FAKE NEWS DETECTION USING MACHINE LEARNING APPROACHES

Purvi Harniya, 1814023
Department of Information Technology
KJ Somaiya College of Engineering
Mumbai, India
purvi.h@somaiya.edu

Neelay Jagani, 1814024
Department of Information Technology
KJ Somaiya College of Engineering
Mumbai, India
neelay.j@somaiya.edu

Esha Gupta, 1814025
Department of Information Technology
KJ Somaiya College of Engineering
Mumbai, India
esha.gupta@somaiya.edu

***Abstract*— In recent years, fake news has become more prevalent in the online world as a result of the rapid growth of online social networks as well as multitude of political and commercial purposes. Fake information is purposely or unintentionally spread throughout the internet. The massive dissemination of fake news has left an indelible mark on people and culture. The timely identification of the fake news is a prime goal of increasing social network's information reliability. This paper reviews various fake news detection methods involving feature extraction methods like Count vectorizer, TF-IDF Vectorizer, Word Embedding and also, different classification algorithms like SVM, Logistic Regression and Gradient Boosting, Random Forest, Decision Trees, KNN and XG-Boost.**

***Keywords*— SVM, Logistic Regression, Gradient Boosting,Random Forest, Decision Trees, KNN, Feature Extraction, Machine learning, Fake news detection, Semi-supervised learning, Graph neural networks**

I. Introduction

Data has been increasing at an unprecedented range in an exponential manner and is producing 2.7 quintillion bytes of data everyday that comes as 1.7MB of data each second and hence it is important to have machine learning classification algorithms in place which will help us differentiate between the true and false news. The definition of fake news is information that pushes people down the wrong road. Fake news is spreading like wildfire these days, and people are sharing it without confirming it. This is frequently done to promote or impose specific views, and it is frequently accomplished through political agendas. To produce online advertising revenue, media outlets must be able to draw viewers to their websites. As a result, it is vital to recognise phoney news.

Machine learning is an AI application that allows a system to learn without having to be explicitly designed. Machine learning is based on data and will learn from it. Machine learning is not the same as the traditional method. For this research paper, we will be using supervised learning where we use labelled examples to train our model, which means the machine first learns from those examples before doing the task on unlabeled data, and hence we split our data into test and train data.

II. Types of Data in Social Media Posts

There are three major ways by which social media networking sites read news items:

- Text: Computational linguistics analyzes text (multilingual), focusing on the genesis of text semantically and methodically. Because many of the posts are written in the form of texts, much work has been carried out into analysing them.
- Multimedia: Several types of media are combined in a single post. Audio, video, photos, and graphics may all be included. This is highly appealing, because it captures the attention of the visitors without requiring them to read the content.
- Hyperlinks: Hyperlinks allow the post's creator to cross-reference to other sources, gaining viewers' trust by confirming the post's genesis. Cross-reference to other social media networking sites, as well as the embedding of photos, is common practise.

III. Types of Fake News

Fake news is categorized as follows:

- Visual-based: False news generally contains a lot of images to make it visually appealing and make use of doctored or photoshopped images to deceive users.
- User-generated news: This sort of falsified news is generated by phoney accounts and is targeted to certain audiences, which might reflect specific age groups, gender, culture, or political affiliations.
- Knowledge-based: These posts provide scientific (so-called) explanations to some unresolved problems, leading people to feel they are genuine. For example, natural therapies for high blood sugar levels in the human body.

- Style based: Pseudo Journalists who impersonate and mimic the style of some accredited journalists write style-based posts.
- Stance-based: It is a portrayal of true statements in such a way that its meaning and purpose are altered.

IV. METHODOLOGY

The fake news detection model is built using the following steps:

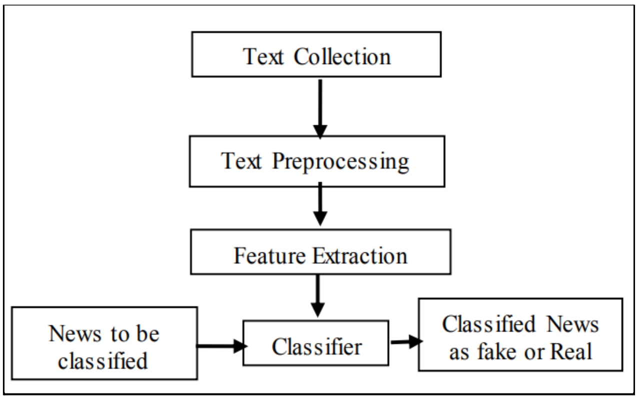

Fig 1: Model architecture

The steps in detail are explained below:

A. Text Collection

The text collection process is carried out by referring to Kaggle datasets. Data is gathered from 244 websites. It is made up of approximately 13000 posts that were recorded over the course of 30 days. Whereas the training data set is 18574 posts and the testing data set is 9149. Other alternatives are the ISOT fake news dataset and the KDnugget dataset. The ISOT dataset was compiled exclusively from real-world sources. The true news was gathered by crawling Reuters.com articles and bogus news taken from untrustworthy websites highlighted by Politifact and Wikipedia. The majority of the articles were published From 2016 until 2017, This dataset contains a total of 44898 records. 21417 are true news (labelled as 1), whereas 23481 are false news. (labelled as 0). The title and text are among the features (news body), the subject, date, and label are all required. The news topics are divided into several categories, including 'politicsNews,' 'worldnews,' 'News,' and 'political', 'Government News,' 'Left News,' 'US News,' 'Middle East.'

Title	Text	Subject	Date	Label
UK transport police leading in...	LONDON (Reuters) - British counter-terrorism.	worldnews	September 15, 2017	REAL
Pacific nations crack down on...	WELLINGTON (Reuters) - South Pacific island nation.	worldnews	September 15, 2017	REAL
Three suspected al Qaeda...	ADEN, Yemen (Reuters) - Three suspected al Qaeda...	worldnews	September 15, 2017	REAL
Chinese academics prod Beijing...	BEIJING (Reuters) - Chinese academics are publicly.	worldnews	September 15, 2017	REAL
Classic! Kid Rock Hits Back At...	Not much to say after this classic response from.	politics	Sep 2, 2017	FAKE
'My Pillow' CEO Mike Lindell...	Who hasn t seen his commercials over and over and...	politics	Sep 1, 2017	FAKE
Bitter John McCain Calls Trump...	What the heck! Senator John McCain just admitted..	politics	Sep 1, 2017	FAKE
Muslim Activist Caught Sending...	This woman has no shame1 Muslim activist Linda...	politics	Sep 1, 2017	FAKE

Fig 2: Demonstration of ISOT Dataset

Id	Title	Text	Label
7614	Globalization Expressway to...	If humans were largely moral and ethical beings...	FAKE
10294	Watch The Exact Moment Paul...	Google Pinterest Digg Linkedin Reddit...	FAKE
7060	Now Malaysia Dumps US for...	Now Malaysia Dumps US for Chinese Naval Vessels...	REAL
10142	Bernie supporters on Twitter...	Kaydee King (@KaydeeKing) November 9, 2016 The...	FAKE
875	The Battle of New York: Why...	It's primary day in New York and front-runners...	REAL
6903	Tehran, USA	I'm not an immigrant, but my grandparents are...	FAKE
7341	Girl Horrified At What She...	Share This Baylee Luciani (left), Screenshot of...	FAKE
95	'Britain's Schindler' Dies at...	A Czech stockbroker who saved more than 650 Jewish...	REAL

Fig 3: Demonstration of KDnugget Dataset

B. Text Pre-processing

Various techniques are used on the text data before we feed this data into machine learning algorithms, using techniques such as stop word removal, tokenization, sentence segmentation, and punctuation removal. These processes can greatly assist us in selecting the most relevant phrases and improving model performance. Because both of our datasets are derived from real-world news stories, there are many nonsensical urls that contain no information. So model building starts by cleaning up the data by deleting these urls and unwanted text.

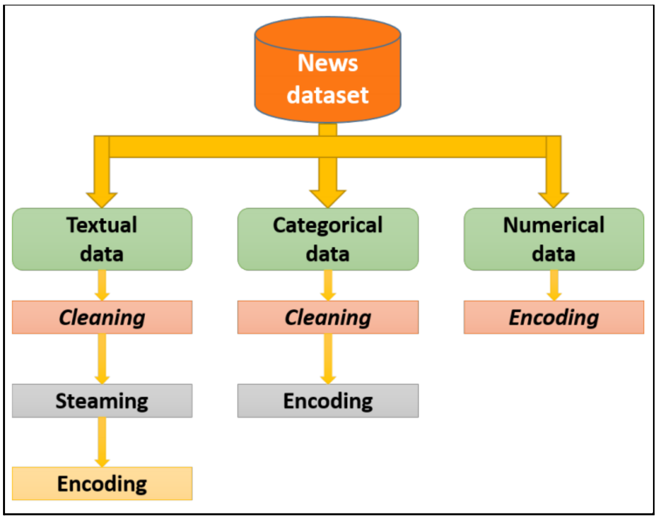

Fig 4:Preprocessing of different categories of news characteristics

- **Textual data**: Represent the text written by the author in a news and pre-processed by the following operations:
  1. Cleaning: eliminating stop words and special characters.

2. Stemming: transforming the useful words into roots.
3. Encoding: transforming all the words of the comment into a numerical vector. This needs two steps: the combination of two techniques, namely, bag of words and N-grams, then the application of the TF-IDF method on the result.

- **Categorical data**: Represent the source of the news such as TV channel, newspaper or magazine, and its author. The pre-treatment of these data is performed through two steps:
  – Cleaning: eliminating special characters and transforming letters into lowercase.
  – Encoding: for sources we used a label encoding. For authors, the method of encoding is to convert the author's names into digital numbers, so that authors from the same source are close to each other compared to authors from other sources. A list is created containing two fields, the first for the source and the second for its authors, then replaced each author by its index number by adding the sum of the sizes of the previous sources plus one.

- **Numerical data**: Represent the date of posting the comment and the sentiment given by the text. Since the date is already represented by a numerical value, we only split it into three unique values: day, month and year. For the sentiment given by the text, we calculate the sum of the sentiment degrees of the words. According to the experts, each word has a degree of sentiment which allows it to be classified into three classes: –If the sum is less than 0, the feeling is negative. –If the sum exceeds 0, the sentiment is positive.
  –If the sum is 0, the feeling is neutral.

C. Feature Extraction

Before feeding the text into the machine learning algorithm, the text must be parsed in order to evaluate words, and words must be encoded as integers or floating-point values. A bag of words and word embedding are two vectorization methods used in the proposed system .

Bag of Words

Machine learning from text employs a technique known as the bag of words, which takes any text and counts the frequency of words after removing all stop words. Following tokenization, it must be quantized and used on appropriate ML classifiers.

Countvectorizer

The count vector is one of the easiest methods to tokenize a collection of archives including documents and create a jargon of recognised words, and encode new reports using that jargon. An encoded vector is a response with the length of the entire jargon and a whole number indicating the number of occurrences of each word appearing in the record.

TF-IDF

This is used to convert content to vectors while keeping the semantics of the word in mind. TF is a typical tokenization technique that calculates document similarity by counting the number of words in the documents. Using the TF approach, each document is represented by a vector containing the word counts. Then, for each vector, the sum of its elements will be one, converting the word counts into probabilities. The TF-IDF metric is a weighting metric that is extensively used in text classification problems. It is used to assign a score to each term in the document, indicating the relevance of the term. The relevance of a phrase in this strategy grows in proportion to its frequency in the dataset.

Eq.3 is obtained by multiplying the term frequency (1) by the inverse document frequency (2).

TF = (Number of occurrences of term t in a document)/(Number of terms in the document) (1)

$$TF_{i,j} = \frac{n_{i,j}}{\Sigma_k n_{i,j}} \qquad (1)$$

$$IDF = log(N/n) \qquad (2)$$

where N denotes the number of archives and n denotes the number of times a term t appears in word.

$$TF - IDF\, value\, of\, a\, term = (TF * IDF)$$
(3)

Word Embedding using Spacy

To generate a vector, this concept of feature representation is used. A dense matrix with a low dimension is obtained here. Spacy is a Natural Language Processing library that generates numeric vectors that represent words for word embedding. It consumes a lot of memory and has negative side effects.

D. Classifiers

To make predictions in this proposed system, various Machine learning algorithms are used. To generate vectors, TF-IDF is used, and all of the listed algorithms are used to investigate the best calculation for counterfeit news recognition, with the same procedure applied to word embedding and count vectorization.

1. Support Vector Machine

SVM is one of the most used models for binary and multi-classification tasks. It is a supervised machine learning classifier that has been used by numerous academics to solve binary and multi-classification issues. In a binary classification issue, the instances are separated by a hyperplane in such a way that $w^T x + b = 0$, where w is a dimensional coefficient weight vector that is normal to the hyperplane. The bias term b represents the values offset from the origin, while data points are represented by x. The fundamental job of SVM is to determine the values of w and b.

The Lagrangian function can be used to solve w in the linear case. The data points on the maximal border are referred to as support vectors. As a result, the answer to w can be stated mathematically in the form of equation 4.

$$w = \sum_{i=1}^{n} \alpha_i Y_i X_i \quad (4)$$

In equation 6, n represents support vectors, and Yi represents the target class label, which corresponds to sample x. $y_i(w^T x_i + b) - 1 = 0$ can be used to calculate the term bias b.

SVM is used to precisely classify machines. Drawing decision boundaries is known as creating a hyperplane that separates two classes. Unoptimized decision boundaries may result in misclassification; to overcome this, SVM are regarded as important by examining extreme cases. Some functions could be used to convert nonlinear SVM to linear SVM. Calculation generates the most ideal hyperplane, which normally personifies new data.

2. Logistic Regression

Logistic regression is a popular classification approach in machine learning to predict the values of the predictive variable y in a binary classification issue, where y = [0, 1]. The negative class is represented by 0 and the positive class by 1.

To classify two classes, 0 and 1, a hypothesis $h(\theta) = \theta^T X$ will be constructed, and the classifier's output threshold is when $h\theta(x) = 0.5$. If the value of hypothesis $h\theta(x) \geq 0.5$, it predicts y = 1, indicating that the news is true. If the value of hypothesis $h\theta(x) < 0.5$, it predicts y = 0, indicating that the news is false.
Hence, the prediction of logistic regression under the condition $0 \leq h\theta(x) \leq 1$ is done. Logistic regression sigmoid function can be written in equation 5 as follows:

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

where $g(z) = 1/(1 + x^{-z})$ and $h\theta(x) = 1/(1 + x^{-\theta^T X})$
Similarly, the logistic regression cost function can be written in equation 6 as follows:

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

(6)

3. Decision Trees

DT is a well-known supervised learning algorithm. On a wide range of problems, researchers employ tree-based ensemble models such as Random Forest or Gradient Boosting. The primary principle behind DT is that it creates a model to forecast the value of a dependent component by learning numerous decision rules derived from the entire set of data. Decision Tree features a top-down structure and tree-like shapes, with nodes that can only be a leaf node that is bound with a label class or a decision node that is responsible for making decisions. The process of making decisions and forecasts is simply understood through the Decision Tree. However, because it is a slow learner, it may perform poorly on small datasets.

The most important learning process in DT is choosing the right attribute. To tackle this problem, different trees employ different measures, such as information gain in the ID3 algorithm and gain ratio in the C4.5 algorithm. Assume discrete attribute A has n distinct values, and Di is the set that contains all samples in training dataset D that have a value of i. For attribute A, the gain ratio and information gain can be calculated as follows:

$$Gain(A, D) = Entropy(D) - \sum_{i=1}^{n} \frac{|D_i|}{|D|} Entropy(D_i)$$

(7)

$$GainRatio(A, D) = \frac{Gain(A,D)}{IV(A)}$$

(8)

where intrinsic value of attribute A can be calculated as:

$$IV(A) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} log2 \frac{|D_i|}{|D|}$$

(9)

4. Random Forest

Random Forest is an ensemble of unpruned decision trees bagged with a randomised set of features at each split. Each tree in the random forest makes a forecast and the forecast with the highest number of votes becomes our final prediction. According to the No Free Lunch theorem, no algorithm is always the most accurate; hence, RF is more accurate and robust than individual classifiers.

The decision tree's high variance was reduced to a low variance by using row sampling and feature sampling. The number of decision trees could be determined using hyperparameters. It's an ensemble algorithm that combines more than one calculation of the same or distinguishing kind for characterizing objects.

The random forest algorithm can be expressed as:

$$F(x) = arg\,max \left\{ \sum_{i=1}^{n} T(A(B, \theta_k)) \right\} \quad (10)$$

If F(x) represents the random forest model, j represents the target category variable, and F represents the characteristic function. To ensure the decision tree's diversity, the sample selection of random forest and the candidate attributes of node splitting are both random. The random forest algorithm's pseudocode is presented in the algorithm below.

TABLE I: ALGORITHM FOR RANDOM FOREST

Algorithm: Random Forest	
Require:	Training set (m is the number of training set, f is the feature set)
Ensure:	Random forest with msub CART trees
1:	Draw Bootstrap sample sets msub with replacement
2:	Choose a sample set as the root node and train in a completely split way
3:	Select fsub randomly from f and choose the best feature to split the node by using minimum principle of Gini impurities
4:	Let the nodes grow to the maximum extent. Label the nodes with a minimum impurity as leaf node
5:	Repeat steps 2-4 until all nodes have been trained or labeled as leaf nodes.
6:	Repeat steps 2-5 until all CART has been trained
7:	Output the random forest with msub CART trees

5. Gradient Boosting

Gradient boosting in machine learning is used for regression and classification. It's a way for boosting. Leaf indicates an initial prediction, which is log(odds) for classification; this is turned into a probability using the logistic function (10).

$$Probability = \frac{e^{log(odd)}}{1+e^{log(odd)}} \quad (11)$$

6. XG-Boost

It's a very strong Gradient boosting classifier. Designed for usage with large, complex datasets. It is an ensemble strategy that prevents overfitting and thereby regularizes boosting. In all circumstances, it is scalable. It can handle sparse data as well as parallel and distributed processing, making learning faster and more efficient.

7. K-NN

K-NN is a well-known machine learning algorithm. The K-NN techniques are extremely straightforward. Given a test sample, it initially determines the k nearest neighbours based on a distance measure. Then, using the major vote strategy, it predicts the class label of the test instance. Sometimes the classification performance of K-NN is poor, owing to the curse of dimensionality. K-NN is also a lazy learning method that can take a long time to classify data. Algorithm given below describes the core methods of the K-NN algorithm.

TABLE II: KNN ALGORITHM

Algorithm: KNN Algorithm	
1:	for all unlabeled data u do
2:	for all labeled data v do
3:	compute the distance between u and v
4:	find k smallest distances and locate the corresponding labeled instances v1,.. vk
5:	‘ assign unlabeled data u to the label appearing most frequently in the located labeled instances

6:	end for
7:	end for
8:	End

V. Proposed Algorithm

The following table represents the algorithm to build a complete fake news detection model :

TABLE III: ALGORITHM FOR MODEL BUILDING

Algorithm	
Input:	News Content
1.	Convert text to lowercase
2.	Remove punctuations, digits, stop words from text
3.	Repeat: Input: Receive each news article Calculate count vector for it Append the count vector to count_feature vector Until the end of news article
4.	Repeat Input: Receive each news article Calculate the TF-IDF vector for it Append the count vector to tfidf_feature vector Until the end of news article
5.	Repeat Input: Receive each news article Calculate the Spacy vector for it Append the spacy vector to Spacy_feature vector Until the end of news article
6.	Parse count_feature vector, tfidf_feature vector and spacy_feature vector into classifier Return feature vector gives us highest accuracy
7.	Build model with the feature vector
Output:	Predict label of news - Fake or Real

VI. Results and Analysis

ML classifiers were evaluated using a variety of metrics. Accuracy measures the difference between predicted and actual labels. Precision is used for retrieval of information. Precision is calculated as a percentage of the total positive results predicted by the model.The proportion of genuine positive results that are relevant. Recall is True positive rate or True negative rate. F1score is a combination of precision and recall.

$$Accuracy(Acc)\% = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (12)$$

$$Recall(Re)\% = \frac{TP}{TP+FN} \times 100 \quad (13)$$

$$Precision(Pre)\% = \frac{TN}{TN+FP} \times 100 \quad (14)$$

$$F1 - Score = 2\frac{(precision)(recall)}{precision+recall} \quad (15)$$

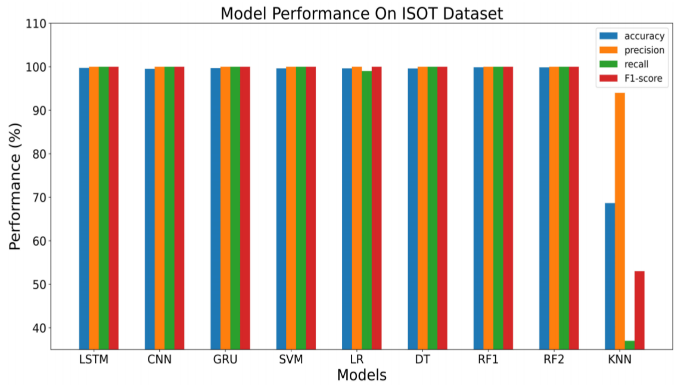

Fig 5: Comparison of Model Performance on ISOT dataset

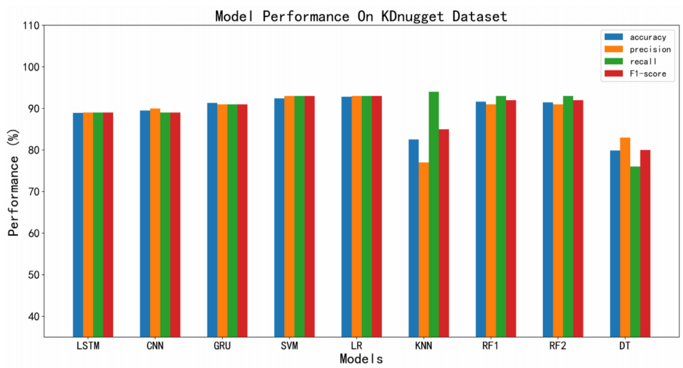

Fig 6: Comparison of Model Performance on KDnugget dataset

Table IV shows performance evaluation of ML classifiers with a count vector where different metrics are used. Table V shows performance evaluation of ML classifiers with TF-IDF where different metrics are used, same with Table VI for word embedding.

TABLE IV: CLASSIFICATION METRICS WITH COUNT VECTORIZER

Classifiers	Performance evaluation			
	Accuracy	*Precision*	*Recall*	*Fscore*
SVM Linear	0.91	0.90	0.91	0.90
Logistic Regression	0.93	0.91	0.92	0.92
Decision Tree	0.82	0.80	0.81	0.80
Random Forest	0.88	0.94	0.79	0.86
XG-Boost	0.89	0.88	0.88	0.88
Gradient Boosting	0.89	0.89	0.88	0.88
Neural Network	0.94	0.94	0.93	0.93

TABLE V: CLASSIFICATION METRICS WITH TF-IDF VECTORIZER

Classifiers	Performance evaluation			
	Accuracy	*Precision*	*Recall*	*Fscore*
SVM Linear	0.94	0.93	0.93	0.93
Logistic Regression	0.93	0.93	0.91	0.92
Decision Tree	0.82	0.79	0.80	0.80
Random Forest	0.90	0.94	0.83	0.88
XG-Boost	0.89	0.89	0.88	0.88
Gradient Boosting	0.90	0.89	0.88	0.88
Neural Network	0.93	0.93	0.91	0.92

TABLE VI: CLASSIFICATION METRICS WITH WORD EMBEDDINGS

Classifiers	Performance evaluation			
	Accuracy	*Precision*	*Recall*	*Fscore*
SVM Linear	0.87	0.88	0.83	0.85
Logistic Regression	0.87	0.88	0.82	0.85
Decision Tree	0.73	0.64	0.69	0.69
Random Forest	0.84	0.85	0.79	0.82
XG-Boost	0.83	0.84	0.76	0.80
Gradient Boosting	0.83	0.84	0.76	0.80
Neural Network	0.90	0.92	0.86	0.89

Figure 7 indicates that a neural network outperforms other classifiers employing a count vectorizer, with an accuracy of 0.94. Figure 8 depicts a comparison of several classifiers using the TF-IDF vectorizer, with SVM Linear outperforming the others with 0.94 accuracy. Figure 9 depicts a comparison of different classifiers with word embedding, with the neural network outperforming the others with 0.90 accuracy.

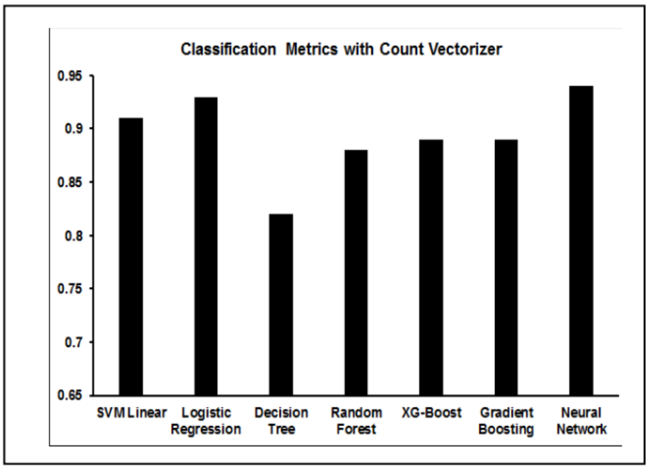

Fig 7:Classification Metrics with Count Vectorizer

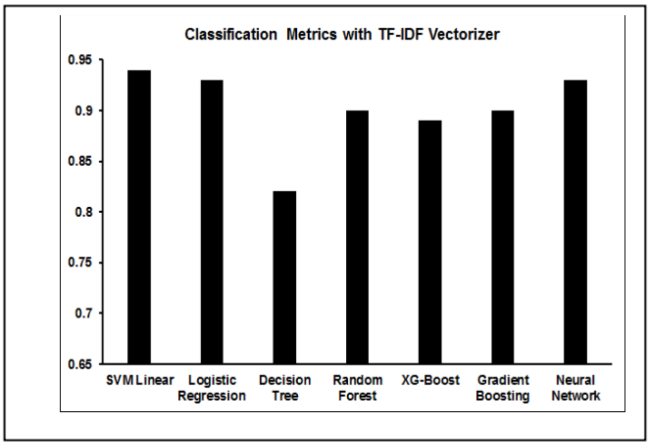

Fig 8: Classification Metrics with TF-IDF Vectorizer

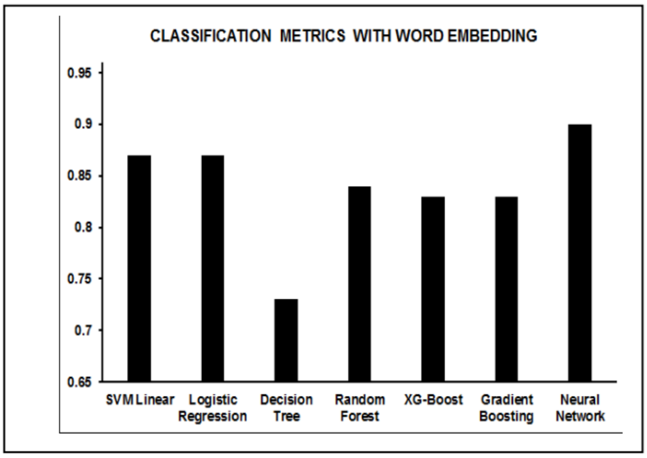

Fig 9: Classification Metrics with Word Embedding

VII. OTHER APPROACHES

Neural Networks

A neural network is made up of densely interconnected processing pieces called neurons that cooperate to solve very specific problems. Deep neural networks are one of the most popular deep learning techniques in machine learning. Deep neural networks and advances in pre-trained word embedding have become a source of fresh creative ideas in recent years. However, the entire model ignores the importance of key words and instead treats all words as a network of input, with no particular consideration given to significant words.

The final text classification results should improve substantially as a result of combining the benefits of the two methods, rebuilding the neural network model, and increasing the weight keywords in the network.

VIII. Conclusion

In recent years, deceptive content has grown in popularity, and its influence on online users has increased. The authors of this paper have surveyed on fake news detection methods involving three different feature extraction methods like Count vectorizer, TF-IDF Vectorizer, Word Embedding and also, different classification algorithms. The greatest accuracy attained by classification techniques is by using SVM Linear classification algorithm with TF-IDF feature extraction with 0.94 accuracy, as shown in TABLE IV, V, VI. Even though both Neural Network and Count Vectorizer achieve the same accuracy, the Neural Network takes longer to train and is more sophisticated. So, in the proposed system, Linear SVM is used, which is less difficult and takes less time to compute.

IX. Future Work

Deep learning methods and sentiment analysis to categorise news with high accuracy might be considered in the future, and more useful text such as the publication of the news, URL domain, and so on, could be extracted for the process.

For more accuracy, a dataset with a higher number of articles from various sources can be employed, as it includes more jargon and notable material.

X. References

[1] N. Smitha and R. Bharath, "Performance Comparison of Machine Learning Classifiers for Fake News Detection," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 696-700.

[2] A. Benamira, B. Devillers, E. Lesot, A. K. Ray, M. Saadi and F. D. Malliaros, "Semi-Supervised Learning and Graph Neural Networks for Fake News Detection," 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2019, pp. 568-569.

[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.

[4] A. Kesarwani, S. S. Chauhan and A. R. Nair, "Fake News Detection on Social Media using K-Nearest Neighbor Classifier," 2020 International Conference on Advances in Computing and Communication Engineering (ICACCE), 2020, pp. 1-4.

[5] K. Poddar, G. B. Amali D. and K. S. Umadevi, "Comparison of Various Machine Learning Models for Accurate Detection of Fake News," 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), 2019, pp. 1-5.

[6] K. Rajesh, A. Kumar and R. Kadu, "Fraudulent News Detection using Machine Learning Approaches," 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1-5.

[7] N. F. Baarir and A. Djeffal, "Fake News detection Using Machine Learning," 2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH), 2021, pp. 125-130.

[8] 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.