Towards the Detection of Fake News using different Machine Learning and Natural Language Processing Algorithms

Md Naimul Islam Suvon
Electrical & Computer Engineering
North South University
Dhaka, Bangladesh
naimul.suvon@northsouth.edu

Md. Emanul Haque Rafi
Electrical & Computer Engineering
North South University
Dhaka, Bangladesh
emanul.haque@northsouth.edu

Evan Alam
Electrical & Computer Engineering
North South University
Dhaka, Bangladesh
evan.alam@northsouth.edu

Md. Habibullah
Electrical & Computer Engineering
North South University
Dhaka, Bangladesh
md.habibullah@northsouth.edu

***Abstract* — The amount of information shared on the internet, primarily via web-based networking media, grows day by day. Because of the simple availability and exponential expansion of data through social media networks, distinguishing between fake and real information. Most smartphone users tend to read news on social media rather than on the internet. The information published on news websites often needs to authenticate. The simple spread of reports by way of sharing has included the exponential development of its misrepresentation. So, fake news has been a major issue ever since the web developed and expanded it to the general mass. This paper demonstrates several models and techniques for detecting false news by using different machine learning and natural language processing (NLP) models such as Logistic Regression, Decision Tree, Naïve Bayes, Support Vector Machine (SVM), Long Short-Term Memory (LSTM), Bidirectional Encoder Representation from Transformers (BERT). We tried to combine the news, then find out if the information was authentic or fake. Various feature engineering methods such as Regex, Tokenization, stop words, Lemmatization, Term Frequency-Inverse Document Frequency (TF-IDF) generate feature vectors in this paper. Every Machine Learning and NLP model was evaluated with test data. For the machine learning model Logistic Regression, Decision Tree, Naïve Bayes, and SVM, we got 73.75%, 89.66%, 74.19%, and 76.65%, respectively. But the highest accuracy we git is for the NLP method, which is 95% for LSTM and 98% for the BERT language model.**

Keywords — fake news, machine learning, deep learning, logistic regression, naïve bayes, decision tree, support vector machine (SVM), tokenization, lemmatization.

I. INTRODUCTION AND RELATED WORKS

Information is significant for human dynamics and affects life practices. In earlier days, the daily news or information used to occur through electronic media such as newspapers, television, and radio. Such data is more honest as it is either self-screened or constrained by specialists. These days, individuals are presented with an extreme amount of data through various sources, particularly with the prominence of the internet and web-based media stages. The ease of internet access has caused the hazardous development of a wide range of falsehoods like discussion, double-dealing, fabrications, fake news, spam assessment, which diffuses quickly and widely in the human culture. The misinformation has become a global problem in society and public trust. Thus, this has created research interests among the researchers in detecting fake news for a better future endeavor.

In [1], the authors worked on fake news detection with the help of a deep learning technique, CNN-LSTM. The authors had obtained different results when they pre-processed data and when they did not. In this paper, they have used the FNC dataset. They matched the claim with the news article body whether the claim matches with the article body or not. They have developed four data models. First, they use data without pre-processing, second with pre-processing, and in the third and fourth models, they created the models by dimensionality reduction techniques, including PCA. They trained on forty-nine thousand and nine hundred seventy-two samples and tested on twenty-five thousand and four hundred thirteen headlines and articles by CNN-LSTM. On their model with no knowledge of cleanup or pre-processing, the accuracy was seventy-eight. When pre-processing, the accuracy goes up to 93.0%. Next, the application of chi-square raises the accuracy by 95%. Lastly, they conclude that using PCA with CNN and LSTM design resulted in the highest accuracy of 96%, which can be considered a very high accuracy level.

In [2], T. Jiang et al. used baseline fake news identification techniques to locate the baseline methods' flaws and provide a viable alternative. First, the authors performed five completely different machine learning models and three deep learning models to assume and compare the excellence between these models. Next, they implemented two datasets of various sizes to test the corresponding models' power on datasets of multiple dimensions. Finally, they take advantage of a remedied adaptation of McNemar's check to decide if they are square measure contrasts between these two models' presentation, then determine the simplest model for detecting the fake news. The authors obtained 99.94% exactness on the ISOT dataset and 96.05% precision on the KDnugget dataset. They compared their outcomes with other existing work and concluded that the proposed stacking model is vastly better.

In a recent work [3], the authors designed a system for detecting fake news using machine learning. This paper reviewed numerous machine learning tactics within the detection of fake and fancied news. This process and improvisation worked with the manner of implementing deep understanding and are additionally reviewed. Each tweet/post has been categorized as a binary categorization result by the writers. The researchers categorized every tweet/post as a binary classification problem, with the origin of the post/tweet being categorized depending upon this. The researchers collected data manually from their own research sets while Twitter API and the DMOZ directory were used. The authors ran a test of their proposed system on Twitter. The results show that fifteen percent of faux tweets and forty-five percent of the actual tweets were adequately classified, and the remainder of the posts were not decided. In their paper, the authors recommended fake news detection data tagged by benchmark information set 'FAKE' along with manifestly improved potency in a police investigation for fraud news that spreads. The authors have introduced the requirement for hoax detection. They used the cc approach by combining news content and social content techniques. The researchers uncovered the motivations, concepts, methodologies, and, most significantly, the algorithms used to distinguish false and fictional news writers and topics from virtual communities and other related impact and efficiency metrics. The research also suggested that the analysis be hampered by the unknown features of false news and the various links between news items, writers, and topics, all of which continuously produce illusions among the general public.

In [4], A. Jain et al. design a system that detects the news as false or true. Social media like Facebook and other platforms like news platforms spread violent words then catch fake news. In this paper, the author demonstrates natural language processing, a Machine learning-based fake news detection system. The author uses Naive Bayes classification and the SVM algorithm, Logistic Regression, the model, and the data classification in this system. They implement their model and classify the authentic news. After that, they can also add the recommendation system where the information is fake. A similar accurate report suggests based on user keywords. In this paper, the accuracy result is 93.5%.

This paper has designed a fake news detection system using a different types of machine learning techniques. The datasets of the proposed artificial news detection system were collected from an open-source dataset website called Kaggle. This dataset was collected from authentic world sources. Initially, we pre-processed this dataset using Regex, Tokenization, Stop Words, Lemmatization, and then applied NLP techniques, count vectorizer, TF-IDF vectorizer. For the classification of our proposed system, we have used the logistic regression machine learning algorithm.

In our paper, we have used (i) Machine Learning algorithms and so dynamic (ii) Natural Language Processing Algorithm and compared the result of them. The *nobility* of our work is that we have used a language model BERT for detecting fake news.

The other part of the paper is constructed as follows. In Section 2, the proposed system has been discussed with appropriate equations. The actual results of the research have been shown in Section 3. Lastly, Section 4 concludes the paper with some directions for the future improvement of this work.

II. Proposed System Methodology

In this Section, we have discussed the methodology of our work. We have discussed all the machine learning and NLP methods that we have used in our data set.

A. Dataset

We used the dataset [5] from Kaggle in our system. It is an open-source dataset. Lots of fake news and agenda were going on at that time, so the whole data was curated to do something with the help of data science technologies. It was posted on the data science community as a challenge to use those data to implement fake news detection. The dataset has four columns – id, title, author, text. The id column represents a unique id for a news article; the title holds the title of a news article; the author column contains the information about the author of the news article, and under the text column, we can see the text of the article, and it may be incomplete. The training dataset has the label column, which marks the article as potentially unreliable or reliable. The dataset has 20,822 unique values in the text column.

B. Data Pre-Processing

We need to transform the text data using preprocessing techniques like NLP, tokenization, and lemmatization before feeding the data set through the ML and DL models. Data pre-processing helps to remove the noises and inconsistency of data. That increases the performance and efficiency of our model. In this work, we have used regex, tokenization, stopwords, lemmatization, NLP technique, and TF-IDF for data pre-processing.

1) Regex: We use regex to remove punctuations from the text data. Often in the sentences, there may have extra punctuations like exclamatory signs. We use regex to remove those extra punctuations to make the dataset noise-free. Regex is based on context-free grammar.

2) Tokenization: We use tokenization to break the sentences into words.

3) StopWords: We use the English stopwords library in our pre-processing technique because our model data is English. We need to use the stopwords preprocessing technique to remove noises, make the model faster and efficient, and save memory space.

4) Lemmatization: Lemmatization is used to transform the words into root words. We can resolve data ambiguity with lemmatization.

5) NLP Techniques: We use NLP techniques to convert the texts into meaningful numbers to feed these numbers into our proposed machine learning algorithm.

Bag of words: The bag of words technique converts texts into machine-understandable numbers.

$$TF - IDF = TF_{td} \, . IDF_t \tag{1}$$

TF stands for Term Frequency. TF is a measurement of how frequently a term appears in a document. Here, t is a term, and d is the documents.

$$TF = \frac{q_{td}}{Number\ of\ terms\ in\ the\ document} \tag{2}$$

where q is the number of times the term, t appears in the document, d.

IDF stands for Inverse Document Frequency. IDF Indicates how important a particular term is.

$$IDF = \frac{\log(1+n)}{(1+df)_{dt}} + 1 \quad (3)$$

Where n means the number of documents and the denominator indicates the document frequency of the term, t.

C. Machine Learning Algorithms

To classify fake news, we have used different Machine learning algorithms: Logistic Regression, Naïve Bayes Decision Tree, and Support Vector Machine.

1) Logistic Regression

The basis of our proposed system consists of the binary classification problem. Logistic Regression is a statistical ML classification model. Logistic Regression is manipulated to model the probability of a certain event existing, such as true/false, reliable/unreliable, win/lose. So, this logistic model is one of the most appropriate models for our fake news detection system. The condition for predicting logistic model is,

$$0 =< h\theta(x) =< 1 \quad (4)$$

The logistic regression sigmoid function is:

$$h\theta(x) = g(\theta^T X) \quad (5)$$

here,

$$g(z) = \frac{1}{(1+x^{-z})} \quad (6)$$

and the cost function of logistic regression is:

$$J(\theta) = 1/m \sum_{i=1}^{m} cost(h\theta(x^i, y^i)) \quad (7)$$

2) Naïve Bayes

The Naive Bayes method is at the basis of Bayesian classifiers. It is a strategy for looking at possible outcomes that allows you to flip the state around straightforwardly. A conditional probability is a probability that incident X will happen provided information Y. The typical notation for this is $P(X|Y)$. We can use the Naive Bayes rule to compute this probability when we only have the probability of the opposite result and the two components separately.

$$P(X|Y) = \frac{P(X)\, P(Y|X)}{P(Y|X)\, P(Y)} \quad (8)$$

This restatement can be extremely useful when we're trying to predict the likelihood of something based on examples of it happening.

In this example, we're attempting to determine if an article is false or genuine based on its contents. We may rephrase it in terms of the likelihood of that document being Real or Fake if it has been predetermined to be Real or Fake. This is useful since we already have instances of real and fake articles in our data collection.

We make a big assumption about how we may compute the likelihood of the article happening; it is equal to the product of the probabilities of each word inside its occurrence, making this procedure a "naive" Bayesian one. This suggests that there is no connection between the two words. It's known as the assumption of independence.

We can estimate the likelihood of a term occurring by looking at a set of Real and Fake article samples and noting how many times it appears in each class. The necessity for pre-classified samples to train on is what distinguishes this as supervised learning.

3) Decision Tree

The J48 method is one of the most widely used classification algorithms. It is based on the C4.5 algorithm, which requires all data to be studied to be quantitative and categorical. As a result, continuous data will not be investigated [6,7,8]. J48 employs two distinct pruning techniques. J48 employs two distinct pruning techniques. The first approach is known as subtree replacement, and it refers to replacing nodes in a decision tree's leaves to reduce the number of tests in the convinced route. In most cases, subtree raising has a minor influence on decision tree models. In most cases, there is no accurate method to forecast an option's usefulness. However, it may be advisable to turn it off if the induction operation takes longer than expected due to the subtree's raising being computationally difficult.

Predefined classes are used as input.
Output: decision tree construction.
The number of features is 17000.
Begin
Step 1: Make the tree's root node.
Step 2:
Return leaf node 'positive' if all instances are positive.
Return leaf node 'negative' if all instances are negative.
Step 3: Determine the current state's entropy H. (S)
Step 4: Calculate the entropy for each characteristic.
Step 5: Choose the attribute with the highest IG value (S, x)
Step 6: From the list of attributes, remove the attribute with the greatest IG.
Step 7: Continue until all characteristics have been exhausted or the decision tree has all leaf nodes.
End

4) Support Vector Machine (SVM)

An SVM, which is also known as a support vector network, is a supervised learning method. SVMs are trained using particular data that has previously been divided into two groups. As a result, once the model has been trained, it is created. Moreover, the goal of the Support Vector Machine technique is to decide which group any new information belongs to and to increase the class label [9]. The final goal of the SVM is to locate a subspace that divides the data into two parts groups. Because RBF is suitable for large systems like a collection of media articles, it was chosen as the kernel for this system. On two samples x and x', the Radial Basis Function is:

$$K(x, x') = e^{-\frac{||x-x'||^2}{2\sigma^2}} \quad (8)$$

Where $||x\text{-}x'||^2$ is a free parameter that denotes the squared Euclidean distance.

D. Natural Language Procession Algorithms

To classify fake news, we have used to NLP algorithms and both of them are dynamic. They are LSTM and language model BERT.

1) Long Short-Term Memory (LSTM)

A recurrent neural network is a long short-term memory (LSTM). Separate hidden motors are used in LSTMs, and their natural nature is to recall inputs for a long time. A memory cell, also known as a gated leaky neuron or an accumulator, has a relationship in the following stages with its weight of 1. It mimics its genuine position and inserts an external signal, but this signal is multiplied coded by that other unit that determines when to wipe information from memory. Fig.1 shows a generic LSTM based neural network architecture.

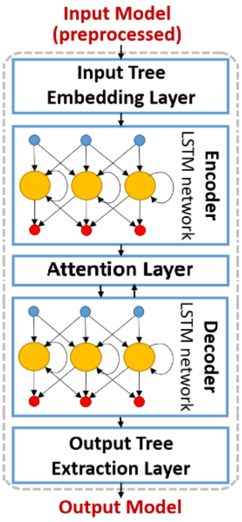

Fig.1. Generic LSTM Architecture

For our classification, we used an LSTM model with an input layer that takes the input titles and article body and an embedding layer that turns every word into a 300-pixel vector. As there are 256 features, this layer will produce a 256*300 matrix. The weights we obtain from matrix multiplication will be in the output matrix, which will generate a vector for every word. These vectors are input through an LSTM, which is subsequently transferred to a fully linked dense layer, resulting in a single final output. Fig.2 shows the model layers and parameters, which were trained on batches of size 256.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 256)]	0
embedding (Embedding)	(None, 256, 300)	60974100
spatial_dropout1d (SpatialDr	(None, 256, 300)	0
bidirectional (Bidirectional	(None, 256)	439296
dense (Dense)	(None, 64)	16448
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 61,429,909
Trainable params: 61,429,909
Non-trainable params: 0

Fig.2. Layers and Parameters of our LSTM model

2) Bidirectional Encoder Representation from Transformers (BERT)

BERT is designed to pre-trained bidirectional representations from an unlabeled text by conditioning both right and left backgrounds in all levels. As a result, the BERT model might suffice with only one additional output layer to produce advanced models for various tasks, including query answers.

For tokenizing sentences into words, converting token strings to ids and back, and encoding/decoding, BertTokenizer from the pretrained ‘bert-base-uncased' model, was utilized in this study. The max sentence length is 60 characters, and we utilized the encode plus technique to encode each one. This technique will tokenize the phrase, prep the [CLS] (classification) token at the beginning, and append the [SEP] (which tells BERT where to start the next phrase). In most cases, it is inserted after each phrase) token. Tokens should be mapped to their ids; the phrase should be padded to the attention masks, and the maximum length for [PAD] (padding) tokens should be created. Bert model uses the argument of attention mask. It specifies which tokens should be dealt with and which should be ignored. It is up to it to notify the model whether tokens include valid data. Fig.3 shows the architecture of the BERT model.

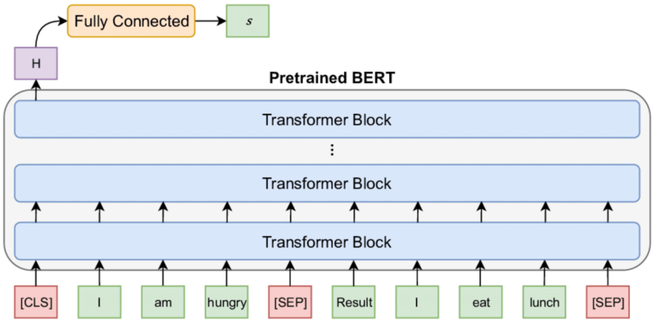

Fig.3. BERT model Architecture

Fig.4 shows the layers and parameters of our BERT model input ids and attention masks used as an input layer and after that the output of the input layers goes to the transformer BERT model which is subsequently transferred to a fully linked dense layer, resulting in a single final output.

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 60)]	0	
attention_masks (InputLayer)	[(None, 60)]	0	
tf_bert_model (TFBertModel)	TFBaseModelOutputWit	109482240	input_ids[0][0] attention_masks[0][0]
dense (Dense)	(None, 32)	24608	tf_bert_model[0][1]
dropout_37 (Dropout)	(None, 32)	0	dense[0][0]
dense_1 (Dense)	(None, 1)	33	dropout_37[0][0]

Total params: 109,506,881
Trainable params: 109,506,881
Non-trainable params: 0

Fig.4. Layers and Parameters of BERT model

III. Result and Analysis

We can now assess our models once we've completed all of the necessary processing on our dataset and training. The model may be evaluated in a variety of ways. The accuracy matrices are the most frequent. Confusion matrix, Recall, Precision, F1-score, ROC curve, and other accuracy matrices are among them. We attempted to evaluate our models with these methods.

A. *Performance of Logistic Regression Model*

In Fig. 5, the confusion matrix for the logistic regression model of the proposed system has been shown. The Real news class has 862 right predictions and 170 wrong predictions from 1032 test samples of real news. So, the accuracy for real news prediction is 83.52%, and for the Fake news class, it has 487 correct predictions but a huge wrong prediction of 310 from 797 test samples of fake news. So, the accuracy for fake news is 61%. The overall accuracy is 74%.

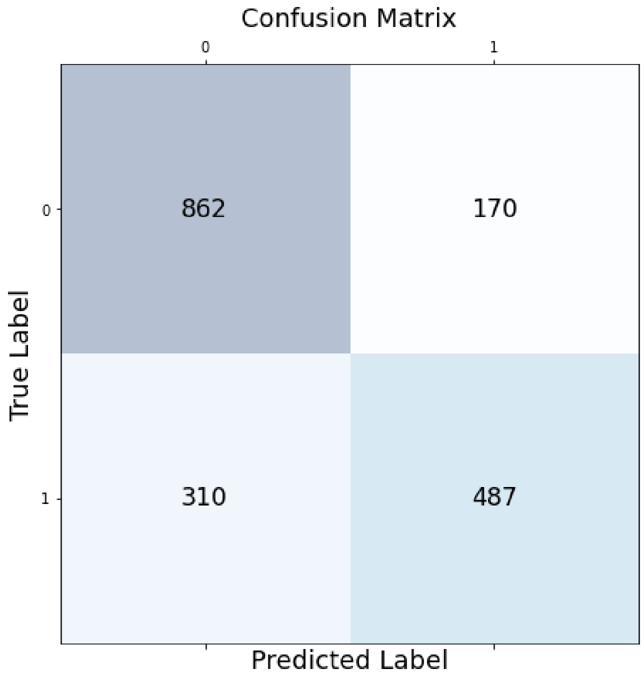

Fig. 5. Confusion Matrix for Logistic Regression

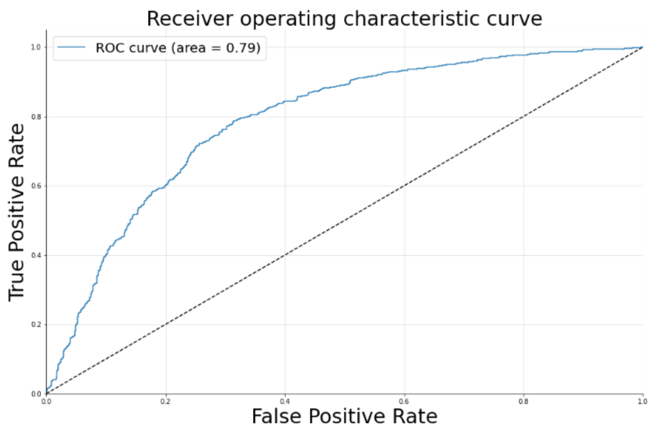

Fig.6. ROC Curve for Logistic Regression

In Fig.7, the rest of the performance metrics for the logistic regression model are demonstrated. The proposed logistic regression model's precision, recall, and F1-score are 74%, 72%, and 73%, respectively. According to Fig. 6, the ROC value of the proposed logistic regression algorithm is 0.79.

	precision	recall	f1-score	support
0	0.74	0.84	0.78	1032
1	0.74	0.61	0.67	797
accuracy			0.74	1829
macro avg	0.74	0.72	0.73	1829
weighted avg	0.74	0.74	0.73	1829

Fig.7. Logistic Regression Model Accuracy Metrics

B. *Performance of Naïve Byes Model*

In Fig. 8, the confusion matrix for the naïve Bayes model of the proposed system has been shown. The Real news class has 830 right predictions and 202 wrong predictions from 1032 test samples of real news. So, the accuracy for real news prediction is 80%, and for the Fake news class, it has 527 correct predictions but a huge wrong prediction of 270 from 797 test samples of fake news. So, the accuracy for fake news is 66%. The overall accuracy is 74%.

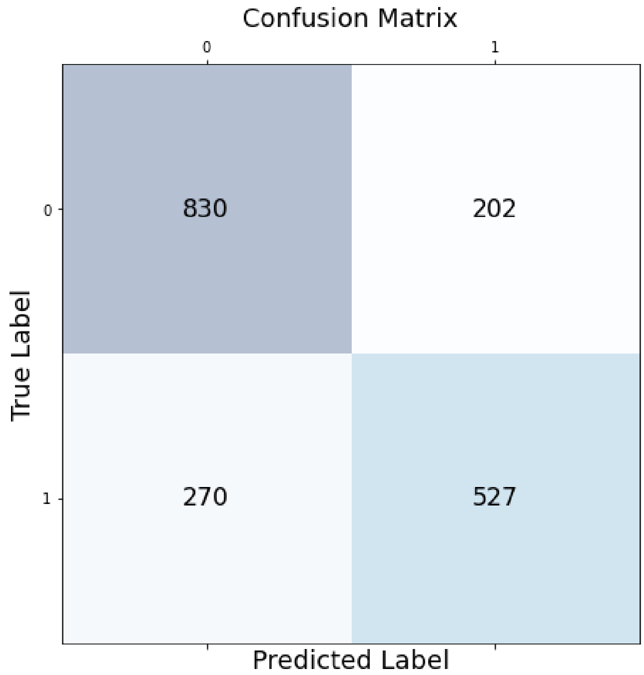

Fig.8. Confusion Matrix for Naïve Bayes

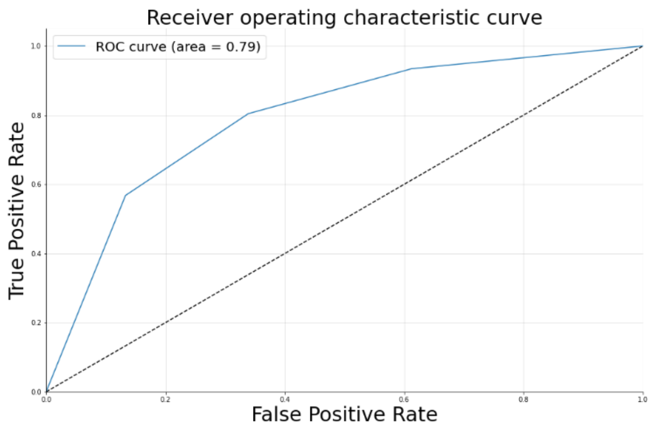

Fig.9. ROC Curve for Naïve Bayes

In Fig.10, the rest of the performance metrics for the naïve Bayes model are demonstrated. The precision, recall, and F1-score of the proposed naïve Bayes model are 74%, 73%, and 73%, respectively. According to Fig. 9, the ROC value of the proposed naïve Bayes algorithm is 0.79.

	precision	recall	f1-score	support
0	0.75	0.80	0.78	1032
1	0.72	0.66	0.69	797
accuracy			0.74	1829
macro avg	0.74	0.73	0.73	1829
weighted avg	0.74	0.74	0.74	1829

Fig.10. Naïve Bayes Model Accuracy Metrics

C. Performance of Decision Tree Model

In Fig. 11, the confusion matrix for the decision tree model of the proposed system has been shown. The Real news class has 940 right predictions and 92 wrong predictions from 1032 test samples of real news. So, the accuracy for real news prediction is 91%, and for the Fake news class, it has 700 correct predictions but a huge wrong prediction of 97 from 797 test samples of fake news. So, the accuracy for fake news is 88%. The overall accuracy is 90%.

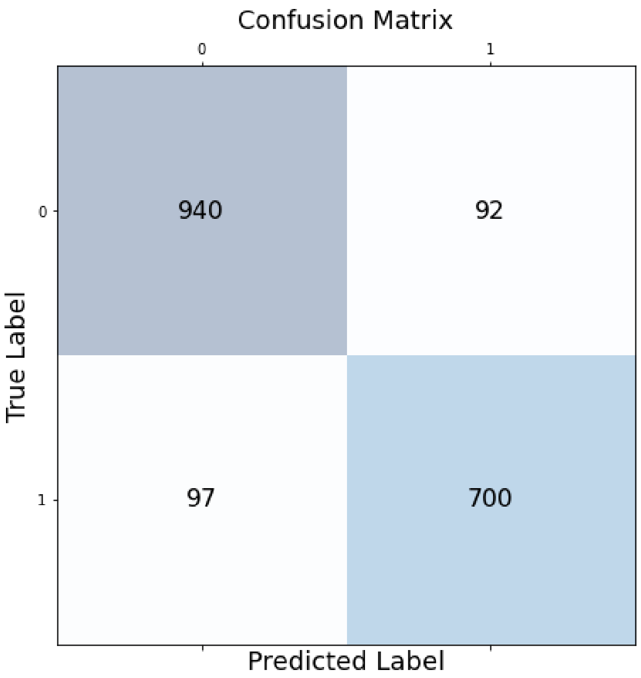

Fig.11. Confusion Matrix for Decision Tree

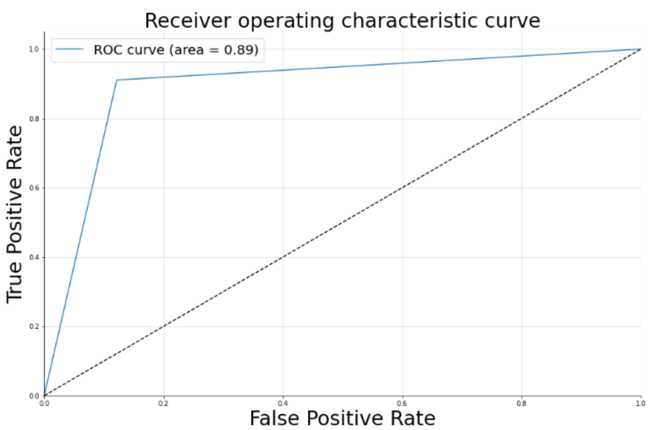

Fig.12. ROC Curve for Decision Tree

In Fig.13, the rest of the performance metrics for the decision tree model are demonstrated. The precision, recall, and F1-score of the proposed decision tree model are 90%, 89%, and 89% respectively. According to Fig. 12, the ROC value of the proposed decision tree algorithm is 0.89.

	precision	recall	f1-score	support
0	0.91	0.91	0.91	1032
1	0.88	0.88	0.88	797
accuracy			0.90	1829
macro avg	0.90	0.89	0.89	1829
weighted avg	0.90	0.90	0.90	1829

Fig.13. Decision Tree Model Accuracy Metrics

D. Performance of Support Vector Machine Model

In Fig. 14, the confusion matrix for the SVM model of the proposed system has been shown. The Real news class has 848 right predictions and 184 wrong predictions from 1032 test samples of real news. So, the accuracy for real news prediction is 82%, and for the Fake news class, it has 554 correct predictions but a huge wrong prediction of 243 from 797 test samples of fake news. So, the accuracy for fake news is 70%. The overall accuracy is 77%.

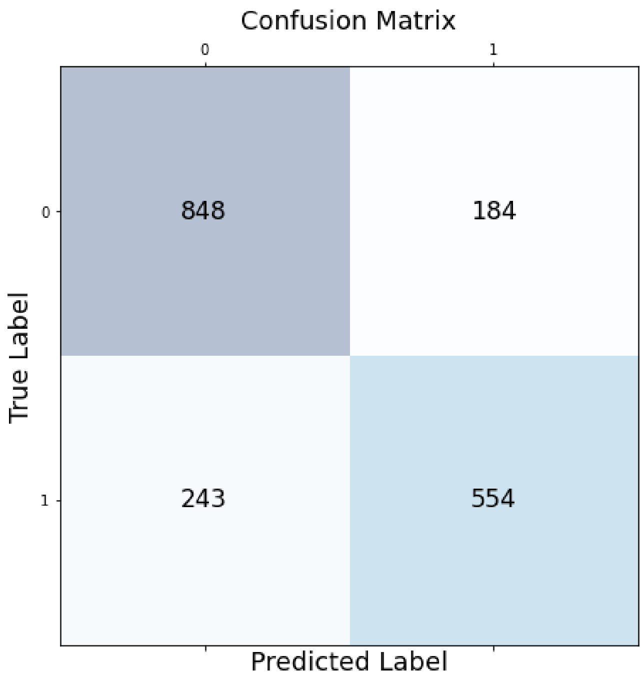

Fig.14. Confusion Matrix for SVM

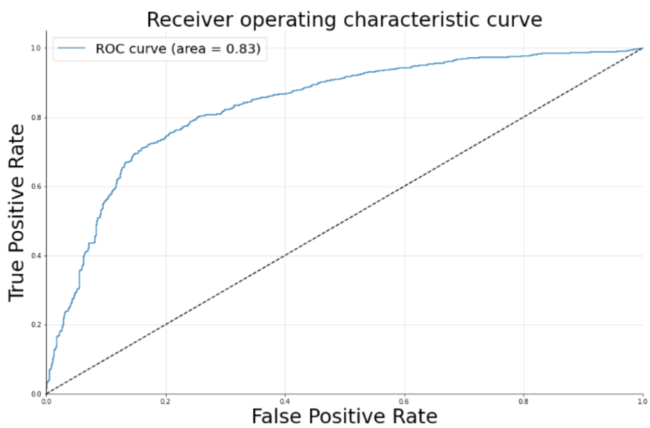

Fig.15. ROC Curve for SVM

In Fig.16, the rest of the performance metrics for the SVM model are demonstrated. The precision, recall, and F1-score of the proposed SVM model are 76%. According to Fig. 15, the ROC value of the proposed SVM algorithm is 0.83.

	precision	recall	f1-score	support
0	0.78	0.82	0.80	1032
1	0.75	0.70	0.72	797
accuracy			0.77	1829
macro avg	0.76	0.76	0.76	1829
weighted avg	0.77	0.77	0.77	1829

Fig.16. SVM Model Accuracy Metrics

E. Performance of LSTM Model

In Fig. 17, the confusion matrix for the LSTM model of the proposed system has been shown. The Real news class has 1920 right predictions and 157 wrong predictions from in total 2077 test samples of real news. So, the accuracy for real news prediction is 92%, and for the Fake news class, it has 1537 correct predictions but a huge wrong prediction of 43 from in total 1580 test samples of fake news. So, the accuracy for fake news is 97%. The overall accuracy is 95%. The total number of test samples for each class is different from the machine learning testing because of the better pre-processing for NLP methods which helps to decreased the chances of removing samples.

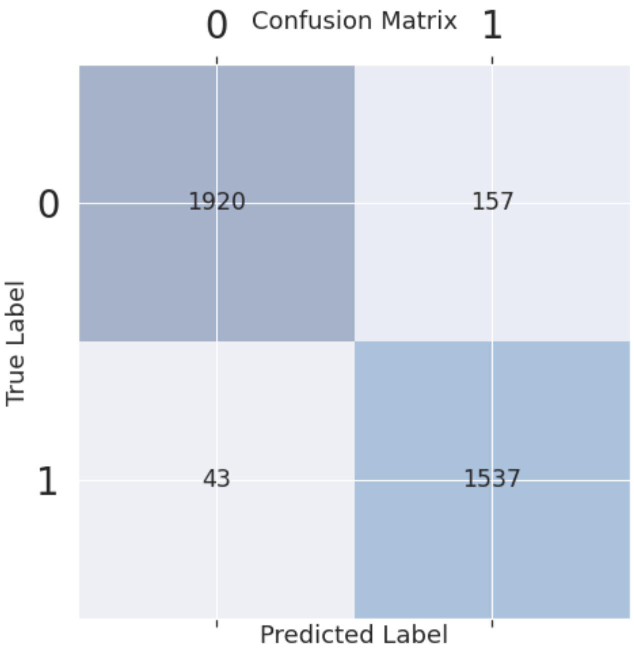

Fig.17. Confusion Matrix for LSTM

In Fig.18, the rest of the performance metrics for the LSTM model are demonstrated. The precision, recall, and F1-score of the proposed LSTM model are 94%, 95%, and 94%, respectively.

	precision	recall	f1-score	support
0	0.98	0.92	0.95	2077
1	0.91	0.97	0.94	1580
accuracy			0.95	3657
macro avg	0.94	0.95	0.94	3657
weighted avg	0.95	0.95	0.95	3657

Fig.18. LSTM Model Accuracy Metrics

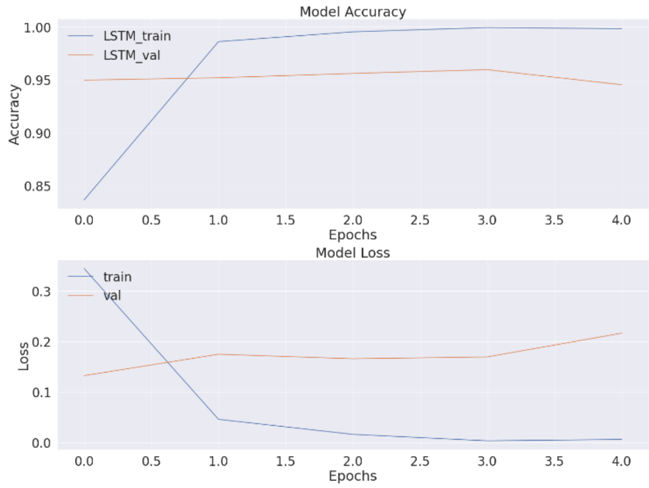

Fig.19. Accuracy and loss VS Epochs Graph of LSTM

Fig.19 shows the accuracy and loss graph of LSTM with respect to epoch. For our LSTM model at the start of the training, the model's validation starts from 95%, which didn't change that much, and after 4 epochs, it remained 95%.

F. Performance of BERT Model

Fig.20 shows the encoder and decoder result on an example sentence. The purpose of this result is to show how all of our sentences are encoding and decoding. Here the input for the encoding is "Hi nice meet you!". After encoding, we can see that all the words and symbols represent a value. For example, here, hi=7632. If we decode it, we will get the exact output that we have given to the encoder as an input. There are two new words after decoding. One is at the beginning of the sentence, which is CLS. It represents classification. Another one is SEP at the end of the sentence, which tells BERT where to start the next sentence.

```
encode = bert_tokenizer.encode("Hi nice meet you !")
decode = bert_tokenizer.decode(encode)

print("Encode: ", encode)
print("Decode: ", decode)

Encode:  [101, 7632, 3835, 3113, 2017, 999, 102]
Decode:  [CLS] hi nice meet you! [SEP]
```

Fig.20. Encoder and Decoder Example Result

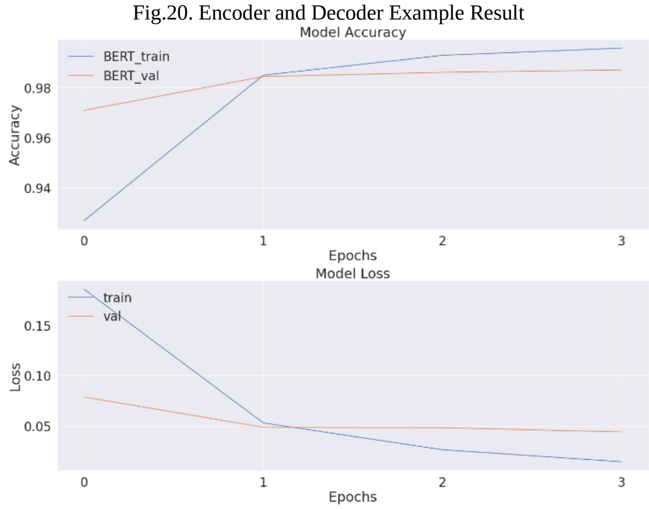

Fig.21. Accuracy and loss VS Epochs Graph of BERT

Fig.21 shows the accuracy and loss graph of BERT with respect to epoch. For our BERT model at the start of the training, the model's validation starts from 97%, which didn't change that much, and after 3 epoch, it increased only by 1% and became 98%.

G. Model Comparison of our paper

In TABLE I, we have created a comparison table for all the models that we have trained. We can see that our dataset performs well for the decision tree classifier, and the naïve Bayes and logistic regression classifier, it performs very poorly. By analyzing the other model metrics, we can say that maximum of our model is not biased. It can detect both the class pretty well. The highest accuracy from machine learning models is 90% for the decision tree. For NLP models, both LSTM and BERT give high accuracy 95% and 98%, respectively. For LSTM, we were able to find precision, recall, and f1-score, but for BERT, we can only find accuracy.

TABLE I: Models Accuracy Comparison

Models	Precision	Recall	F1-Score	Accuracy
Logistic Regression	74%	72%	73%	74%
Naïve Byes	74%	73%	73%	74%
Decision Tree	90%	89%	89%	90%
Support Vector Machine	76%	76%	76%	77%
LSTM	94%	95%	94%	95%
BERT				98%

H. Model Comparison with others work

Our study's best model was BERT which has 98% accuracy. This model is compared with some previous works models with respect to accuracy in TABLE II. Our best model BERT has higher accuracy than the other works.

TABLE II: Models Accuracy Comparison with related works

Reference	Method Name	Accuracy
In study [1]	CNN+LSTM with PCA	96%
In study [4]	SVM	93.5%
Our study	BERT	98%

IV. Conclusion and Future work

Finding the accuracy of news that is available on the internet is critical nowadays. It has recently been discovered that various online platforms significantly influence disseminating misleading information and spreading fake news to serve several purposes and benefit many people. As a result, there is a growing demand for automated false news identification systems that are accurate and efficient. This proposed system has demonstrated a model and technique for detecting false news using machine learning and NLP methods. The components for spotting fake news are described in the report. Various feature extraction methods, such as Regex, Tokenization, Stopwords, Lemmatization, NLP, TF-IDF, were used in this suggested system. For classification, several models have been used, such as Logistic Regression, Decision Tree, Naïve Bayes, Support Vector Machine (SVM), Long Short-Term Memory (LSTM), Bidirectional Encoder Representation from Transformers (BERT). For the machine learning model Logistic Regression, Decision Tree, Naïve Bayes, and SVM, we got 73.75%, 89.66%, 74.19%, and 76.65%, respectively. But the highest accuracy we git is for the NLP method, which is 95% for LSTM and 98% for BERT, a language model.

We have tried almost all the popular models to predict fake news for machine learning. Only a decision tree gives the best accuracy. For both of the NLP methods have higher accuracy. In the future, we can try to increase the accuracy of the machine learning models that we got lower accuracy. There are already many related works where they got higher accuracy with SVM, but we didn't, so we should try to find out the main reason behind it. In terms of accuracy, we already got more than enough accuracy now, and we can try to test our model on a whole new fake news dataset to evaluate it further.

References

[1] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi and B. -W. On, "Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)," in IEEE Access, vol. 8, pp. 156695-156706, 2020, doi: 10.1109/ACCESS.2020.3019735.

[2] T. Jiang, J. P. Li, A. U. Haq, A. Saboor and A. Ali, "A Novel Stacking Approach for Accurate Detection of Fake News," in IEEE Access, vol. 9, pp. 22626-22639, 2021, doi: 10.1109/ACCESS.2021.3056079.

[3] S. I. Manzoor, J. Singla and Nikita, "Fake News Detection Using Machine Learning approaches: A systematic Review," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 230-234, doi: 10.1109/ICOEI.2019.8862770.

[4] A. Jain, A. Shakya, H. Khatter and A. K. Gupta, "A smart System for Fake News Detection Using Machine Learning," 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2019, pp. 1-4, doi: 10.1109/ICICT46931.2019.8977659.

[5] UTK Machine Learning Club, (2017, July). Fake News, Version 1. Retrieved August 6, 2021 from https://www.kaggle.com/c/fake-news/data.

[6] S. A. Yousif, I. Elkabani, V. W. Samawi, R. Zanout, "The Effect of Combining Different Semantic Relations on Arabic Text Classification", in WCSIT, Vol. 5, pp. 112-118, 2015.

[7] S. A. Yousif, H. Y. Wahed, N. M. G. Al-Saidi, "Extracting a new fractal and semi variance attributes for texture images", in AIP, Vol. 2183, 2019, doi: https://doi.org/10.1063/1.5136199.

[8] Fletcher, S. and Islam, M., 2019. Decision Tree Classification with Differential Privacy. ACM Computing Surveys, Vol. 52(4), pp.1-33, 2019, doi: https://doi.org/10.1145/3337064.

[9] B. Brick, Aylien, July 2016, KDnuggets, accessed: 6th August 2021, "https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html".