# Fake News Detection using Natural Language Processing, Machine Learning and Deep Learning

Aakash Patel, Nisarg Patel, Malav Joshi
MSc. Big Data Analytics - Trent University
aakashpatel@trentu.ca, nisargpatel@trentu.ca, malavjoshi@trentu.ca

## I. Keywords

NLP, Machine Learning, Deep Learning, News, Prediction, LSTM, Word2Vec, SVM, TF-IDF, Logistic regression, Random Forest

## II. ABSTRACT

The proliferation of fake news and its propagation on social media has become a major concern due to its ability to create devastating impacts. Different machine learning approaches have been suggested to detect fake news. However, most of those focused on a specific type of news (such as political) which leads us to the question of dataset-bias of the models used. In this research, we conducted a study to assess the performance of different applicable machine learning, deep learning and natural language processing approaches on a dataset which includes news articles from various types of subject. We used various types of pre-processing algorithms and also explored a number of advanced pre-trained language models for fake news detection along with the traditional and deep learning ones and compared their performances from different aspects for the first time to the best of our knowledge. We find that Bi-LSTM and similar Random Forest perform the best for fake news detection. We believe that this study will help the research community to explore further and news sites/blogs to select the most appropriate fake news detection method.

## III. Introduction

In our modern era, where the internet is ubiquitous, everyone relies on various online resources for news [3]. Along with the increase in the use of social media platforms like Facebook, Twitter, etc. News spread rapidly among millions of users within a very short span of time. In between all these, there comes fake news. Fake news also referred to as hoax news occupies a large sphere of cyberspace today worldwide [7]. Cyber technology's wide reach and fast spread contribute to its menace. Today, publicity through such fake news on cyberspace has been adopted by States, institutions [5] as well as individuals for various reasons and varied forms. Often sensational news is created and spread through social media to achieve the intended end. The spread of fake news has far-reaching consequences of the creation of biased opinions to sway election outcomes for the benefit of certain candidates [9]. On the other hand, it may also involve narration of a true fact, however, it is deliberately exaggerated. This may also include titling the web pages with misleading titles or taglines in order to seize the attention of readers. Moreover, spammers use appealing news headlines to generate revenue using advertisements via click baits. Such misinformation may lead to committing offenses, social unrest, financial frauds upon such misrepresentation. This may also affect the importance of serious news media. The further danger lies in other electronic media using this as a source for their news, thereby carrying forward the further spread of such news. The problem is to identify the authenticity of the news and online content. Equally, the important issue is to identify the bots involved in spreading false news. In this paper, we aim to perform binary classification of various news articles available online with the help of concepts pertaining to Natural Language Processing and Machine Learning. We aim to provide the user with the ability to classify the news as fake or real.

## IV. Our Contribution

Fortunately, there are some computational techniques which can be useful to classify fake news but they use fact-checking websites like "PolitiFact" and "Snopes". Furthermore, there are a number of repositories maintained by researchers that contain lists of websites that are identified as ambiguous and fake but the problem with these resources is that human expertise is required to identify articles as fake. Second, we can get fake news articles from multiple domains and there is a lack of solutions available which can be generalized to identify fake news on all domains. Here, in the research, news data will be used and classification of text data will be done with the help of machine learning, deep learning and natural language processing algorithms.

## V. Related Work

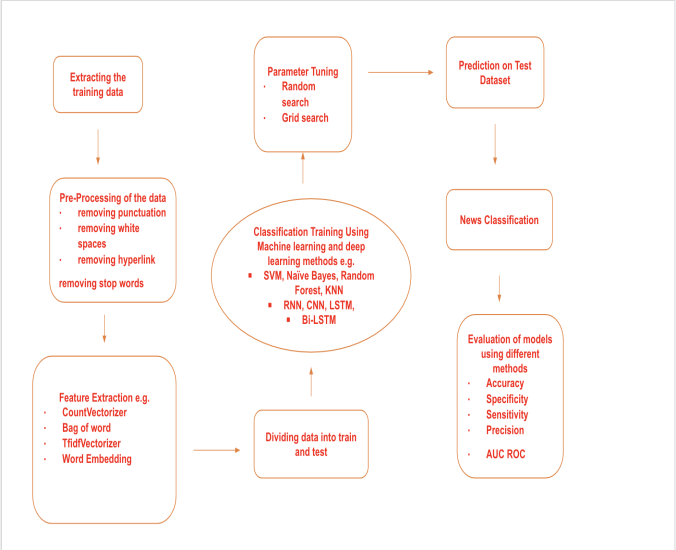
Fake news or misinformation on social media has gained a lot of attention due to the exponential usage of social media. Earlier one group tried to detect Covid related fake news and developed Cross-SEAN: cross-stitch based semi-supervised end-to-end neural attention model [1] which leverages a large amount of unlabeled data. Tavishee and Hemant [2] used the LSTM neural network to differentiate false news from the original ones. Veronica and her group members [4] focus on the automatic identification of fake content in online news using comparative analysis. For which they have built two novel data sets covering seven different news domains. A survey conducted on NLP for fake news detection recommended nine requirements for fake news detection corpus. Nguyen Vo, a student of Ho Chi Minh City University of Technology (HCMUT)Cambodia, did his research on fake news detection and implemented it in 2017. He used Bi-directional GRU.

with Attention mechanism in his project fake news detection; Yang Yang et al. l. originally proposed this mechanism.He also used some Deep learning algorithms and tried to implement other deeplearning models such that Auto-Encoders, GAN, CNN. SamirBajaj of Stanford University [11] published a research paper on fake news detection. He detects fake news with the help of the NLP perspective and implements some other deep learning algorithm. He took an authentic data set from the Signal Media-News data set A research conducted in Lucknow, a smart system for detection of fake news using a machine learning algorithm named Na¨ıve Bayes and Support vector machine.Iftikhar and Ovais [7] proposed how to use a machine learning ensemble approach for the automated classification of news articles. A paper published by Veronica and her teammates where they developed a fake news classification model and trained and tested the model on two datasets, which were collected using two different ways. The model mainly relies on a combination of lexical, syntactic, and semantic information,as well features representing text readability properties.A number of studies have primarily focused on detection and classification of fake news on social media platforms such as Facebook and Twitter . At a conceptual level, fake news has been classified into different types; the knowledge is then expanded to generalize machine learning (ML) models for multiple domains. In the current fake news corpus, there have been multiple instances where both supervised and unsupervised learning algorithms are used to classify text. However,most of the literature focuses on specific data sets or domains,most prominently the politics domain . Therefore, the algorithm trained works best on a particular type of article's domain and does not achieve optimal results when exposed to articles from other domains. Since articles from different domains have unique textual structure, it is difficult to train a generic algorithm that works best on all particular news domains. So,to solve this problem our main focus in this paper is to build system that can identify whether the given article from particular domain is fake or not. We have articles from different domains and in general we can say this is a very generalized-data which can help us to create a system that can help us to find-out fake articles from many domains not a particular domain.As mentioned in the workflow model we will apply various-machine and deep learning algorithms and we will compare them to check which model gives us the better output.

## VI. Data Source

The size of the data set is approximately 78 MB. This data set is designed a larger and more generic Word Embedding over Linguistic Features for Fake News Detection data set of 31,445 news articles with 16,477 real and 14,951 fake news. This data set is consist of popular news from different types of subjects to prevent over-fitting of classifiers and create more generalised model which can be applicable for all fields. Data set contains four columns: Id (starting from 0); Title (about the text news heading); Text (about the news content); Date (when news is published) ; subject ( kind of news like politicsNews, worldnews, News, political, left-news, Government News, USNews, Middle-east ; and Label (0 = fake and 1 = real). There is no duplicate and missing entry in the dataset.

## VII. System Architecture




## VIII. Data Pre-processing

In this step, all the rows containing news data required some preprocessing before giving it to the model. First, all the rows containing null or duplicate values were dropped. Second, we converted all the data into lowercase because the same words written in different cases are considered as different entities by the computer. The next step was to remove all the numbers, punctuation, hyperlink, and stop words. By removing all the low-level information our model will focus more on important information. Furthermore, The English language has several variants of a single term. The presence of these variances in a text corpus results in data redundancy when developing NLP or machine learning models. Such models may be ineffective and to build a robust model, it is essential to normalize text by removing repetition and transforming words to their base form through stemming. Hence, we applied Word Stemming to our data which is a technique that lowers inflection in words to their root forms by removing the prefix and suffix.

• CountVectorizer : CountVectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. This is helpful when we have multiple such texts, and we wish to convert each word in each text into vectors (for using in further text analysis).

• Bag of word : Whenever we apply any algorithm in NLP, it works on numbers. We cannot directly feed our text into that algorithm. Hence, Bag of Words model is used to preprocess the text by converting it into a bag of words, which keeps a count of the total occurrences of most frequently used words.

## IX. Feature Extraction

TF-IDF can be described as a well-defined standard method used to manipulate natural language processing and creating a vector space model for extracted features [10]. In the text, the meaning of the term is evaluated. The importance of the

word is evaluated in the document. Significance is increased in proportion to the number of times the word has appeared in that document. In comparison with the inverse of the word itself in the entire set of documents. Essentially, the TF-IDF measurement is related to the term t that takes:

• A lower value in the case where a term t appears fewer times in the document or appears in several documents;

• A higher value in the case where a term t occurs multiple times in a small number of documents;

• A lower value in the case where a term t occurs in nearly wholly documents.

Let D = [ d1, d2, . . . , dn ] be an entire group of documents, and t represents a term in that group. The term frequency-inverse document size is calculated as follows:

$$TF - IDF(t, d, D) = TF(t, d)IDF(t, D)$$

Mainly, TF(t, d) characterizes the term t frequency in document d (in other words, the number of times a term appears in the document), which is represented as:

$$TF(t, d) = F(t, d)|d|$$

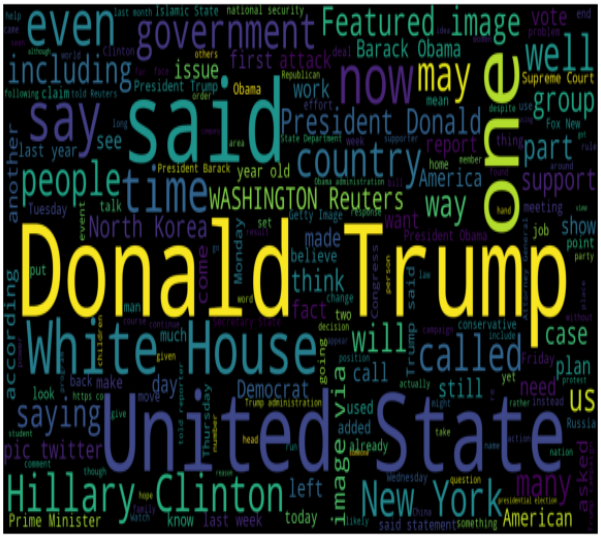
F(t,d) represents how often a term t has occurred in document d, and the denominator is the length of d, which is represented as its own terms' cardinality. The inverse document frequency IDF (t, D) can be defined below:

$$IDF(t, D) = log|D||d|tEd|$$

The denominator is responsible for characterizing the number of documents in which a term t has occurred [10].

## X. Exploratory Data Analysis (EDA)

For visualizing and analyzing the data we are going to use the word cloud data visualizing tool which helps us rough estimate the words that have the highest frequency in the data. Word clouds or tag clouds are graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. The larger the word in the visual the more common the word was in the document. N-gram model visualization which will display the frequency of the words that appears the most using 1-gram, 2-gram, etc.




This image represents the occurrence of the word in the data set. From the above figure we can see Even, said, Donald Trump, One, White House, America, Hillary Clinton, Image, Featured, Country, United state, time, Obama, are the most common word which appears in our text data set. Noe, we will check the distribution of words when news is either fake and non-fake.

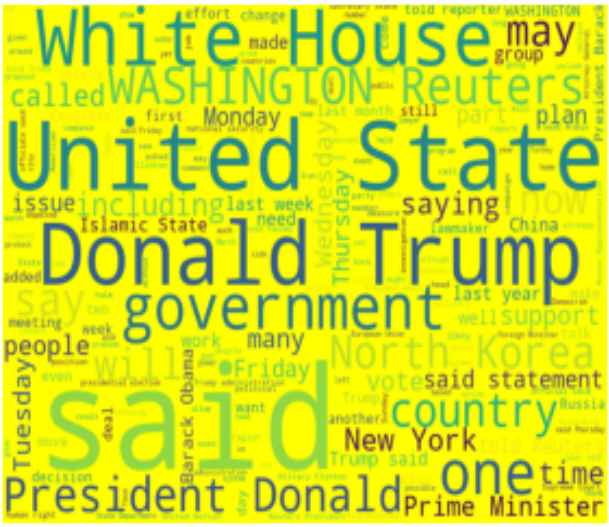



Image 2 : In this Word cloud Of True Data we can get an overview of which words have the most frequency when news are genuine. Words like Donald Trump ,White House ,United State ,Said ,New York ,President ,Donald , Prime Minister ,Washington Reuters ,saying ,statement are very common.

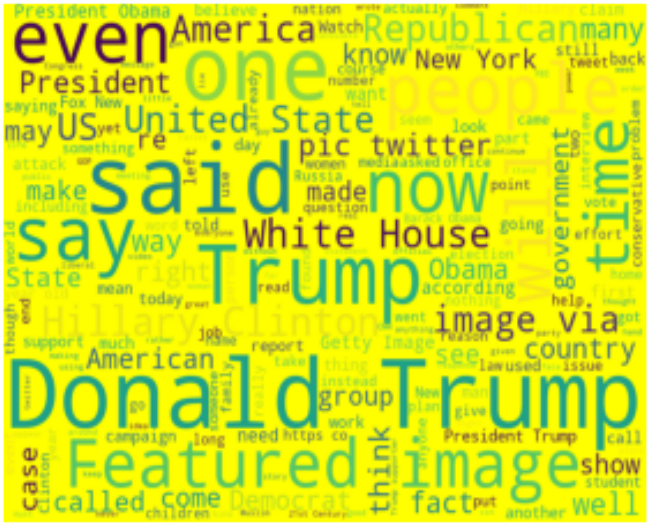



Image 3: It represent the Word cloud of False Data, we can see that it also have some words that present in true image word cloud. From the above figure Even, Donald Trump ,One, White house, America ,Image ,Featured ,Country ,United state, time, Obama ,American are the most common word which appear in our text data set.

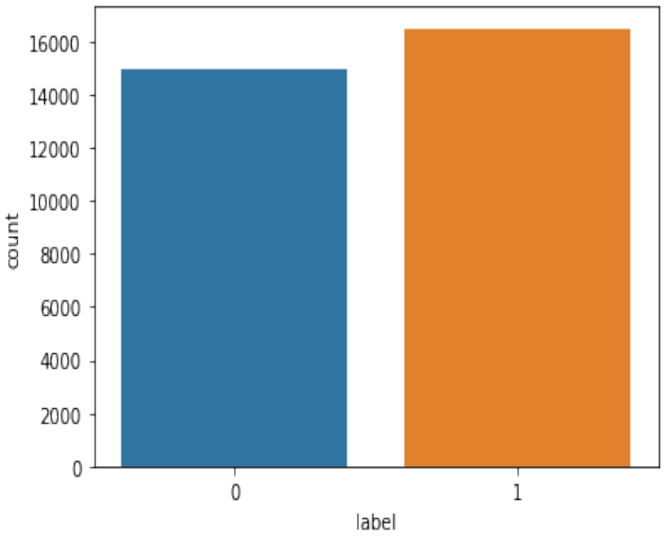



Image 4 : From the above count plot, it can be clearly seen that data is balanced as both label have approximately have same number of records. Here, in the graph 1 is represented as Unreliable or fake news while 0 as Reliable or true data. Distribution of labels tells us that, there are 16477 instances for 1 whereas 14951 rows for 0. Percentage of reliable news is 48 and non reliable news is 52. So we don't need to apply any method to make data balanced.

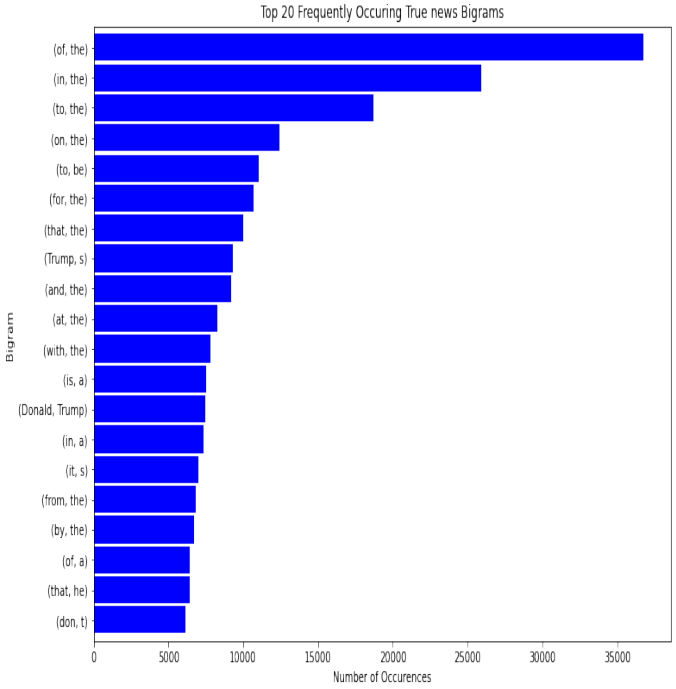



Image 5: Here from the plot it can be seen that pair of (of, the) have the highest number of frequently in this data set. This pair occurred more than 35000 times. After that (in, the),(to, the), and (on, the) have an occurrence of 25000 times, 15000 times, and 10000 times respectively. Moreover, all remaining pairs are occurring less than 10000 times. Here, from the graph, It is clear that we need to remove all these English prepositions as they will affect the models to learn other important words.

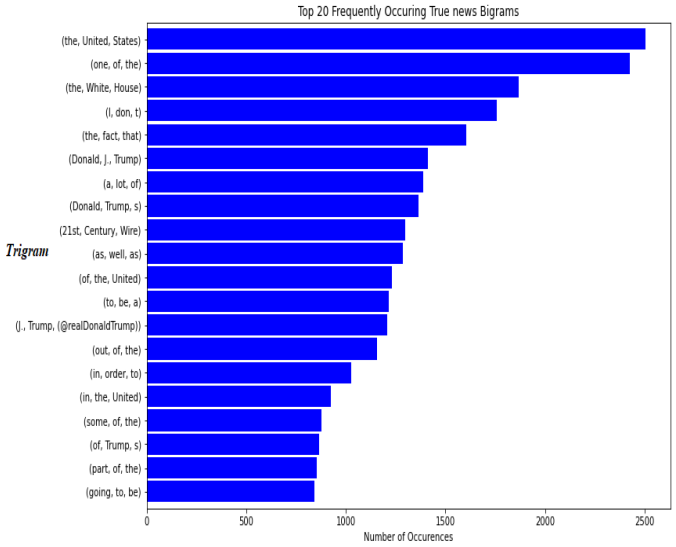



Image 6: From the above trigram it can be seen that(the, united, states) occurred 2500 times. Followed by (one, of, the) which occurred almost 2400 times. (the,white,house) ,(I ,don't) and (the, fact, that) these three pairs have an occurrence between 1500 and 2000 times. There are five pairs that occurred less than 1000 times and 10 pairs which occurred between 1000 and 1500 times.

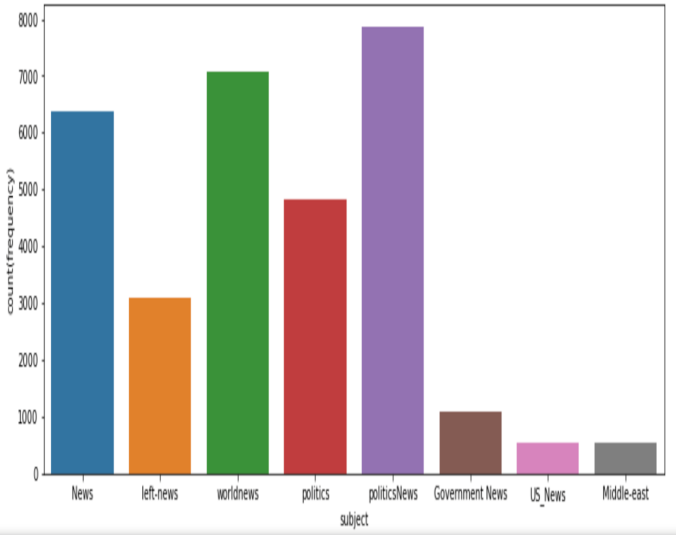



Image 7: From the above bar chart it can be seen that the majority of the news is from politics with a news count of approximately 8000. Then comes world news with approximately 7000 counts followed by US news, government news, and news from the middle east.

## XI. Proposed Solution:

SVM: In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. It creates a decision boundary line with maximum margins to the support vectors. These support vectors are critical points created from the dataset that optimize the hyper plane. SVMs can have nonlinear boundaries, defined by 'kernels' such as sigmoid, RBF, or polynomial. These kernels have parameters that need to be optimized to maximize accuracy. We conducted

many experiments on our SVM model in order to optimize its accuracy. We tuned many parameters and recorded the results from each trial. We began by testing various different kernel types for the SVM model (Linear, Radial Basis Function, Polynomial, and Sigmoid). Kernels allow for the inputted data to be transformed in order for the model to more accurately and efficiently determine the decision boundary and classify points. We then explored the linear kernel further by testing various values of C, a parameter that determines how large the margins are in the SVM model. We also conducted additional experiments on the Radial Basis Function (RBF) kernel function. We tested various values for gamma, a parameter that determines how far each data point's influence reaches.We also tested various C values for the Radial Basis Function. Additionally, we decided to run experiments on the polynomial kernel function. We tested various polynomial degrees and recorded the accuracy of each.

Random Forest: It is a learning method that operates by constructing multiple decision trees. The final decision is made based on the majority of the trees and is chosen by the random forest.There are a lot of benefits to using Random Forest Algorithm, but one of the main advantages is that it reduces the risk of overfitting and the required training time. Additionally, it offers a high level of accuracy. Random Forest algorithm runs efficiently in large databases and produces highly accurate predictions by estimating missing data.

Why should we use Random Forest in some cases instead of Neural Network? Random Forest is less computationally expensive and does not require a GPU to finish training. A random forest can give us a different interpretation of a decision tree but with better performance. Neural Networks will require much more data than an everyday person might have on hand to actually be effective. The neural network will simply decimate the interpretability of your features to the point where it becomes meaningless for the sake of performance. While that may sound reasonable to some, it is dependent on each project.

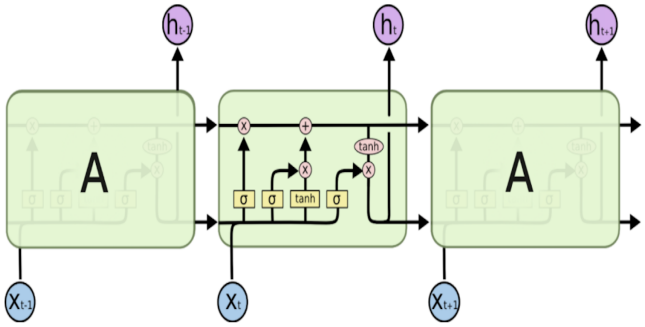
If the goal is to create a prediction model without care for the variables at play, by all means, use a neural network, but we'll need the resources to do so. If an understanding of the variables is required, then whether we like it or not, typically what happens in this situation is that performance will have to take a slight hit to make sure that we can still understand how each variable is contributing to the prediction model.

K-nearest neighbors: The KNN algorithm is useful when we are performing a pattern recognition task for classifying objects based on different features.it classifies a data point based on its neighbors' classifications. It stores all available cases and classifies new cases based on similar features. K in KNN is a parameter that refers to the number of nearest neighbors in the majority voting process. In this particular problem with the help of parameter tuning, we have identified the best k value for our model. Since the KNN algorithm requires no training before making predictions, new data can be added seamlessly, which will not impact the accuracy of the algorithm.

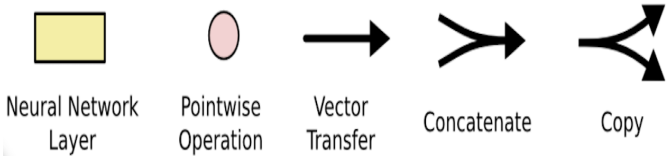
Logistic Regression: Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is dichotomous: i.e. binary. Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables. The independent variables can be nominal, ordinal, or interval type. The name "logistic regression" is derived from the concept of the logistic function that it uses. The logistic function is also known as the sigmoid function. The value of this logistic function lies between zero and one.

Naive Bayes: A Naive Bayes classifier is a type of supervised learning algorithm that is used for the classification task. A Naive Bayes classifier is a simple classifier, which is an application of the Bayes theorem. It is the probabilistic classification method as it uses the likelihood probability to predict the purpose of the classification task. The model used by a Naive Bayes classifier makes strong conditional independence assumptions. The conditional independence assumption is that given a class, the predictor or feature values are independent. There is no correlation between the features of a certain class.

LSTM: Long short-term memory networks, usually called LSTM – are a special kind of RNN. They were introduced to avoid the long-term dependency problem. In regular RNN, the problem frequently occurs when connecting previous information to new information. If RNN could do this, they'd be very useful. This problem is called long-term dependency.

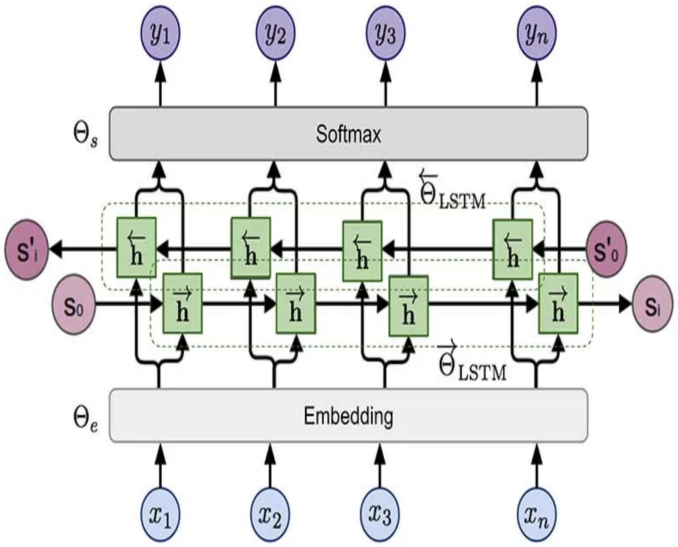



The repeating module in LSTM consists of four interacting layers.




To remember the information for long periods in the default behavior of the LSTM. LSTM networks have a similar structure to the RNN, but the memory module or repeating module has a different LSTM. The block diagram of the repeating module will look like the image above.

BI-LSTM: Bidirectional long-short term memory, usually called BI-LSTM is the process of making any neural network have the sequence information in both directions backward (future to past) or forward(past to future). In bidirectional, our input flows in two directions, making a bi-lstm different from the regular LSTM. With the regular LSTM, we can make input flow in one direction, either backward or forward. However, in bi-directional, we can make the input flow in both directions to preserve the future and the past information.

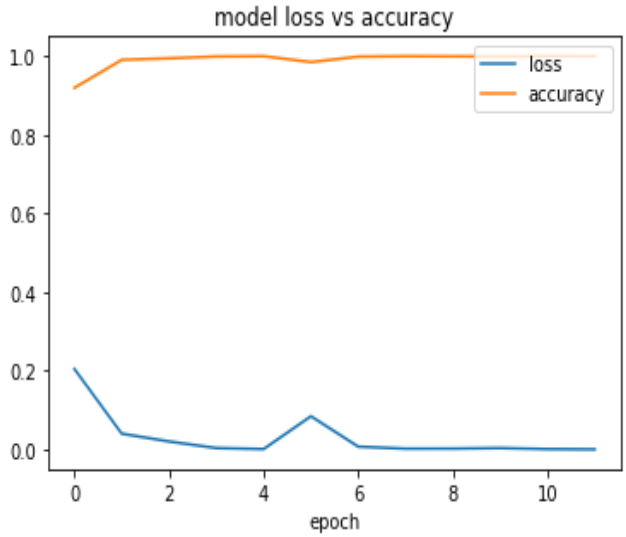



In the diagram, we can see the flow of information from backward and forward layers. BI-LSTM is usually employed where the sequence to sequence tasks are needed. For our problem, we have used both LSTM and Bi-LSTM on our dataset.

## XII. Result and Discussion

| Algorithm | | precision | recall | f1-score | Accuracy |
|---|---|---|---|---|---|
| Naive Bayes | (0) | 0.94 | 0.95 | 0.95 | 0.95 |
| | (1) | 0.95 | 0.95 | 0.95 | |
| Random Forest | (0) | 0.98 | 0.99 | 0.99 | 0.99 |
| | (1) | 0.99 | 0.99 | 0.99 | |
| KNN | (0) | 0.82 | 0.94 | 0.87 | 0.87 |
| | (1) | 0.94 | 0.81 | 0.87 | |
| SVM | (0) | 0.99 | 0.98 | 0.99 | 0.99 |
| | (1) | 0.98 | 0.99 | 0.99 | |
| Logistic | (0) | 0.98 | 0.99 | 0.98 | 0.98 |
| Regression | (1) | 0.99 | 0.98 | 0.98 | |
| LSTM | | | | | 0.95 |
| Bi-LSTM | | | | | 0.9822 |

In this section, we have compared the result of all five algorithms that we performed and chose the best model. Firstly, we have built the models with the default parameters after which for better accuracy we performed hyperparameter tuning. For KNN with n=13, we got an accuracy of around 87 percent. For SVM we used the RBF kernel with gamma values 1 and 0.1, C values 0.1 and 1. We got C=1 and gamma=1 after tuning which increased the accuracy of the model to 98 percent. For the random forest, the biggest challenge was the parameter tuning because there were many important parameters like n-estimators, max-features, max-depth, min-samples-split, min-samples-leaf, and bootstrap. However, even without performing hyperparameter tuning, the algorithm perform very well on the dataset and gave an accuracy of around 99 percent. After experimenting with the machine learning models we went ahead with deep learning-based models. We used LSTM and BI-LSTM for the classification task. Both of the models performed very well, but BI-LSTM outperformed LSTM by performing with an accuracy of 98 percent while LSTM was 3 percent shorter than the prior.




Here, we can see the performance of the BI-LSTM. It is clear that the accuracy line stays near the 1 throughout every epoch, and the loss is almost zero. Thus, the model has performed well in training.

During the phase of building models, we knew that Hyperparameter tuning at a large scale is very different from applying tuning to small data sets. A single model can take days to weeks to train, which means we must exploit parallel computing which can maximize resource efficiency to make the problem tractable, both in terms of performing Hyperparameter search and training considered model efficiently in a distributed fashion. To illustrate it, we tried to apply tuning to a random forest model using 6 parameters. However, even after waiting for 2 days, we did not get our result so we had to stop tuning and move on to other algorithms. Moreover, while tuning the SVM model, the most difficult task was to select the appropriate kernel, the value of gamma, and parameter C. As the range of parameter increase, the system requires more computation power and time. Furthermore, at the time of implementing lstm model, We used only 2 layers in our algorithm because it requires a lot of data to train a model and if we would have added more layers then there were chances that our model could have led us to overfitting.

To answer the question of how our work is better than others, we have read several papers and found out that most of the researchers have focused on one type of news e.g. sports, politics. Furthermore, those who have used data from more diverse topics were lacking in implementing and comparing different types of models. Here, we have kept both aspects in our minds. A Smart system for fake news detection using machine learning a paper written by Anjali Jain [6], used the Naive Bayes and NLP methods and they achieved maximum accuracy of 93 percent. while we used the same methods with different parameters and pre-processing and we got better results. Another paper [7] published by Iftikar Ahmed where they used the algorithms such as linear SVM, random forest,

and LSTM. However, the Bi-LSTM model they used has only 62 percent because of the small dataset. Moreover, the best performing algorithm was random In our case, we achieved 98 percent accuracy on LSTM and 99 percent with Bi-LSTM. A benchmark [12] study done by Junaed Younus khan used both machine learning and deep learning algorithms. In that study, they have achieved quite a decent accuracy but the model was not generalized to broader news article topics which can be seen as their best performing models did not achieve well on newer data. The size of our dataset is 2 times bigger than the data they used and all models have an accuracy got more than 90 percent.

## XIII. CONCLUSION

In this study, we present an overall performance analysis of different approaches to our dataset using various pre-processing techniques. We found that Bi-LSTM, Random forest, SVM, and LSTM models have achieved better performance on datasets. We also find that Random forest and SVM can attain similar results to neural network-based models on a dataset when the dataset size is sufficient. The performance of LSTM-based models greatly depends on the length of the dataset as well as the information given in a news article. The results and findings based on our comparative analysis can facilitate future research in this direction and also help the organizations (e.g., online news portals and social media) to choose the most suitable model that is interested in detecting fake news. Furthermore, they can also use hyperparameter tuning methods to find out the best combinations of parameters to get desired results. Our future work in this direction will focus on designing models that can detect misinformation and health-related fake news that are prevalent in social media during the COVID-19 pandemic. And also we will try to deploy the model using the Flask framework so one can have access to find out whether the news is fake or not using our link.

## REFERÊNCIAS


[1] William Scott Paka, Rachit Bansal, Abhay Kaushik, Shubhashis Sengupta, Tanmoy Chakraborty, Cross-SEAN: A cross-stitch semi-supervised neural attention model for COVID-19 fake news detection, Applied Soft Computing, Volume 107, 2021, 107393, ISSN 1568-4946,

[2] Tavishee Chauhan, Hemant Palivela, "Optimization and improvement of fake news detection using deep learning approaches for societal benefit", Volume 1, Issue 2, 2021, 100051, ISSN 2667-0968

[3] Uma Sharma, Sidarth Saran, Shankar M. Patil, 2021, Fake News Detection using Machine Learning Algorithms, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) NTASU – 2020 (Volume 09 – Issue 03),

[4] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea, "Automatic Detection of Fake News",volume 1708.07104,2017

[5] Zhou, Yichao Sheng, Ying Vo, Nguyen Edmonds, Nick Tata, Sandeep. (2022). Learning Transferable Node Representations for Attribute Extraction from Web Documents. 1479-1487. 10.1145/3488560.3498424.

[6] Jain, Anjali Shakya, Avinash Khatter, Harsh Gupta, Amit (2019). A smart System for Fake News Detection Using Machine Learning. 1-4.10.1109/ICICT46931.2019.8977659

[7] Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, Muhammad Ovais Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods", Complexity, vol. 2020, Article ID 8885861, 11 pages, 2020.

[8] J. C. S. Reis, A. Correia, F. Murai, A. Veloso and F. Benevenuto, "Supervised Learning for Fake News Detection,"in IEEE Intelligent Systems, vol. 34, no. 2, pp. 76-81, March-April 2019, doi: 10.1109/MIS.2019.2899143.

[9] Monther Aldwairi, Ali Alwahedi, "Detecting Fake News in Social Media Networks",Zayed University, vol. 141, page 215-222, ISSN 1877-0509, 1-1-2018

[10] Erra, Ugo Senatore, Sabrina Minnella, Fernando Caggianese, Giuseppe. (2015). Approximate TF–IDF based on topic extraction from massive message stream using the GPU. Information Sciences. 292. 143–161. 10.1016/j.ins.2014.08.062.

[11] Samir Bajaj, Stanford University, CS 224N - Winter 2017, samirb@stanford.edu

[12] Junaed Younus Khan, Md. Tawkat Islam Khondaker, Sadia Afroz, Gias Uddin, Anindya Iqbal, "A benchmark study of machine learning models for online fake news detection, Machine Learning with Applications", Volume 4, 2021, 100032, ISSN 2666-8270

[13] Ray Oshikawa and Jing Qian and William Yang Wang, "A Survey on Natural Language Processing for Fake News Detection", journal CoRR,volume 1811.00770, 2018, http://arxiv.org/abs/1811.00770