International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I5P135
Volume 72 Issue 5, 346-354, May 2024

Original Article

Social Media Bangla Fake News Detection Using Deep and Machine Learning Algorithms

Nipa Rani Das[1], Md. Mahbubul Alam[2], Asif Pervez Polok[3], Mirza Raquib[4], Abdullah Al Mamun[5], Md Jakir Hossen[6]

[1,2,3,4]*Department of Information and Communication Engineering, Noakhali Science and Technology University, Bangladesh.*
[5]*School of Information and Communication, Griffith University, Australia.*
[6]*Faculty of Engineering and Technology, Multimedia University, Malaysia.*

[6]*Corresponding Author: jakir.hossen@mmu.edu.my*

Received: 27 October 2023 Revised: 01 February 2024 Accepted: 15 April 2024 Published: 26 May 2024

Abstract - *In this current world, media and online news publications are spreading rapidly. The dissemination of inaccurate information on social media platforms is increasingly becoming a concern. The proliferation of fake news is due to the ease with which data can be accessed and shared as a result of the mobile technology revolution. Like many countries, fake news is spreading very fast in Bangladesh. The situation gets worse with the spreading of misinformation about epidemics like COVID-19. Created a novel dataset of the Bengali language and achieved to goal by using LSTM and machine learning models. Now, other algorithms are used, but the LSTM and machine learning models have good performance. This program's algorithm to select the attribute, a text feature based on TF-IDF and Word Embedding was used. Focused LSTM-base model and machine learning models, especially the Bangla-LSTM-base model and machine learning models. Finally, add a dense layer as a summary layer responsible for generating summary sentences to text. According to all of the evaluations performed above, the additional Trees Classifier outperformed the other six Machine Learning methods. The accuracy rate for identifying false news in news headline data is roughly 86.14%. The second-best accuracy provided by the Random Forest Classifier algorithm is close to 85%. The third-best accuracy provided by the Decision Tree Classifier method is approximately 84%. Moreover, seeing that deep learning algorithms outperformed machine learning ones. Furthermore, LSTM has a 96.14% training accuracy rate and an 86% testing accuracy rate for identifying false news in news headline data.*

Keywords - *LSTM, Word Embedding, Dataset, Headline, TF-IDF, Accuracy.*

1. Introduction

Today, the World Wide Web can be thought of as a space where people can freely and nearly without any limitations generate and share information. The majority of internet users behave ethically and work to make the internet efficient and secure. Nonetheless, some online users engage in behaviours that can be characterized as antisocial. There are numerous definitions of antisocial behaviours, but generally speaking, it can take one of two major types [1]. The first is the dissemination of false information, which can take many different forms. Another one is that the reactions from specific users, such as conversation manipulation, cyberbullying, or other similar behaviours, represent the other group.

1.1. Problem Statement

Like many countries, fake news is spreading very fast in Bangladesh. A national survey conducted by the Management and Resources Development Initiative (MRDI) with the help of UNICEF shows that 63.6 percent of people in Bangladesh are affected by this misinformation being affected. Some fact-checking websites are used to solve this problem. However, the existing system is not efficient for the Bangla language. An automated system would, therefore, be cleaner and more visible. Additionally, analyzing Bangla letters and words is beyond the scope of the current systems. Because nearly all models developed to date can function with English letters and words. It is imperative to have a model that can recognize words and letters in Bangla. Anyone who receives phony news using a messaging app, such as WhatsApp, Imo, Viber, or another one, is unable to determine if the content is true or false. So, to address all of these shortcomings of the current models, a system that can address all of these issues is required. So, to deal with the limitations of the model, there needs to be a system to solve all problems. For this, deep and machine learning models are used.

1.2. Objectives

People do not have much time these days to read the entire article. They frequently simply learn the wrong things by reading only confusing or misleading headlines.

Hence made the decision to conduct this poll to find bogus news in Bangla. To ensure the integrity of social media news headlines proposed system wishes to fulfill the following objectives:

- To classify the social media news headlines using a deep learning and machine learning algorithm into a true or fake category so people cannot be deceived by false information.
- To publish false information is used to gain political favor, promote businesses and products, and gain revenge. The system aims to make people aware of this.
- To improve the existing system in terms of accuracy and other performance parameters.
- To obtain a clear idea about the intended models and the working principle of the systems.
- To find out the issues and propose solutions for these issues.
- To find out which one is the better model.

2. Literature Review

Some of the previous works are mentioned here: Tasnuba Sraboni et al. (2021) [2] used some feature extraction and pre-processing methods for the dataset. Passive Aggressive Classifier and Support Vector Machine achieved accuracy of 93.8% and 93.5%, respectively, according to the experimental examination of real-world data. Shafayat Bin Shabbir Mugdha (2020) [3] employed TF-IDF to select the tribute. Gaussian Naive Bayes provided 87% accuracy in the model, which is comparable to the greatest performance of any other method used to identify bogus Bangla news. S. Vosoughi et al. (2018) [4] use a novel method to examine the properties of fake news that spreads on social media, how rumors spread on Twitter and compare fake news to legitimate news in terms of Twitter diffusion. Falsehood takes 20 times longer to reach a cascade depth of ten people.

Riedel et al. (2017) [5] Make a competitive argument for a system that categorizes news headlines and the body of the piece to match—the use of Multilayer Perceptron (MLP) classifiers by the authors (TF-IDF). The authors attained an accuracy of 88.46% using a straightforward Multi-Layer Perceptron (MLP).

Ahmed et al. (2017) [6] employed various machine learning models, such as Logistic Regression (LR), Support Vector Machine (SVM), linear support vector machine (LSVM), K-nearest neighbor (KNN), Decision Tree (DT), and Stochastic Gradient Descent (SGD). The SVM and logistic regression models achieved the highest accuracy of 92 percent. Additionally, a false news detection model was developed utilizing n-gram analysis and multiple feature extraction methods. It achieved the maximum accuracy of 92% for unigram features and a Linear SVM classifier. Mykhailo Granik et al. (2021) [7] use a Naive Bayes classifier to illustrate a straightforward method for identifying bogus news. CNN, and ABC News. A 74% accuracy was achieved by these models. Avinash Shakya et al. (2017) [8] used Naive Bayes classifiers, SVM, and semantic analysis for the multidimensionality of fake news. The entire recommended approach is built on AI methods, for telling the real from the phony. Marco L. Della Vedova et al. (2019) [9] introduced a fake news detection system of machine learning (ML, which provides an accuracy of 78.8%. It also implemented its technique in a Messenger Chabot, and it identified phony news 81.7% of the time. UL Haque R et al. (2019) [10] provided a graph-based, semi-supervised method to detect fake news. This graph-based semi-supervised method provides 84% accuracy. A. Thota et al. (2014) [11] use a revolutionary attitude evaluation technique to spot fake news. They used stance detection on two text pieces. For their model, they employed the Fake News Challenge (FNC-1) dataset. They trained the model using TF-IDF. They also used dense neural networks.

V. Perez-Rosas et al. (2014) [12] focus on particular topics or datasets. The most well-known of these is the political sphere. Hence, the CNN algorithm trained the data. The various textual traits that can be used to discern between reliable and false information are examined in this study. M.Z. Rahman et al. (2019) [13], the Lack of datasets is one of the factors contributing to the dearth of studies on false news identification in Bangla. That is currently accessible in Bangla for false news identification even though there are roughly 37.47 times as many true news reports as false ones. Sraboni, T et al. (2019) [14], In the experimental examination of the suggested model on actual data, the Passive Aggressive Classifier gain an accuracy of 93.8%, and the Support Vector Machine find 93.5% accuracy. Based on the [18] dataset, this investigation was also conducted. Wolpert and D.H. (2018) [15] used an efficient method for handling unbalanced datasets in model stacking. While a single classifier is not efficient, to increase model performance there is a strategy known as a stacked generalization.

A. S. Sharma and M. A. Mridul (2017) [16] are required to look into the problems that internet users have because of fake news. Using conventional CNN architecture, S. Sharma et al. presented a mixed extraction method that combines Word2Vec and TF-IDF and can detect whether a Bangla text document is satirical or not, and the precision is more than 96%. Kai Shu et al. (2018) [17] presented Social Article Fusion (SAF), a methodology that integrates social context data with linguistic characteristics of news material to identify false information. They effectively used RNN to record consumers' temporal interactions with the false information and accuracy achieved 82%. Mohamed Torky et al. (2020) [18], Here, presented for identifying and blocking false news and misleading social media material. The results of that the accuracy was about 89%. According to who "likes" a Facebook post, Eugenio Tacchini et al. showed that it is possible to identify hoaxes or not with good performance.

The previous work of this study (2021) deployed two machine learning techniques that worked with an English dataset and produced accuracies surpassing 99% even with training data by using user IDs as features for post-classification. Tecniques were logictic regression and HBLC.

Kai Shu et al. (1992) [19] established the revolutionary Trifn technique to identify bogus news. This solution aims to isolate valuable functionality autonomously from the obligations of news providers and users while also capturing interdependencies concurrently. Here XG Boost classifier model is used.

Marco L. Della Vedova et al. (2019) [9] offered a cutting-edge machine-learning approach that takes into account social media news information. They attain more than 90% accuracy using their respective social media data sets. Roy et al. (2018) [20] took into account when CNN and Bi-LSTM model article representations were supplied into MLP for final classification. Examining the news sources rather than the article's text content can help spot fake news because it can offer insightful information.

3. Methodology

A methodology is a method for creating a procedure and carrying out research. The data collecting, data preprocessing, dataset splitting into training and testing sets, and model construction, which involves training the algorithms using the training dataset, are some of the processes that this study endeavour underwent. The testing dataset was then utilized to determine whether or not the news was accurate. Finally, utilizing a few performance evaluation approaches, the algorithms' performance has been assessed.

3.1. Data Collection

The process of gathering data from various sources is known as data collection. It is the process of gathering, estimating, and analyzing accurate research insights using standard procedures. The most important and initial step in any research project is data collection. Data were collected from Kaggle, which allows one to download any dataset that is wanted, using a BANS dataset from "kaggle.com" that contained 14000 Bangla fake news detections. That also makes use of the BNLPC3 dataset. These datasets are then used to train and evaluate the model.

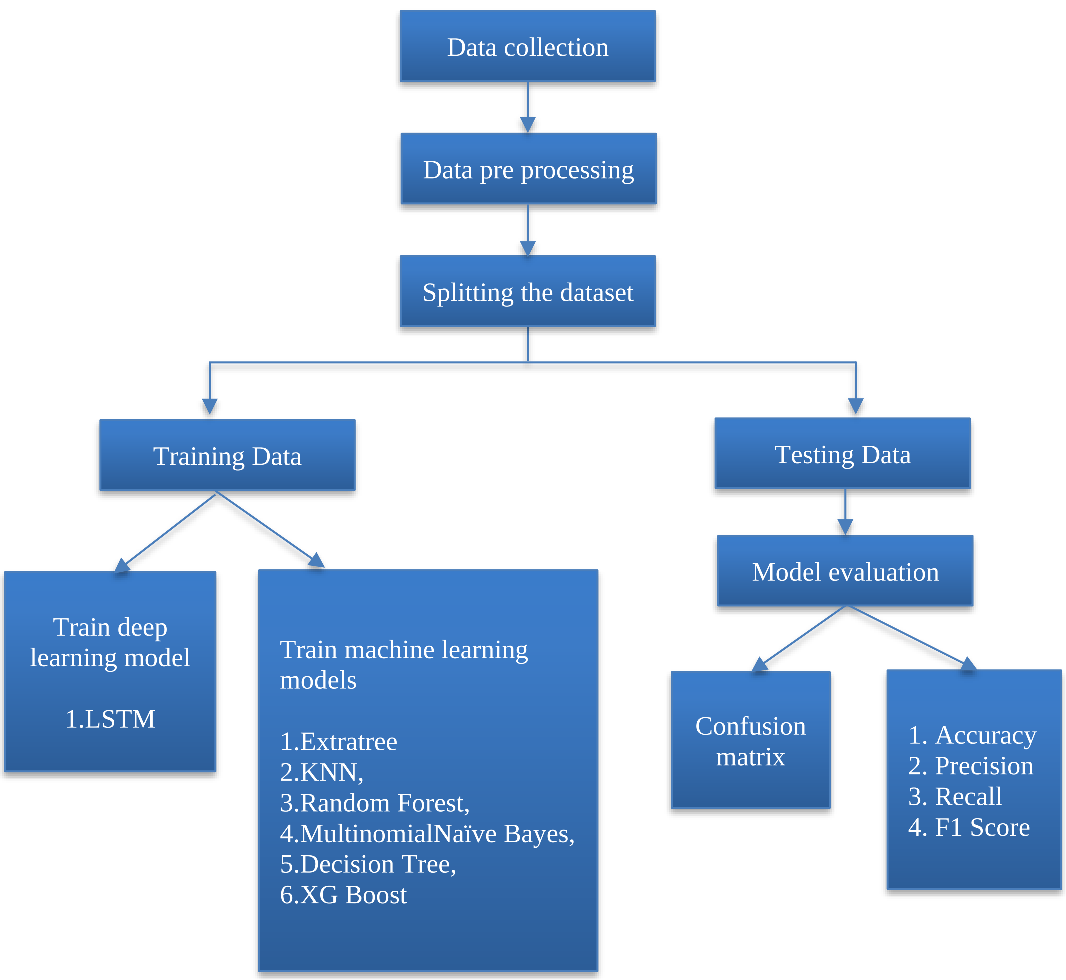

Fig. 1 Workflow of fake news detection model

Before dropping column-

```
In [43]: df.head(5)
Out[43]:
```

	category	headline	content	label
0	National	৮ দিনে ১৮ বিল পাস!	দশম জাতীয় সংসদের মেয়াদ শেষ হয়ে যাচ্ছে। কার্যত ...	0.0
1	Sports	আ'লীগের জনসভায় লোকে লোকারণ্য ফেনী ট্রাংক রোড	একাদশ জাতীয় সংসদ নির্বাচনকে সামনে রেখে সাংগঠনি...	0.0
2	National	মাদ্রাসায় জোড়া খুন: পরিচালক তিন দিনের রিমান্ডে	গাজীপুরে জোড়া খুন মামলার প্রধান আসামি মাদ্রাসা...	0.0
3	Sports	নেপালকে হারিয়ে গ্রুপ চ্যাম্পিয়ন বাংলাদেশ	সাফ অনূর্ধ্ব-১৮ নারী ফুটবল চ্যাম্পিয়নশিপে নেপা...	1.0
4	National	কুড়িগ্রামে ২ শিক্ষার্থীর লাশ উদ্ধার	কুড়িগ্রাম প্রতিনিধি : কুড়িগ্রাম সদর উপজেলার বে...	1.0

Fig. 2 Before dropping the column

After dropping unnecessary columns-

```
In [48]: df.head(5)
Out[48]:
```

	headline	label
0	৮ দিনে ১৮ বিল পাস!	0.0
1	আ'লীগের জনসভায় লোকে লোকারণ্য ফেনী ট্রাংক রোড	0.0
2	মাদ্রাসায় জোড়া খুন: পরিচালক তিন দিনের রিমান্ডে	0.0
3	নেপালকে হারিয়ে গ্রুপ চ্যাম্পিয়ন বাংলাদেশ	1.0
4	কুড়িগ্রামে ২ শিক্ষার্থীর লাশ উদ্ধার	1.0

Fig. 3 After dropping the unnecessary column

3.2. Data Preprocessing

For preprocessing, we needed to remove all the unnecessary words, commas, stop words, and other symbols.

3.2.1. Removing the Unnecessary Column

Working with Headline and Label columns, so dropped the unnecessary columns from the dataset.

3.2.2. Convert the Sentence into a Word

It entails dissecting a string of characters into distinct phrases, words, symbols, and other components. That depicts the process of breaking a statement into words. These entities are referred to as tokens. Tokenizing the raw Bengali text was the subsequent step in the preprocessing procedure. Punctuation marks and other characters were removed during the tokenization process.

3.2.3. Removing Bad Characters, Punctuation, etc

After the text data was collected, the dataset was cleaned by removing non-letter characters, including commas, dots, semicolons, hyphens, underscores, exclamation points, question marks, etc. Any token with a frequency of less than five was removed. The disruptive and undesirable characters were eliminated to help the system work better. This was necessary for the Unicode encoding to function properly.

3.2.4. Bangla Stemmer

The process of stemming involves removing the supplied word's root word. The core concept of stemming is to simplify intricate grammatical and word structures to their simplest forms. Any natural language can inflect words by the rules. Verbal and nominal inflections make up the majority of word inflections in Bengali. The model in this study simply takes into account verb and noun inflection.

3.2.5. Removing Bangla Stopwords

Stopwords are the often-used group of words that do not add significant information to the text's classification. During these processes, words that do not significantly further the meaning of a Bengali sentence were eliminated. One example of deleting Bangla stop words. A stopword list with about 400 words is provided. Social media posts can contain comments that use misspelt words or abbreviations that make it difficult to recognize the terms. Therefore, frequently used stopwords in various spellings have been included in the list.

3.2.6. Word Embedding

In NLP, word embedding means the word representation to analyze the text. These representations encode the meaning of words as they converge closer to the vector space. Various language modeling and feature-learning techniques can be used to achieve word embedding. These techniques are used as training and testing inputs to the deep and machine learning models.

Sample input: আমি ওর হাতগুলি ভেঙে দিয়েছিলাম

(I broke his arms)

Sample output: 'আমি', 'ওর', 'হাতগুলি', 'ভেঙে', 'দিয়েছিলাম'

('I', 'broke', 'his', 'arms')

Fig. 4 Text segmentation

Sample input: 'আমি', 'ওর', 'হাতগুলি', 'ভেঙে', 'দিয়েছিলাম'

('I', 'broke', 'his', 'arms')

Sample output: 'আমি', 'ওর', 'হাত', 'ভেঙে', 'দেই'

('I', 'break', 'his', 'arm')

Fig. 5 Example of bangla stemmer

Sample input: 'আমি', 'ওর', 'হাত', 'ভেঙে', 'দেই'

('I', 'break', 'his', 'arm')

Sample output: 'হাত', 'ভেঙে', 'দেই'

('break', 'arm')

Fig. 6 Example of removal of bangla stopwords

3.2.7. Feature Extraction

Typically, n-grams are taken out of a corpus of content or discourse. N-grams are also known as shingles when the items are words. There are unigrams, bigrams and trigrams based on size. Here, it divides each word with a space to create a bordered succession of n items from a given arrangement of content records. Each statement in a specific content report makes it easier to understand and employ unigrams.

The primary goal of employing n-grams is to turn the archives into a collection of words from which it can quickly get the TF-IDF value. It dropped the remaining data, such as title, date, and subject, and chose the features that are linked to the objective, like headline and label, eliminating the null value.

3.2.8. TF-IDF Vectorizer

Using the data recovery method TF-IDF, the terms recurrence (TF) and its inverse archive recurrence are determined (IDF). The TF and IDF scores are unique to each word or term.

Very well-known term-weighting algorithms are now in use; TF-IDF is employed in 83 percent of content-based recommender systems in computerized libraries. Using a term's raw inclusion in an archive, which counts how many times the term t appears in the archive due to the term recurrence TF, is the simplest option (t, d).

$$idf_i = \log\left(\frac{n}{df_i}\right) \quad (1)$$

TF-IDF can be calculated by multiplication of TF with IDF. So, saying that:

$$w_{i,j} = tf_{i,j} * idf_i \quad (2)$$

3.3. Dataset Splitting and Building Models

Dividing the entire collection of data into two sets in order to conduct the prediction.

3.3.1. Train Set

For training the models. Considering 80% of the data to be in the training set for the machine learning model but 75% of the data is to be considered in the training set for the deep learning model.

3.3.2. Test Set

For testing the models. Considering 20% of the data to be a test set for the machine learning model, but, 25% of the data is to be considered in a training set for the deep learning model.

4. Results Analysis

After the implementation of the LSTM model, the training accuracy is 0.9623, and the training accuracy loss is 0.0487. Again, the value accuracy is 0.8602, and the value accuracy loss is 0.1549. From here see the actual score and loss of the implemented model. Here, used 10 epochs only; if used more than 10 epochs, then get less value loss and more training accuracy. Then, the LSTM model class is imported, which in turn is associated with the scikit-learn package. To fit the LSTM model, also used a similar TF-IDF-vectorizer to split the tweets into n-gram words. It also found that LSTM predicts 0.83-0.89 precision, 0.90-0.82 recall and 0.87 -0.85 f1 score, and support 2498-2502 accurately. A bidirectional LSTM may be required in that situation instead of a straightforward unidirectional one because it may not work as well. It would be interesting to learn how well the pre-trained model performs in additional downstream tasks, such as Spam Detection.

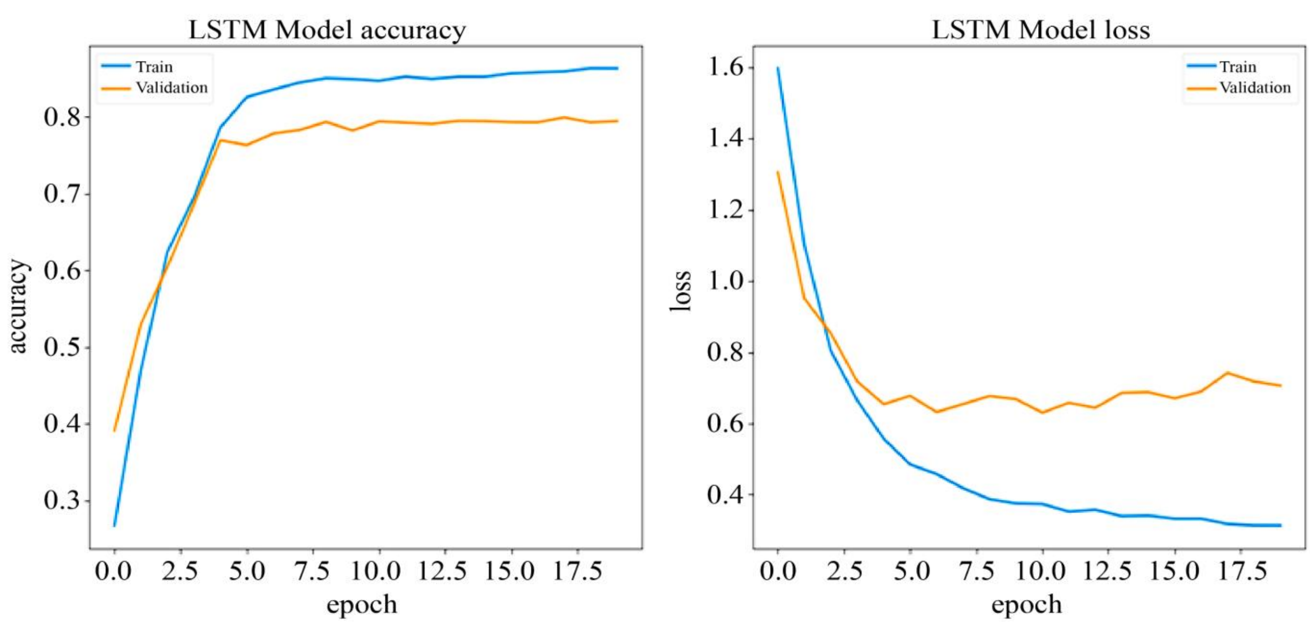

Fig. 7 LSTM Model training accuracy and loss plot

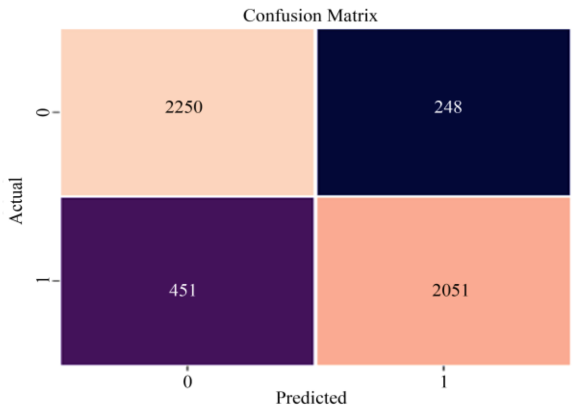

Fig. 8 Confusion matrix of the LSTM model

The findings appear to be reliable; however, they did not include this strategy in this model because it ran too slowly and did not have enough time to adjust the hyperparameters. Even though to obtain excellent performance on this dataset, we will continue to perform well on tasks that categorize news into more than two categories, like the Fake News Challenge. A data analysis and visualization technique was used to provide color to the bar graph to reflect height and width and how it was utilized to handle the dataset. It aids in identifying patterns and provides a sense of depth when seeing the concentration of values between two dimensions of a matrix. Also used, a confusion matrix was used to observe the data in this case. Hence, for this dataset, a generalized perspective of the numerical values is obtained. The confusion matrix shows different qualities in the situation.

4.1. Compare Algorithm Accuracy, Precision, Recall, and F1 Score

Here, the accuracy of the machine learning algorithm can be represented. The accuracy of the extra tree classifier is 86%, precision 88%, recall 85%, f1 score 86%, the accuracy of the K nearest Neighbors Classifier is 72%, precision 75%, recall 68%, f1 score 71%, the accuracy of the Random Forest Classifier is 85%, precision 87%, recall 83%, f1 score 85%, the accuracy of Multinomial Naïve Bayes is 68%, precision 68%, recall 69%, f1 score 68%, the accuracy of Decision Tree Classifier is 84%, precision 88%, recall 80%, f1 score 84%, the accuracy of XG Boost Classifier is 65%, precision 62%, recall 83%, f1 score 71%. Here extra tree gives the best accuracy because it gives accuracy depending on the maximum predicted data; if the maximum predicted data is real, then the result is real. If the maximum predicted data is fake, then the result is fake, then depending on the majority the result is declared. For this reason, the probability of predicting wrong is very low. For this, the extra tree gave the best accuracy. Precision also depends on the confusion matrix. The F1 score depends on the confusion matrix; if the confusion matrix provides a better result, then the F1 score also gives a better result. In the extra tree, the classifier provides a better confusion matrix and also gives the best F1 score.

Table 1. Compare algorithm accuracy

Algorithm	Accuracy
Extra Trees Classifier	86%
K nearest Neighbors Classifier	72%
Random Forest Classifier	85%
Multinomial Naïve Bayes	68%
Decision Tree Classifier	84%
XG Boost Classifier	65%

Table 2. Compare algorithm precision

Algorithm	Precision
Extra Trees Classifier	88%
K nearest Neighbors Classifier	75%
Random Forest Classifier	87%
Multinomial Naïve Bayes	68%
Decision Tree Classifier	88%
XG Boost Classifier	62%

Table 3. Compare algorithm recall

Algorithm	Recall
Extra Trees Classifier	85%
K nearest Neighbors Classifier	68%
Random Forest Classifier	83%
Multinomial Naïve Bayes	69%
Decision Tree Classifier	80%
XG Boost Classifier	83%

Table 4. Compare algorithm F1 score

Algorithm	F1 Score
Extra Trees Classifier	86%
K nearest Neighbors Classifier	71%
Random Forest Classifier	85%
Multinomial Naïve Bayes	68%
Decision Tree Classifier	84%
XG Boost Classifier	71%

4.1.1. Confusion Matrix

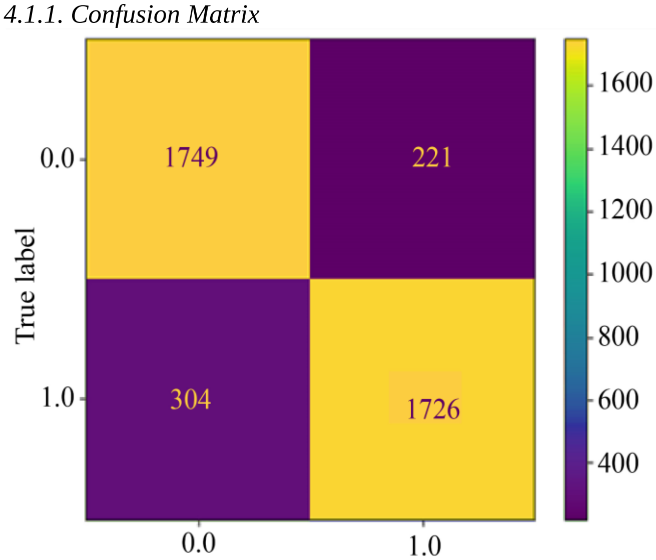

Fig. 9 Confusion matrix of extra trees classifier

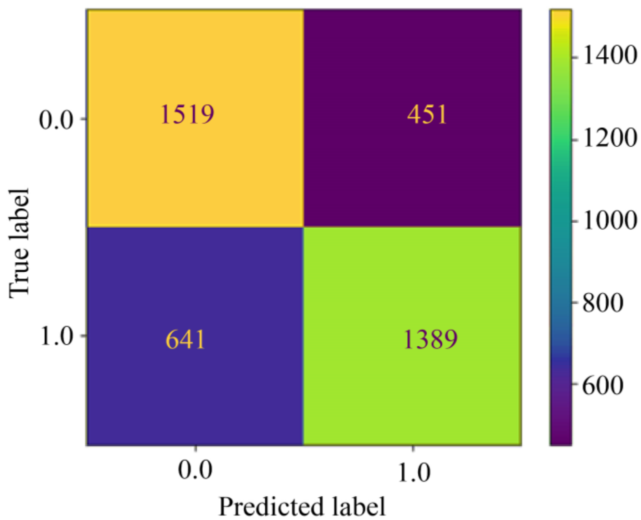

Fig. 10 Confusion matrix of K nearest neighbors classifier

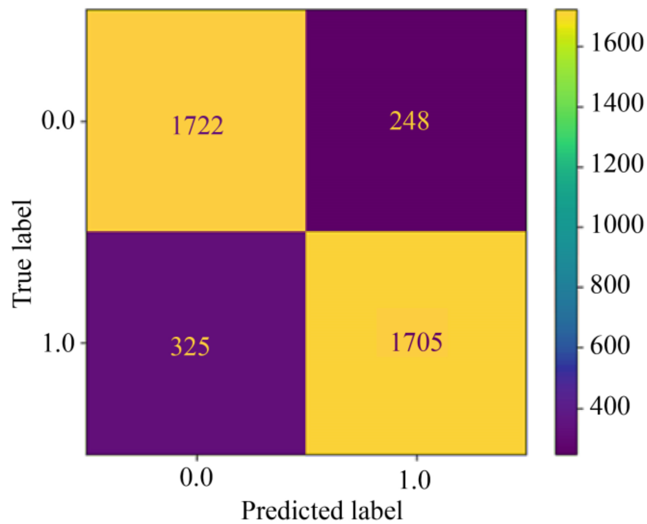

Fig. 13 Confusion matrix of random forest classifier

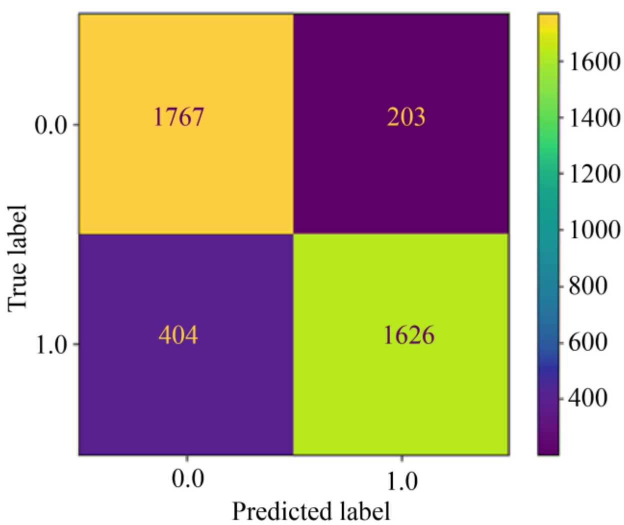

Fig. 11 Confusion matrix of decision tree classifier model

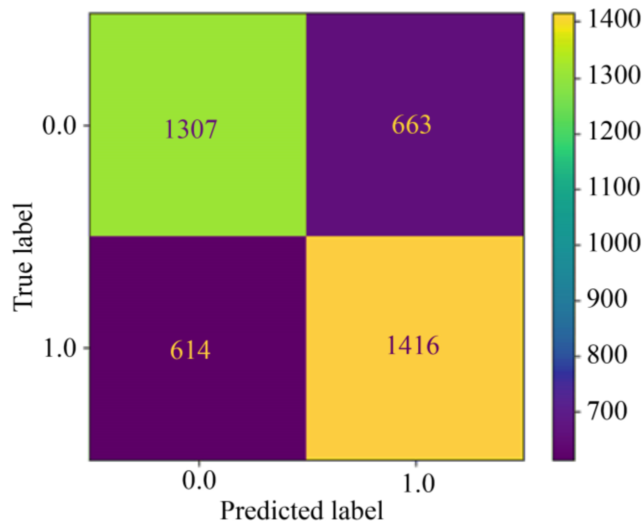

Fig. 14 Confusion matrix of multinomial Naïve Bayes

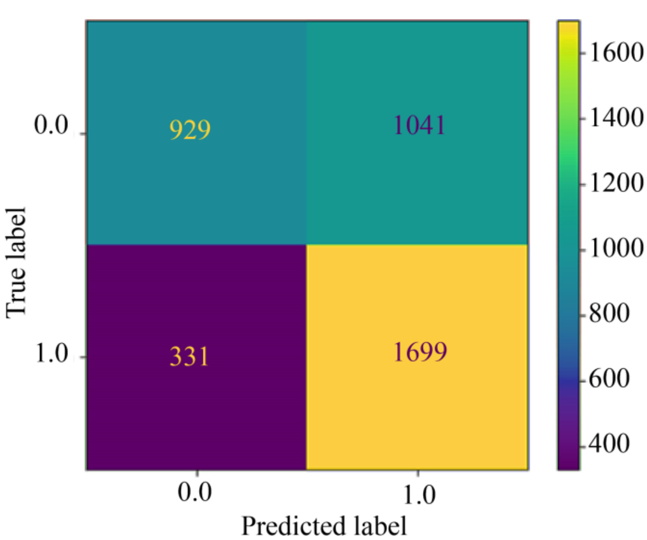

Fig. 12 Confusion matrix of XG boost classifier model

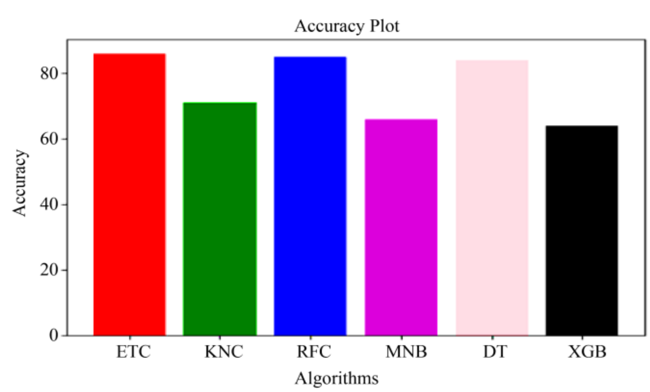

Fig. 15 Comparison among machine learning algorithm accuracy

4.2. Diagram for the value of Accuracy, Precision, Recall, and f1-Score

Here, the score of accuracy, precision, recall, and f1 score of all machine learning algorithms.

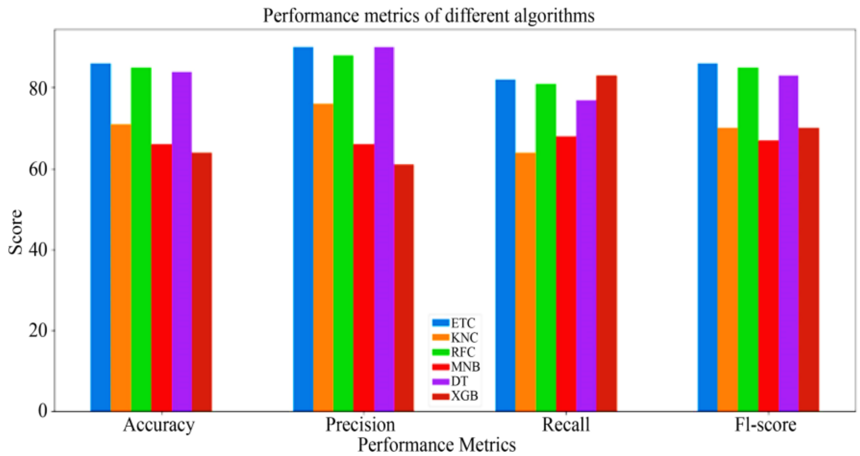

Fig. 16 Accuracy precision, recall, and F1-score of all algorithm

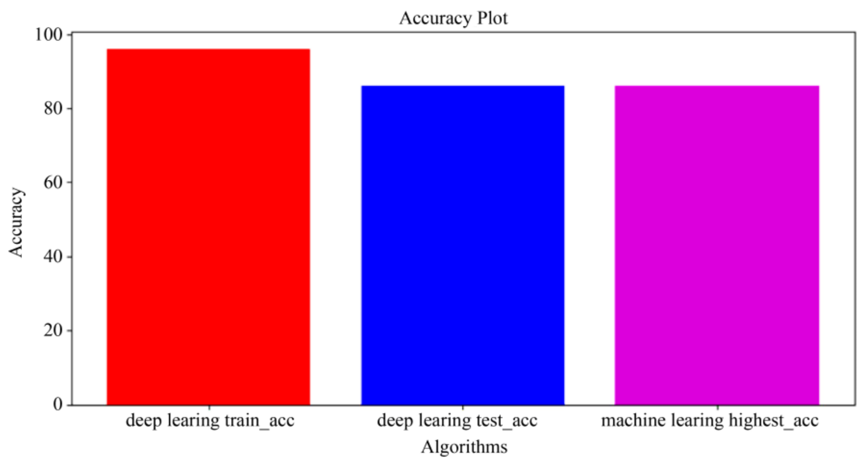

Fig. 17 Comparison between deep learning and machine learning models accuracy

The Extra Trees Classifier gave the highest accuracy of 86%, the Decision Tree Classifier and the Extra Trees Classifier gave the highest precision of 88%; the Extra Trees Classifier also gave the highest recall score of 85%, again Extra Trees Classifier gave the highest f1 score of 86%. The XG Boost Classifier gave the worst accuracy at 65%, and the XG Boost Classifier gave the worst precision at 62%; Multinomial Naïve Bayes gave the worst recall score at 69%, again Multinomial Naïve Bayes also gave the highest f1 score at 68%.

4.3. Comparison between Deep Learning and Machine Learning Models Accuracy

The figure shows the comparison of deep learning and machine learning training and testing. Identifying false news in news headline data is roughly 86.14%.

Moreover, seeing that deep learning algorithm outcomes are better than machine learning outcomes. Furthermore, LSTM has a 96.14% training accuracy rate and an 86% testing accuracy rate for identifying false news in news headline data. So, by plotting a figure showing the comparison between deep and machine learning algorithm’s accuracy.

5. Conclusion

The manual classification of news requires a deep knowledge of the field as well as the capacity to identify irregularities in the text. This study covered the issue of classifying false news stories using ensemble approaches and deep learning models. More accurate models have been discovered to exist than others. People will interact more frequently and express their feelings about many issues as time goes on. Bangladesh's condition in this regard is comparable to that of other developed nations. So, in order to prevent such a worldwide crisis, a system or model that can capture the emotional reactions of the concerned audience as a whole must be created. In this work create models that can detect fake news.

Here, the top model LSTM has an accuracy of 96.14%, and the extra tree classifier has an accuracy of 86% when identifying fake news headlines from textual input. Moreover, the model accurately detects fake news headlines from uncomplicated news headlines. These models are capable of identifying fake news. With more data, the models will perform better. So, there is a scope to test these models with more data.

References

[1] David M.J. Lazer et al., "The Science of Fake News," *Science*, vol. 359, no. 6380, pp. 1094-1096, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] Tasnuba Sraboni et al., "*FakeDetect: Bangla Fake News Detection Model Based On Different Machine Learning Classifiers*," Brac University, pp. 1-45, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Shafayat Bin Shabbir Mugdha, Sayeda Muntaha Ferdous, and Ahmed Fahmin, "Evaluating Machine Learning Algorithms For Bengali Fake News Detection," *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Soroush Vosoughi, Deb Roy, and Sinan Aral, "The Spread of True and False News Online," *Science*, vol. 359, no. 6380, pp. 1146-1151, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Benjamin Riedel et al., "A Simple but Tough-to-Beat Baseline for the Fake News Challenge Stance Detection Task," *arXiv*, pp. 1-6, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] Hadeer Ahmed, Issa Traore, and Sherif Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, Vancouver, BC, Canada, pp. 127-138, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mykhailo Granik, and Volodymyr Mesyura, "Fake News Detection Using Naive Bayes Classifier," *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Kyiv, Ukraine, pp. 900-903, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Anjali Jain et al., "A Smart System for Fake News Detection Using Machine Learning," *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Ghaziabad, India, pp. 1-4, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Marco L. Della Vedova et al., "Automatic Online Fake News Detection Combining Content and Social Signals," *2018 22nd Conference of Open Innovations Association (FRUCT)*, Jyvaskyla, Finland, pp. 272-279, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Rakib ul Haque et al., "A Complete Bengali Stop Word Detection Mechanism," *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV)*, Spokane, WA, USA, pp. 103-107, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Aswini Thota et al., "Fake News Detection: A Deep Learning Approach," *SMU Data Science Review*, vol. 1, no. 3, pp. 1-20, 2018. [Google Scholar] [Publisher Link]

[12] Verónica Pérez-Rosas et al., "Automatic Detection of Fake News," *arXiv*, pp. 1-10, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[13] Md Zobaer Hossain et al., "Banfakenews: A Dataset for Detecting Fake News in Bangla," *arXiv*, pp. 1-10, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Tasnuba Sraboni et al., "*FakeDetect: Bangla Fake News Detection Model Based on Different Machine Learning Classifiers*," Thesis, Brac University, pp. 1-45, 2021. [Google Scholar] [Publisher Link]

[15] David H. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[16] Arnab Sen Sharma, Maruf Ahmed Mridul, and Md Saiful Islam, "Automatic Detection of Satire in Bangla Documents: A CNN Approach Based on Hybrid Feature Extraction Model," *arXiv*, pp. 1-5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kai Shu, Deepak Mahudeswaran, and Huan Liu, "Fake News Tracker: A Tool for Fake News Collection, Detection, and Visualization," *Computational and Mathematical Organization Theory*, vol. 25, pp. 60-71, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Mohamed Torky, Emad Nabil, and Wael Said, "Proof of Credibility: A Blockchain Approach for Detecting and Blocking Fake News in Social Networks," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, pp. 321-327, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] Kai Shu, Suhang Wang, and Huan Liu, "Exploiting Tri-Relationship for Fake News Detection," *arXiv*, pp. 1-10, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[20] Arjun Roy et al., "A Deep Ensemble Framework for Fake News Detection and Multi-Class Classification of Short Political Statements," *Proceedings of the 16th International Conference on Natural Language Processing*, International Institute of Information Technology, Hyderabad, India, pp. 9-17, 2019. [Google Scholar] [Publisher Link]