLP would do better if the dataset had more false news. The Passive Aggressive Classifier came in third. It achieved 94.55 percent accuracy using TF-IDF vectorizers. The table below summarises all model results.

Most models perform better for lemmatized text than for stemmed text if we examine them attentively. Unlike a stemmer, a lemmatizer also groups words in their root form. So use comparable word forms. That's why lemmatized text outperformed stemmed text.

Other models worked well for the TF-IDF vectorizer, but only Logistic Regression did well for the CountVectorizer. A count vectorizer accuracy of 94.12% and a TF-IDF vectorizer accuracy of 93.22 are achieved. After evaluating them against the character-based embedding method, we can see that the Support Vector Machine and Passive Aggressive classifier both outperform all models by detecting more than 400 true negatives out of 440 total fake news. That means we can conclude that both models work better for character-based embedding rather

TABLE IV: Overall Performances of the Models for Lemmatizer

	Lemmatizer + CV				Lemmatizer + TF-IDF			
Algorithm	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score	Accuracy
SVM	0.94	0.94	0.94	93.61	0.94	0.94	0.94	94.39
Logistic Regression	0.94	0.94	0.94	94.0	0.93	0.93	0.92	93.1
Passive Aggressive	0.94	0.94	0.94	93.73	0.94	0.95	0.94	94.55
Multinomial NB	0.94	0.93	0.93	92.94	0.94	0.94	0.94	94.16
SGD Classifier	0.94	0.94	0.94	94.24	0.95	0.95	0.95	95.45
MLP Classifier	0.95	0.95	0.95	94.83	0.95	0.95	0.95	95.1
Random Forests	0.93	0.92	0.91	92.47	0.93	0.92	0.91	92.24
KNN	0.87	0.88	0.85	87.61	0.90	0.91	0.89	90.67
XGBoost	0.94	0.94	0.93	93.96	0.94	0.94	0.93	93.77

TABLE V: Overall Performances of the Models for Stemmer

	Stemmer + CV				Stemmer+ TF-IDF			
Algorithm	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score	Accuracy
SVM	0.93	0.93	0.93	93.22	0.94	0.94	0.94	93.81
Logistic Regression	0.94	0.94	0.94	94.12	0.93	0.93	0.93	93.22
Passive Aggressive	0.93	0.94	0.94	93.65	0.94	0.95	0.94	94.51
Multinomial NB	0.94	0.93	0.93	92.9	0.94	0.94	0.94	93.92
SGD Classifier	0.93	0.93	0.93	92.94	0.95	0.95	0.95	94.71
MLP Classifier	0.95	0.95	0.95	94.9	0.95	0.95	0.95	94.83
Random Forests	0.93	0.92	0.91	92.47	0.92	0.92	0.91	91.89
KNN	0.87	0.88	0.85	87.81	0.90	0.91	0.89	90.51
XGBoost	0.94	0.93	0.93	93.41	0.94	0.93	0.93	93.49

than word-based embedding. The below figure shows a comparison of the performances of the models between character-based embedding and work-based embedding.

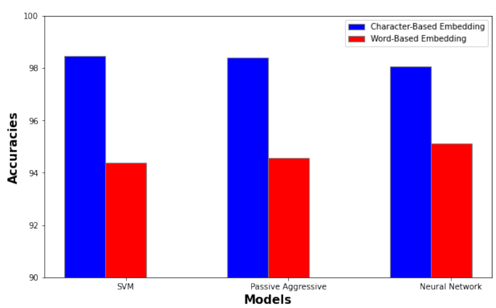

Fig. 6: Model Performances (Character Embedding vs Word Embedding

B. Deep learning Models

As only lexical features can not derive the true context of the news, we have used the word2vec model with deep learning algorithms to get a better classification.

After evaluation of deep learning models like LSTM with different word vectors, RNN, and pre-trained multilingual BERT, RNN gave us the best accuracy which is 96.5%. Moreover, it detected 227 true negatives. The number of false negatives was also pretty much low which was 22. Among 2038 true news, it only detects 57 as false news. The second best performing model was the LSTM with a pre-trained word2vec embedding model. The model gave an accuracy of 94.5%. The multilingual BERT model gave an accuracy of 93.11% after the 3rd epoch but gave an F1-Score of 0.96. Below are shown the performances of deep learning models on the dataset.

TABLE VI: Deep Learning Models with their Performances

Model	Precision	Recall	F1 Score	Accuracy
LSTM+One Hot+ Stemmer	0.93	0.94	0.93	93.50%
LSTM+One Hot+ Lemmatizer	0.94	0.94	0.94	94.11%
LSTM+word2vec	0.94	0.94	0.94	94.5%
LSTM+Glove	0.95	0.95	0.95	94.8%
RNN	0.96	0.96	0.96	96.55%
BERT	0.94	0.98	0.96	93.35%

C. Failed Cases

- **Pre-processor:** If we look closely at our results, we can see that the traditional models can't give us good accuracy when we are using word processors like lemmatizer, stemmer, and POS tagger. That's because there is still no standard POS Tagger, Stemmer, and Lemmatizer. After using the banglakit lemmatizer, we have seen that instead of getting the root form of the word it sometimes breaks the word and can't handle the conjoint letters properly. That's why after processing with these built-in preprocessors, the model shows such poor results.
- **Deep Learning models:** After the traditional machine learning models, we expected to get better results when using semantic word vectors like word2vec and deep learning models like LSTM, RNN, and BERT. But after evaluation, among the models, RNN showed good results but not the expected perfor-

mance. That may be because the dataset is imbalanced and doesn't contain a sufficient amount of fake news. We know that deep learning and neural network models need a huge amount of data to train properly, but after we have added some propaganda-based fake news, the dataset still lacks a sufficient amount of fake news.

VIII. Conclusion

In this paper, we did extensive research on Bengali Fake News detection with the best possible classification models and observed their performances on our self pre-processed data. We have found that classification models perform well for lemmatized text as lemmatized text can provide better semantic context than stemmed text. We have also found that Stochastic Gradient Descent, Logistic Regression, and Passive Aggressive Classifier perform well in Bengali fake news detection. The research also shows that RNN and pre-trained BERT model performed pretty much well but can perform much better if the dataset contained a sufficient amount of fake news. So, the dataset can be extended with more propaganda-based fake news to improve the deep learning classification and avoid any biased results. Another finding of the research is KNN, Random Forests. These classifiers don't perform well in the case of Bengali fake news detection. Another key finding is that only body text and headlines are not enough for fake news detection. So more features or a hybrid approach can be introduced. The research can be helpful for further researchers to improve the dataset and build a hybrid model with more possible features to improve the performance of the models.

IX. Appendix

We have created a custom API based on our machine learning model and developed an Android app using that API. The app was developed using the Flutter SDK, and the API is built using Flask and is temporarily hosted on Heroku. The figure below shows the app in action.

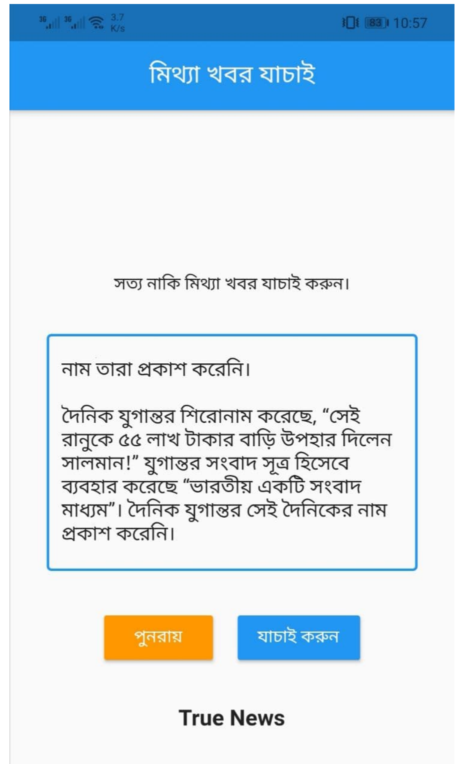

Fig. 7: Detecting True News

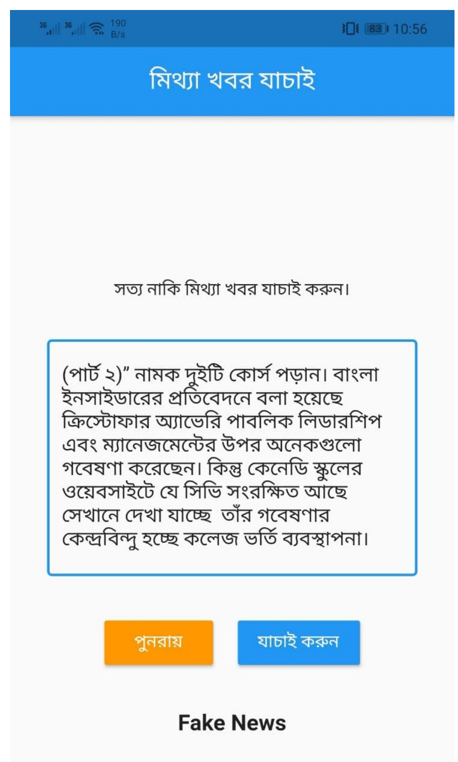

Fig. 8: Detecting fake news

References

[1] D. Tambini, "Fake news: public policy responses," 2017.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[3] N. Fernandes, "Economic effects of coronavirus outbreak (covid-19) on the world economy," 2020.

[4] S. Giusti and E. Piras, "Introduction: In search of paradigms: Disinformation, fake news, and post-truth politics," in *Democracy and Fake News*. Routledge, 2020, pp. 1–16.

[5] B., "Buddhist temples, homes burned, looted in ramu." [Online]. Available: https://bdnews24.com/bangladesh/buddhist-temples-homes-burned-looted-in-ramu

[6] A. News, "A human sacrifice bridge rumour started circling on facebook." [Online]. Available: https://www.abc.net.au/news/2019-07-25/human-sacrifice-bridge-rumours-spark-bangladesh-lynchings/11344914

[7] G. Voices, "Fake news about "human sacrifices" for the padma bridge leads to violence in bangladesh," Jul 2019. [Online]. Available: https://globalvoices.org/2019/07/20/fake-news-human-sacrifice-padma-bridge-violence-bangladesh/

[8] D. Mouratidis, M. N. Nikiforos, and K. L. Kermanidis, "Deep learning for fake news detection in a pairwise textual input schema," *Computation*, vol. 9, no. 2, p. 20, 2021.
[9] Y. h. Tambini, Damian, "Fake news: public policy responses," 2017.
[10] J. L. Egelhofer and S. Lecheler, "Fake news as a two-dimensional phenomenon: A framework and research agenda," *Annals of the International Communication Association*, vol. 43, no. 2, pp. 97–116, 2019.
[11] K. Shu and H. Liu, "Detecting fake news on social media," *Synthesis lectures on data mining and knowledge discovery*, vol. 11, no. 3, pp. 1–129, 2019.
[12] M. Z. Hossain, M. A. Rahman, M. S. Islam, and S. Kar, "Banfakenews: A dataset for detecting fake news in bangla," *arXiv preprint arXiv:2004.08789*, 2020.
[13] A. S. Sharma, M. A. Mridul, and M. S. Islam, "Automatic detection of satire in bangla documents: A cnn approach based on hybrid feature extraction model," in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 2019, pp. 1–5.
[14] M. G. Hussain, M. R. Hasan, M. Rahman, J. Protim, and S. A. Hasan, "Detection of bangla fake news using mnb and svm classifier," *arXiv preprint arXiv:2005.14627*, 2020.
[15] T. Islam, S. Latif, and N. Ahmed, "Using social networks to detect malicious bangla text content," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. IEEE, 2019, pp. 1–4.
[16] K. Shu, H. R. Bernard, and H. Liu, "Studying fake news via network analysis: detection and mitigation," in *Emerging research challenges and opportunities in computational social network analysis and mining*. Springer, 2019, pp. 43–65.
[17] M. L. Della Vedova, E. Tacchini, S. Moret, G. Ballarin, M. DiPierro, and L. de Alfaro, "Automatic online fake news detection combining content and social signals," in *2018 22nd conference of open innovations association (FRUCT)*. IEEE, 2018, pp. 272–279.
[18] A. I. Sudipta Kar, Zobaer Hossain, "Banfakenews." [Online]. Available: https://www.kaggle.com/cryptexcode/banfakenews
[19] Ashwoolford, "Bnltk (bangla natural language processing toolkit." [Online]. Available: https://github.com/ashwoolford/bnltk
[20] Banglakit, "Banglakit lemmatizer: A rule-based lemmatizer for bengali." [Online]. Available: https://github.com/banglakit/lemmatize
[21] D. AI, "Feature extraction definition | deepai," May 2019. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/feature-extraction
[22] S. Sarker, "Glove-bengali: Bengali glove pretrained word vector." [Online]. Available: https://github.com/sagorbrur/GloVe-Bengali
[23] ——, "Banglabert - bengali mask language model for bengali language understading." [Online]. Available: https://github.com/sagorbrur/bangla-bert
[24] Charles9n, "Bert-sklearn: a sklearn wrapper for google's bert model," Feb 2019. [Online]. Available: https://github.com/charles9n/bert-sklearn