|  |
| --- |
| **Tittle: Fake News Detection Using NLP**  Team Member:Vinoth Kumar.R  E-mail:vinoth0704vino@gmail.com  Phase 1 \_submission document  ABSTRACT  Social media is one of the very powerful media in spreading information. People are interested in sharing without any proper checking of any sort of false information. Unstructured text data may be classified into meaningful categorical classifications using text classification, which is a typical study area in the discipline of Natural Language Processing (NLP). The main contribution of this article is to identify a finest framework to tackle the fake news problem with the NLP and Machine Learning techniques. In this empirical research, the fake news data is analysed with the different combinations of Vectorizers and Machine Learning Classifiers. From the experimental results on five benchmark datasets namely fake\_real\_news dataset extracted from Kaggle, COVID-19 Constrain, Politifact, ISOT and Gossipcop, it is observed that the fake news detection with the combination of TF-IDF Vectorizer and Passive-Aggressive Classifier outperforms the other existing methods.  **Keywords:** Machine Learning, Pre-processing, Vectorizer, Classifiers, Natural Language Processing    1.INTRODUCTION  Various difficulties are being created by fake news nowadays, from satirical pieces to fake stories and deliberate government propaganda in some publications. False news and public distrust in the media are developing issues in our society that have serious consequences. The term “fake news” refers to a tale that intentionally misleads the public, but in recent months, social media has begun to redefine the term.  The importance of disinformation during the time of COVID19 pandemic is a subject of weighty attention, particularly following the emergence of multiple variants of Coronavirus. Factually false and deceptive stories produced primarily for the goal of gaining money through page views became known as “fake news.” With the use of this research, we want to develop a model that can properly identify whether or not a piece of content is fake news.  Rasmus Kleis Nielsen [1], says that “the problems of disinformation in a society like India might be more sophisticated and more challenging than they are in the West.” Journalists, media organizations, technology firms and Western policymakers who wish to understand the character, scope and future |

trajectory of disinformation could do well to study Indians’ experience in a corporate polarized society, and to see what the problems of disinformation look like. Kabir Upmanyu [2], says that WhatsApp in India has become one of the Major fake news providers. False news is also suspected to propagate on WhatsApp because it is circulated between small groups of friends and families who commonly believe.

Kevin Ponniah [3], showed that an upsurge in nationalism led to the sharing of false news in BBC research. It was generally assumed that messages from family and friends from WhatsApp would be respected and distributed without fact check. The consequences of fake news and the use of the word are introduced in the 21st century. The Web was introduced to the public in the 1990s and was meant to provide them with the ability to access information. With the passage of time the Internet has risen to unimaginable heights with tons of information that always allow the Internet to host a wealth of unwanted, false and misleading information that anyone can make. Due to a variety of factors, identifying fake news on social media is incredibly difficult. Fake news data is difficult to gather, and it is even more difficult to manually designate it as such [4]. Fake news thrives in an atmosphere where there are a lot of people using social media.Rubin et.al. [5], define fake news detection as the prediction of the chances of a specific news to be purposely deceptive or delusory. Spreading fake news on social media is quite prevalent [6]. Emotional, syntactic, material resemblance, similarity of style, and semantic inconsistency were used to identify false reviews gathered on Amazon.[7]. Text classification is the process of grouping the words together in a text. It can evaluate the text and apply specified tags using Natural Language Processing (NLP). Text classifiers may arrange, organize, and categorise any type of text, including documents, medical studies, files, and information found on the internet. A classifier is a machine learning system that intelligently organises data into one or more “classes.” A spam email classifier is one of the most common instances. To train a machine learning model, vectorization is a technique of extracting the text’s different characteristics. With the use of numerical vectors, text may be transformed into characteristics that the model can use to improve itself.

This research aims at identifying a consistent combination of Vectorizer and Classifier in detecting fake and real news. In order to achieve this objective, five datasets were used along with four classifiers namely Multinomial Naïve Bayes, SVM, Passive Aggressive and Random Forest and two vectorizers namely TF-IDF and Count Vectorizer.

Classification approach was utilised by Gee et.al. [8] to recognise spam profiles based on spam reports from users. They’ve used the Twitter API to gather ordinary user profiles and spam profiles from the “@spam” Twitter handle, and they’ve expressed the data in JSON. Naive Bayes method with a 27 percent error rate and SVM algorithm with a 10 percent error rate were used to classify it in CSV format. Wang et.al. [9], studied the suspicious behaviours of spam accounts on Twitter. They applied machine learning methods to automatically differentiate spam accounts from

|  |
| --- |
| 2.RELATED WORKS  Various researches have been undergone by many researchers to find out the best  classifiers which gives a maximum accuracy in identifying the fake and real news. The excerpts from some of such researches were mentioned here in order to get a clear picture of the objective of this research article.  normal ones. Data generated by spam or fake users is substantial, according to Dutse et al. [10]. One out of every 200 social media postings and one out of every 21 tweets are spam, according to estimates. They've come up with a new way to tell spam from non-spam social media messages, and they’ve revealed more about the Twitter behaviour of the spammers.  According to Jia et al. [11], SVM beat both the rule-based classifier and the Decision Tree  classifier when it came to spam identification on the Internet. Aldwairi et.al [12] suggested an easy but efficient strategy to allow users to install an easy tool in their private browser to detect and filter prospective Clickbaits. Depending on Precision, Recall, F-Measure and ROC, the classifiers are compared. Wang and Alex Hai [13] used Naive Bayesian classification to evaluate a dataset of about 25,000 accounts, 500,000 tweets, and 49 million follower/friend interactions. The Bayesian classifier produced the highest overall output, according to the experiment's F-measure results. Malicious URLs can be identified using lexical and host-based features that can be detected using analytical tools, such as those used by Ma, Justin et al. [14]. By collecting and analysing hundreds of potentially suggestive traits using classifier methods such as SVM, Naive Bayes, and Logistic Regression, they proved that these methods may learn extraordinarily predictive models.  A subjective LSTM architecture presented by Kudugunta et.al. [15] would allow them to identify tweet-level bots by using tweet content and metadata. 3000 Twitter bots were studied by utilising Logistic Regression, Random Forest, and AdaBoost Classifiers. Mc Cord et al. [16] established a model for spam identification based on user-based and content-based factors. Classifiers that can distinguish between sceptical and trusting consumers were tested against four traditional classifiers: Random Forest, Support Vector Machine, Nave Bayesian and K-Nearest Neighbor. They’ve built a prototype to test the detecting method using the specified features.  Zheng et.al., [17] conducted research to find an excellent classifier to find spam messages. SVM (Support Vector Machine)-based spammer detection methods were used to extract characteristics from message content and user social behaviour. Classifiers such as Decision Tree, Nave Bayes, and Bayes Network, all implemented by Weka, were compared to the suggested technique. There are various emotional and point-of-sale (POS) aspects that may be combined with content/user-based factors to distinguish spam tweets from authentic messages on Twitter, one of the most popular social networking websites online. A combination of Twitter's spam detection policies and its own observations of spam behaviour informs its recommendations. Traditional |
|  |

|  |
| --- |
| classifiers like Naive Bayes, Random Forest, Support Vector Machine (SVM), and J48  methods were used to evaluate the effectiveness of the recommended features in spam identification.  A three-level Hierarchical Attention Network (3HAN)-based deep learning-based automated  detector was used by Singhania, Sneha et.al. [19] in order to quickly and precisely detect bogus news. Using a hierarchical bottom-up approach to processing a news item, 3HAN builds a news vector: an effective representation of an input news storey, from the words, phrases, and title. Because of its three levels of focus, 3HAN accorded less weight to some elements of an article than to others. Sherry, Girgis, et al. Liar dataset, which includes 12,836 brief utterances labelled for honesty, subject, context/venue, speaker, state, party, and past history [20], was utilised in this study. Each verdict was accompanied by a detailed analytical report by the labeler. CSI [21] (consisting of three modules: Capture, Score, and Integrate) avoids the expense of arbitrary feature selection by combining neural networks into it. DTC, SVM, LSTM, GRU, and CSI are the methods used for analysis. In a broad method that is not dependent on the data environment or requires distributional assumptions, they employed the characteristics to capture the temporal behaviour and textual content. The TI-CNN approach, used by Yang et. al. [22], was utilised to translate the pictures into text. For detecting false communications, they found that TI-CNN is the best convolution approach available. Convolutional neural networks (CNNs) and Long Short-Term Memory in Neural Networks were proposed by Jain et al. [23] as a new deep learning architecture (LSTM). With the use of knowledge bases such as WordNet and ConceptNet, they included semantic information into the word representation in this model. By giving a more accurate semantic vector representation of the test words, these knowledge bases enhance performance.  According to Jin et.al. [24], rumor detection on microblogs may be improved by using an RNN with an attention mechanism to combine data from text, picture, and social environment. Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models have been combined in an entirely new way through a model named CLSTM by Zhou et al. [25]. Ananth et al.  [26] used machine learning algorithms such as SVM, Decision Tree, KNN, and CNN Max Pooling to analyse 20800 fake and real news data acquired from Kaggle.Propagation Path Classification (PPC) was analysed by Yang Liu et al. [27] using CNN and RNN. Visual Question Answering (VQA), NeuralTalk, att-RNN, and Event Adversarial Neural Network were all used in an experiment carried out by Yaqing Wang et al. [28]. (EANN). Passive-Aggressive Classifier with Hashing Vectorizer, according to Saloni Gupta et.al. [31], outperform other combinations in identifying Fake News. Vectorizer and Classifier combinations that outperform existing approaches in detecting fake news are the subject of this article. |
|  |

|  |
| --- |
| 3.METHODOLOGY  Feature Extraction Vectorizers like TF-IDF and Count Vectorizer are used to do the word embeddings and dataset cleaning in this approach. It’s done by removing stop words like “the,” “when,” and “there,” as well as utilising an n-number of the most often used words, phrases, lowercased or not, as well as using keywords that appear at least a certain number of times in a certain text corpus. The dataset is then divided into two parts: data for testing and data for training. Classifiers like Passive Aggressive Classifier, Multinomial Nave Bayes, Random Forest, and Support Vector Machine are fed the train data after the dataset has been divided. The classifiers train the data and check the relativity of the features in the test data and gives the result as ‘Fake’ or ‘Real’ along with the accuracy of the Classifiers. The classifiers          Fig. 3.1 – Framework of Fake News Detection    train the data and check the relativity of the features in the test data and gives the result as ‘Fake’ or ‘Real’ along with the accuracy of the Classifiers.    The process of fake news detection has been handled in two stages; Vectorization and Classification |
|  |

|  |
| --- |
| **3.1.** Vectorization  It is an NLP approach known as “word embeddings” or “word vectorization” that uses real numbers to map words and phrases from the dictionary to the corresponding vectors of real numbers, which may subsequently be used to create word predictions and meanings. It’s a method for turning text into a graphical representation. Text can be vectorized using Count Vectorizer, TF-IDF Vectorizer, or Hashing Vectorizer, among others.  3.1.1 Count Vectorizer  One of the simplest vectorizers, called the Count Vectorizer [32], employs the number of times a token appears in the text to determine its weight. A sample size of 0.33 and a random state size of 53 is being used.  3.1.2 TF-IDF Vectorizer  TF-IDF [33] weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The significance of a word rises in direct proportion to the number of times it occurs in the text, but is counterbalanced by its frequency in the dataset.   * **Term Frequency (TF):** The frequency with which a term appears in the present document is measured using a metric called Term Frequency (TF). Depending on the length of the text, a certain term may appear more frequently than in shorter ones. To equalize, the word frequency is frequently divided by the input size as shown in Eq. (1)   TF(t) = Number of times term t appears in a document (1)  Total number of terms in the document     * **Inverse Document Frequency (IDF)**: The frequency with which a term appears in the present document is measured using a metric called Term Frequency (TF). Depending on the |
|  |

|  |
| --- |
| length of the text, a certain term may appear more frequently than in shorter ones.Rarer the term more is the IDF score as shown in Eq. (2) and (3).    IDF(t) = loge ( Total number of documents )  Number of documents with term t in it  (2)    Thus,  TF–IDF Score = TF \* IDF (3)  3.1.3 Hashing Vectorizer  In this hashing vectorizer [33], a hashing algorithm is used to discover a feature integer index mapping for the token string name. In order to turn text documents into matrices, this vectorizer uses the sparse matrix that stores the token occurrence count for each document in the collection. The hashing vectorizer calculates the number of 1- and 2-grams in the text, or singletons and consecutive pairs of words (tokens) in articles. The benefits are the vocabulary dictionary does not need to be stored in memory, it is memory scalable for big data sets. Because the fit has no state, it may be used in a streaming or parallel pipeline.  3.2 Classification  Classification is a process of identifying, interpreting, and arranging concepts and things into predetermined groups or “subpopulations” is known as categorization. When a pattern is detected in a subsequent batch of data, it is known as “pattern recognition.” Classification is a sort of pattern recognition. Naive Bayes, Logistic Regression, K-Nearest Neighbors, K-Nearest Neighbors, Decision Trees, and Support Vector Machines are some of the several types of Classifiers.  **3.2.1** Multinomial Naive Bayes Classifier  Naive Bayes is a method of supervised learning based on the Bayes theorem that may be used to solve classification problems. Text classification problems that need a large training dataset frequently make use of this technique. It is possible to quickly create machine learning models that can make accurate predictions by using the Naive Bayes Classifier. In other words, it’s a predictive model that uses probabilities to create predictions about the future. There are several uses for the Nave Bayes Algorithm, including spam filtration and text analytics. An element’s place in a category is determined by the Naive Bayes method. Text analysis can make use of it to determine whether or not certain words or phrases fall within a predetermined “tag”. This is shown in Eq. (4).    (4) |
|  |

|  |
| --- |
| **3.2.2** Passive Aggressive Classifier  Passive Aggressive Online learning algorithms (both for classification and regression) have been proposed by Crammer et.al. [30] as part of the algorithmic family called algorithms. The Passive-Aggressive algorithms are mysterious to many data scientists. But in certain cases, they can be really useful and successful. Passive-aggressive algorithms are frequently used for large-scale learning. In the current market, it is one of the few “online-learning algorithms.” When compared to batch learning, online machine learning algorithms accept input data sequentially and update the machine learning model one step at a time rather than using the entire training dataset at once. There are situations where training the entire dataset is computationally infeasible because of the massive amount of data. This is quite advantageous. As with Perceptron models, passive-aggressive algorithms don’t require a learning rate. Regularization settings exist for them. As an example, in our proposed model, which includes a passive aggressive classifier, the following parameters are included:  In the event of an incorrect forecast, the model penalises the user by this amount. This is known as the regularisation parameter which id denoted as **C**.  Training data iteration limit for the model is denoted as **Max\_iter**.  **tol** is the reason for the stoppage. If (loss > previous loss – tol) is set to None, the model will  stop. The default setting is 1e-3.  **3.2.3** Random Forest Classifier  The supervised classification approach known as the Random Forest [35,36,37] builds  a forest from a collection of trees. The more trees there are in a forest, the more robust it appears.  The more trees in the forest, like in the random forest classifier, the greater the accuracy of the findings.  The features are always discretized at random at each partition. Even with the same training data, max features=n features, and bootstrap=False, the best-found partition may differ if the parameter progress is the same for several partitions specified throughout the search for the best partition. To achieve unambiguous training signs, you must stabilise the random state.  3.2.4. Support Vector Machine (SVM)  It is possible to train an SVM (Support Vector Machine) [38] to assign labels to different  types of data. Classification and regression issues can be solved using this method. Classification is a common problem in Machine Learning, where it is utilised extensively. The SVM approach aims to identify the optimum line or decision limit for categorising n-dimensional space into classes so that following data points may be readily classified into the correct category. A hyperplane is a term used to describe the most ideal decision limit.  SVM selects the extreme points/vectors that help to form the hyperplane. A Support Vector Machine (SVM) is a technique that utilises support vectors as extreme examples.   1. EXPERIMENTAL RESULTS AND ANALYSIS   **4.1 Experimental Setup** |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Using Jupyter Lab and five benchmark datasets from Kaggle, COVID-19 Constrain, Politifact,  ISOT, and Gossipcop, the empirical evaluation is carried out in Python. A Windows 10 PC with an Intel Core i5 - 9300H CPU running at 2.40GHz, 8GB of RAM, and an NVIDIA GEFORCE GTX 1650 graphics card was used for all testing. Table 4.1 provides a thorough breakdown of the five realworld datasets. The training and testing examples were chosen at random, with 67 and 33 percent of the total being taken from each group.  Table 4.1. Properties of experimental datasets       |  |  |  |  |  | | --- | --- | --- | --- | --- | | Sno | Dataset | Fake | Real | Total | | 1 | Kaggle  Fake\_Real | 3154 | 3161 | 7795 | | 2 | Covid 19  Fake\_Real | 3060 | 3360 | 6420 | | 3 | Gossipcop  Fake\_Real | 5322 | 16817 | 22139 | | 4 | ISOT  Fake\_Real | 13728 | 6271 | 19999 | | 5 | Politifact  Fake\_Real | 428 | 567 | 995 | | 6 | Welfake  Fake\_Real | 1306 | 1515 | 2821 | |
|  |

|  |
| --- |
| **4.2. Evaluation Criteria**  Methods for detecting fake news may be assessed based on their confusion matrix and accuracy.    **4.2.1.** Confusion Matrix  A confusion matrix is a visual depiction of the outcomes of a classification activity. The key  to Confusion Matrix is calculating the percentage of correct and incorrect guesses and then dividing it by class using count values. The confusion matrix shows how the classification model becomes confused while making predictions. It exposes not just the number of errors produced by a classifier, but also the kind of errors made by the classifier itself.    **4.2.2.** Accuracy  Accuracy is the percentage of properly identified samples divided by the total number of  samples, as shown in Equation (5)  (5)  Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.    **4.3.** Performance Analysis  The process of fake news detection is implemented on Raw data from the datasets (Kaggle [39], COVID-19 [40], Politifact [41], ISOT [42], Gossipcop [41] and Welfake [43]) which is fed as input to the vectorizers and the components like repeated words, special characters, spaces and non- English words are removed using the vectorizers and then it has been converted into a vector. The test\_train\_split function arranges the data; Four Classifiers namely Multinomial Naive Bayes Classifier, Passive Aggressive Classifier, Random Forest Classifier and Support Vector Machine have been used along with two Vectorizers such as Count Vectorizer and TF-IDF Vectorizer, and the end results of all the combinations are compared to find the best combination of vectorizer and classifier. The stages of empirical analysis are shown in Figure 4.1. |
|  |

|  |
| --- |
| Figure 4.1 Stages of Comparative Analysis    The confusion matrices of the Passive Aggressive classifier with TF-IDF vectorizer and Count Vectorizer for the Fake\_real dataset is shown in Figure 4.2 which depict that the fake news detection with Passive Aggressive and TF-IDF Vectorizer obtains better accuracy with less error rate.      Figure 4.2 Confusion Matrices    For the comparative analysis, in order to find a better combination of classifier and vectorizer,  the accuracies of different combinations of Vectorizer and Classifier are evaluated and presented in Table 4.3.1.  The results in table 4.3.1 clearly show that the combination of Passive Aggressive Classifier, and TF-IDF Vectorizer gives a nominal accuracy of 93.5% in analysing Kaggle fake\_real dataset, 99.2% in analysing ISOT fake\_real dataset and 83.9% in analysing Welfake fake\_real dataset. The Support Vector Machine (SVM) also when combined with TF-IDF vectorizer gave maximum accuracies in the other two datasets. |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The results obtained from different combinations of Vectorizers and Classifiers on all the five datasets depict that the TF-IDF vectorizer performs outstandingly than the count vectorizer. Thecomparison of the resultant accuracies of all classifiers with TF-IDF vectorizer is illustrated in the figure 4.3. Based on the experimental results, one could conclude that, the combination of a Passive Aggressive classifier and TF-IDF vectorizer  Table 4.3.1. Comparative Analysis       |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | **Kaggle Fake\_Re**  **al** | **Covid Fake\_Re**  **al** | **Gossipco p**  **Fake\_Re**  **al** | **ISOT Fake\_Re**  **al** | **Politifact Fake\_Re**  **al** | **Welfake Fake\_Re**  **al** | | **Multinomi al Naïve**  **Bayes** | **COUN**  **T** | 89.3 | 92.1 | 83.7 | 95.5 | 84.2 | 81.1 | | **TF-**  **IDF** | 85.7 | 90.2 | 82.6 | 92.2 | **84.5** | 81.3 | | **Passive**  **Aggressive** | **COUN**  **T** | 89.4 | 92.7 | 79.1 | 98.7 | 80.2 | 83.8 | | **TF-**  **IDF** | **93.5** | 92.9 | 79.7 | **99.2** | 81.8 | **83.9** | | **Support**  **Vector**  **Machine** | **COUN**  **T** | 86.2 | 93.1 | 84.8 | 99.1 | 74.8 | 82.5 | | **TF-**  **IDF** | 92.2 | **93.7** | **85.1** | 98.8 | 83.0 | 83.5 | | **Random**  **Forest** | **COUN**  **T** | 90.5 | 92.0 | 83.0 | 98.9 | 73.9 | 80.9 | | **TF-**  **IDF** | 83.5 | 92.4 | 83.9 | 98.0 | 73.9 | 81.1 | |  |  | **Kaggle Fake\_Re**  **al** | **Covid Fake\_Re**  **al** | **Gossipco p**  **Fake\_Re**  **al** | **ISOT Fake\_Re**  **al** | **Politifact Fake\_Re**  **al** | **Welfake Fake\_Re**  **al** | | **Multinomi al Naïve**  **Bayes** | **COUN**  **T** | 89.3 | 92.1 | 83.7 | 95.5 | 84.2 | 81.1 | | **TF-**  **IDF** | 85.7 | 90.2 | 82.6 | 92.2 | **84.5** | 81.3 | |  | **COUN**  **T** | 89.4 | 92.7 | 79.1 | 98.7 | 80.2 | 83.8 | |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Passive**  **Aggressive** | **TF-**  **IDF** | **93.5** | 92.9 | 79.7 | **99.2** | 81.8 | **83.9** | | **Support**  **Vector**  **Machine** | **COUN**  **T** | 86.2 | 93.1 | 84.8 | 99.1 | 74.8 | 82.5 | | **TF-**  **IDF** | 92.2 | **93.7** | **85.1** | 98.8 | 83.0 | 83.5 | | **Random**  **Forest** | **COUN**  **T** | 90.5 | 92.0 | 83.0 | 98.9 | 73.9 | 80.9 | | **TF-**  **IDF** | 83.5 | 92.4 | 83.9 | 98.0 | 73.9 | 81.1 |                                   CONCLUSION  Fake news classification might benefit from machine learning in a new approach, which is  what this study primarily argues. In this empirical study, the process of fake news detection has been carried out on five benchmark datasets with various combinations of vectorizers and machine |
|  |