

Fake News Detection using Machine Learning Algorithms

Uma Sharma, Sidarth Saran, Shankar M. Patil

Department of Information Technology
Bharati Vidyapeeth College of Engineering
Navi Mumbai, India

Abstract— In our modern era where the internet is ubiquitous, everyone relies on various online resources for news. 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. The spread of fake news has far-reaching consequences like the creation of biased opinions to swaying election outcomes for the benefit of certain candidates. Moreover, spammers use appealing news headlines to generate revenue using advertisements via click-baits. In this paper, we aim to perform binary classification of various news articles available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning. We aim to provide the user with the ability to classify the news as fake or real and also check the authenticity of the website publishing the news.

Keywords—Internet, Social Media, Fake News, Classification, Artificial Intelligence, Machine Learning, Websites, Authenticity.

I. INTRODUCTION

As an increasing amount of our lives is spent interacting online through social media platforms, more and more people tend to hunt out and consume news from social media instead of traditional news organizations.[1] The explanations for this alteration in consumption behaviours are inherent within the nature of those social media platforms: (i) it's often more timely and fewer expensive to consume news on social media compared with traditional journalism, like newspapers or television; and (ii) it's easier to further share, discuss, and discuss the news with friends or other readers on social media. For instance, 62 percent of U.S. adults get news on social media in 2016, while in 2012; only 49 percent reported seeing news on social media [1]. It had been also found that social

media now outperforms television because the major news source. Despite the benefits provided by social media, the standard of stories on social media is less than traditional news organizations. However, because it's inexpensive to supply news online and far faster and easier to propagate through social media, large volumes of faux news, i.e., those news articles with intentionally false information, are produced online for a spread of purposes, like financial and political gain. It had been estimated that over 1 million tweets are associated with fake news "Pizzagate" by the top of the presidential election. Given the prevalence of this new phenomenon, "Fake news" was even named the word of the year by the Macquarie dictionary in 2016 [2]. The extensive spread of faux news can have a significant negative impact on individuals and society. First, fake news can shatter the authenticity equilibrium of the news ecosystem for instance; it's evident that the most popular fake news was even more outspread on Facebook than the most accepted genuine mainstream news during the U.S. 2016 presidential election. Second, fake news intentionally persuades consumers to simply accept biased or false beliefs. Fake news is typically manipulated by propagandists to convey political messages or influence for instance, some report shows that Russia has created fake accounts and social bots to spread false stories. Third, fake news changes the way people interpret and answer real news, for instance, some fake news was just created to trigger people's distrust and make them confused; impeding their abilities to differentiate what's true from what's not. To assist mitigate the negative effects caused by fake news (both to profit the general public and therefore the news ecosystem). It's crucial that we build up methods to automatically detect fake news broadcast on social media [3].

Internet and social media have made the access to the news information much easier and comfortable [2].

Often Internet users can pursue the events of their concern in online form, and increased number of the mobile devices makes this process even easier. But with great possibilities come great challenges. Mass media have an enormous influence on the society, and because it often happens, there's someone who wants to require advantage of this fact. Sometimes to realize some goals mass-media may manipulate the knowledge in several ways. This result in producing of the news articles that isn't completely true or maybe completely false. There even exist many websites that produce fake news almost exclusively. They intentionally publish hoaxes, half-truths, propaganda and disinformation asserting to be real news – often using social media to drive web traffic and magnify their effect. The most goals of faux news websites are to affect the general public opinion on certain matters (mostly political). Samples of such websites could also be found in Ukraine, United States of America, Germany, China and much of other countries [4]. Thus, fake news may be a global issue also as a worldwide challenge. Many scientists believe that fake news issue could also be addressed by means of machine learning and AI [5]. There's a reason for that: recently AI algorithms have begun to work far better on many classification problems (image recognition, voice detection then on) because hardware is cheaper and larger datasets are available. There are several influential articles about automatic deception detection. In [6] the authors provide a general overview of the available techniques for the matter. In [7] the authors describe their method for fake news detection supported the feedback for the precise news within the micro blogs. In [8] the authors actually develop two systems for deception detection supported support vector machines and Naive Bayes classifier (this method is employed within the system described during this paper as well) respectively. They collect the info by means of asking people to directly provide true or false information on several topics – abortion, execution and friendship. The accuracy of the detection achieved by the system is around 70%. This text describes an easy fake news detection method supported one among the synthetic intelligence algorithms – naïve Bayes classifier, Random Forest and Logistic Regression. The goal of the research is to look at how these particular methods work for this particular problem given a manually labelled news dataset and to support (or not) the thought of using AI for fake news detection. The difference between these article and articles on the similar topics is that during this paper Logistic Regression was specifically used for fake news detection; also, the developed system was tested on a comparatively new data set, which

gave a chance to gauge its performance on a recent data.

A. Characteristics of Fake News:

They often have grammatical mistakes. They are often emotionally coloured. They often try to affect readers' opinion on some topics. Their content is not always true. They often use attention seeking words and news format and click baits. They are too good to be true. Their sources are not genuine most of the times [9].

II. LITERATURE REVIEW

Mykhailo Granik et. al. in their paper [3] shows a simple approach for fake news detection using naive Bayes classifier. This approach was implemented as a software system and tested against a data set of Facebook news posts. They were collected from three large Facebook pages each from the right and from the left, as well as three large mainstream political news pages (Politico, CNN, ABC News). They achieved classification accuracy of approximately 74%. Classification accuracy for fake news is slightly worse. This may be caused by the skewness of the dataset: only 4.9% of it is fake news.

Himank Gupta et. al. [10] gave a framework based on different machine learning approach that deals with various problems including accuracy shortage, time lag (BotMaker) and high processing time to handle thousands of tweets in 1 sec. Firstly, they have collected 400,000 tweets from HSpam14 dataset. Then they further characterize the 150,000 spam tweets and 250,000 non-spam tweets. They also derived some lightweight features along with the Top-30 words that are providing highest information gain from Bag-of-Words model. 4. They were able to achieve an accuracy of 91.65% and surpassed the existing solution by approximately 18%.

Marco L. Della Vedova et. al. [11] first proposed a novel ML fake news detection method which, by combining news content and social context features, outperforms existing methods in the literature, increasing its accuracy up to 78.8%. Second, they implemented their method within a Facebook Messenger Chabot and validate it with a real-world application, obtaining a fake news detection accuracy of 81.7%. Their goal was to classify a news item as reliable or fake; they first described the datasets they used for their test, then presented the content-based approach they implemented and the method they proposed to combine it with a social-based approach available in the literature. The resulting dataset is composed of 15,500 posts, coming from 32 pages (14 conspiracy pages, 18 scientific pages), with more than

2, 300, 00 likes by 900,000+ users. 8,923 (57.6%) posts are hoaxes and 6,577 (42.4%) are non-hoaxes.

Cody Buntain et. al. [12] develops a method for automating fake news detection on Twitter by learning to predict accuracy assessments in two credibility-focused Twitter datasets: CRED BANK, a crowd sourced dataset of accuracy assessments for events in Twitter, and PHEME, a dataset of potential rumours in Twitter and journalistic assessments of their accuracies. They apply this method to Twitter content sourced from BuzzFeed's fake news dataset. A feature analysis identifies features that are most predictive for crowd sourced and journalistic accuracy assessments, results of which are consistent with prior work. They rely on identifying highly retweeted threads of conversation and use the features of these threads to classify stories, limiting this work's applicability only to the set of popular tweets. Since the majority of tweets are rarely retweeted, this method therefore is only usable on a minority of Twitter conversation threads.

In his paper, Shivam B. Parikh et. al. [13] aims to present an insight of characterization of news story in the modern diaspora combined with the differential content types of news story and its impact on readers. Subsequently, we dive into existing fake news detection approaches that are heavily based on text-based analysis, and also describe popular fake news datasets. We conclude the paper by identifying 4 key open research challenges that can guide future research. It is a theoretical Approach which gives Illustrations of fake news detection by analysing the psychological factors.

III. METHODOLOGY

This paper explains the system which is developed in three parts. The first part is static which works on machine learning classifier. We studied and trained the model with 4 different classifiers and chose the best classifier for final execution. The second part is dynamic which takes the keyword/text from user and searches online for the truth probability of the news. The third part provides the authenticity of the URL input by user.

In this paper, we have used Python and its Sci-kit libraries [14]. Python has a huge set of libraries and extensions, which can be easily used in Machine Learning. Sci-Kit Learn library is the best source for machine learning algorithms where nearly all types of machine learning algorithms are readily available for Python, thus easy and quick evaluation of ML algorithms is possible. We have used Django for the web based deployment of the model, provides client

side implementation using HTML, CSS and Javascript. We have also used BeautifulSoup (bs4), requests for online scrapping.

A. System Design-

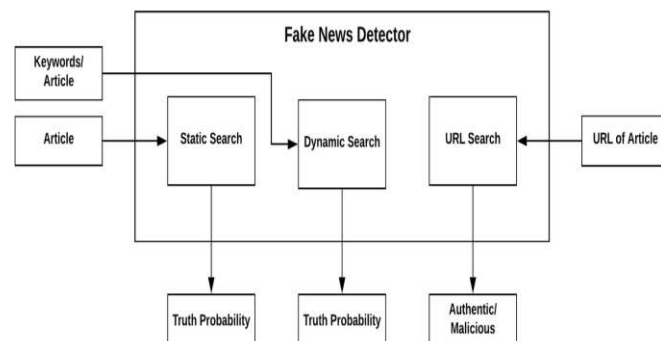


Figure 1: System Design

B. System Architecture-

i) Static Search-

The architecture of Static part of fake news detection system is quite simple and is done keeping in mind the basic machine learning process flow. The system design is shown below and self-explanatory. The main processes in the design are-

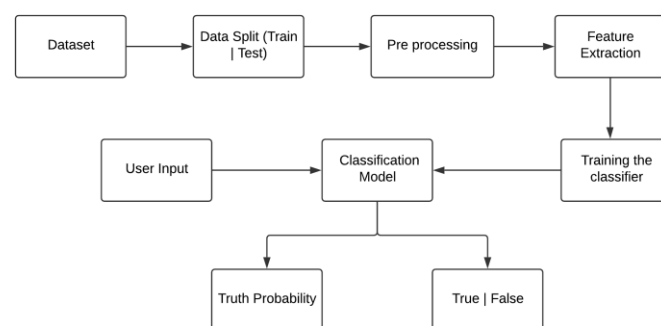


Figure 2: System Architecture

ii) Dynamic Search-

The second search field of the site asks for specific keywords to be searched on the net upon which it provides a suitable output for the percentage probability of that term actually being present in an article or a similar article with those keyword references in it.

iii) URL Search-

The third search field of the site accepts a specific website domain name upon which the implementation looks for the site in our true sites database or the blacklisted sites database. The true sites database holds the domain names which regularly provide proper and authentic news and vice versa. If the site isn't found in either of the databases then the implementation doesn't

classify the domain it simply states that the news aggregator does not exist.

IV. IMPLEMENTATION

4.1 DATA COLLECTION AND ANALYSIS

We can get online news from different sources like social media websites, search engine, homepage of news agency websites or the fact-checking websites. On the Internet, there are a few publicly available datasets for Fake news classification like BuzzFeed News, LIAR [15], BS Detector etc. These datasets have been widely used in different research papers for determining the veracity of news. In the following sections, I have discussed in brief about the sources of the dataset used in this work.

Online news can be collected from different sources, such as news agency homepages, search engines, and social media websites. However, manually determining the veracity of news is a challenging task, usually requiring annotators with domain expertise who performs careful analysis of claims and additional evidence, context, and reports from authoritative sources. Generally, news data with annotations can be gathered in the following ways: Expert journalists, Fact-checking websites, Industry detectors, and Crowd sourced workers. However, there are no agreed upon benchmark datasets for the fake news detection problem. Data gathered must be pre-processed- that is, cleaned, transformed and integrated before it can undergo training process [16]. The dataset that we used is explained below:

LIAR: This dataset is collected from fact-checking website PolitiFact through its API [15]. It includes 12,836 human labelled short statements, which are sampled from various contexts, such as news releases, TV or radio interviews, campaign speeches, etc. The labels for news truthfulness are fine-grained multiple classes: pants-fire, false, barely-true, half-true, mostly true, and true.

The data source used for this project is LIAR dataset which contains 3 files with .csv format for test, train and validation. Below is some description about the data files used for this project.

1. LIAR: A Benchmark Dataset for Fake News Detection

William Yang Wang, "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection, to appear in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), short paper, Vancouver, BC, Canada, July 30-August 4, ACL.

Below are the columns used to create 3 datasets that have been in used in this project-

- Column1: Statement (News headline or text).
- Column2: Label (Label class contains: True, False)

The dataset used for this project were in csv format named train.csv, test.csv and valid.csv.

2. REAL_OR_FAKE.CSV we used this dataset for passive aggressive classifier. It contains 3 columns viz 1- Text/keyword, 2-Statement, 3-Label (Fake/True)

4.2 DEFINITIONS AND DETAILS

A. Pre-processing Data

Social media data is highly unstructured – majority of them are informal communication with typos, slangs and bad-grammar etc. [17]. Quest for increased performance and reliability has made it imperative to develop techniques for utilization of resources to make informed decisions [18]. To achieve better insights, it is necessary to clean the data before it can be used for predictive modelling. For this purpose, basic pre-processing was done on the News training data. This step was comprised of-

Data Cleaning:

While reading data, we get data in the structured or unstructured format. A structured format has a well-defined pattern whereas unstructured data has no proper structure. In between the 2 structures, we have a semi-structured format which is a comparably better structured than unstructured format.

Cleaning up the text data is necessary to highlight attributes that we're going to want our machine learning system to pick up on. Cleaning (or pre-processing) the data typically consists of a number of steps:

a) Remove punctuation

Punctuation can provide grammatical context to a sentence which supports our understanding. But for our vectorizer which counts the number of words and not the context, it does not add value, so we remove all special characters. eg: How are you?->How are you

b) Tokenization

Tokenizing separates text into units such as sentences or words. It gives structure to previously unstructured text. eg: Plata o Plomo-> 'Plata','o','Plomo'.

c) Remove stopwords

Stopwords are common words that will likely appear in any text. They don't tell us much about our data so we remove them. eg: silver or lead is fine for me-> silver, lead, fine.

d) Stemming

Stemming helps reduce a word to its stem form. It often makes sense to treat related words in the same way. It removes suffixes, like “ing”, “ly”, “s”, etc. by a simple rule-based approach. It reduces the corpus of words but often the actual words get neglected. eg: Entitling, Entitled -> Entitle. Note: Some search engines treat words with the same stem as synonyms [18].

B. Feature Generation

We can use text data to generate a number of features like word count, frequency of large words, frequency of unique words, n-grams etc. By creating a representation of words that capture their meanings, semantic relationships, and numerous types of context they are used in, we can enable computer to understand text and perform Clustering, Classification etc [19].

Vectorizing Data:

Vectorizing is the process of encoding text as integers i.e. numeric form to create feature vectors so that machine learning algorithms can understand our data.

1. Vectorizing Data: Bag-Of-Words

Bag of Words (BoW) or CountVectorizer describes the presence of words within the text data. It gives a result of 1 if present in the sentence and 0 if not present. It, therefore, creates a bag of words with a document-matrix count in each text document.

2. Vectorizing Data: N-Grams

N-grams are simply all combinations of adjacent words or letters of length n that we can find in our source text. Ngrams with n=1 are called unigrams. Similarly, bigrams (n=2), trigrams (n=3) and so on can also be used. Unigrams usually don't contain much information as compared to bigrams and trigrams. The basic principle behind n-grams is that they capture the letter or word is likely to follow the given word. The longer the n-gram (higher n), the more context you have to work with [20].

3. Vectorizing Data: TF-IDF

It computes “relative frequency” that a word appears in a document compared to its frequency across all documents TF-IDF weight represents the relative importance of a term in the document and entire corpus [17].

TF stands for Term Frequency: It calculates how frequently a term appears in a document. Since, every document size varies, a term may appear more in a long sized document than a short one. Thus, the length of the document often divides Term frequency.

Note: Used for search engine scoring, text summarization, document clustering.

$$TF(t, d) = \frac{\text{Number of times } t \text{ occurs in document 'd'}}{\text{Total word count of document 'd'}}$$

IDF stands for Inverse Document Frequency: A word is not of much use if it is present in all the documents. Certain terms like “a”, “an”, “the”, “on”, “of” etc. appear many times in a document but are of little importance. IDF weighs down the importance of these terms and increase the importance of rare ones. The more the value of IDF, the more unique is the word [17].

$$IDF(t, d) = \frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}}$$

TF-IDF is applied on the body text, so the relative count of each word in the sentences is stored in the document matrix.

$$TFIDF(t, d) = TF(t, d) * IDF(t)$$

Note: Vectorizers outputs sparse matrices. Sparse Matrix is a matrix in which most entries are 0 [21].

C. Algorithms used for Classification

This section deals with training the classifier. Different classifiers were investigated to predict the class of the text. We explored specifically four different machine-learning algorithms – Multinomial Naïve Bayes Passive Aggressive Classifier and Logistic regression.

The implementations of these classifiers were done using Python library Sci-Kit Learn.

Brief introduction to the algorithms-

1. Naïve Bayes Classifier:

This classification technique is based on Bayes theorem, which assumes that the presence of a particular feature in a class is independent of the presence of any other feature. It provides way for calculating the posterior probability.

$$P(x) = \frac{P(c) * P(c)}{P(x)}$$

$P(c|x)$ = posterior probability of class given predictor

$P(c)$ = prior probability of class

$P(x|c)$ = likelihood (probability of predictor given class)

$P(x)$ = prior probability of predictor

2. Random Forest:

Random Forest is a trademark term for an ensemble of decision trees. In Random Forest, we've collection of decision trees (so known as "Forest"). To classify a new object based on attributes, each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The reason that the random forest model works so well is: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. So how does random forest ensure that the behaviour of each individual tree is not too correlated with the behaviour of any of the other trees in the model? It uses the following two methods:

2.1 Bagging (Bootstrap Aggregation) — Decisions trees are very sensitive to the data they are trained on — small changes to the training set can result in significantly different tree structures. Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees. This process is known as bagging or bootstrapping.

2.2 Feature Randomness — In a normal decision tree, when it is time to split a node, we consider every possible feature and pick the one that produces the most separation between the observations in the left node vs. those in the right node. In contrast, each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification [22].

3. Logistic Regression:

It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as logit regression. Since, it predicts the probability, its output values lies between 0 and 1 (as expected).

Mathematically, the log odds of the outcome are modelled as a linear combination of the predictor variables [23].

Odds = $p/(1-p)$ = probability of event occurrence / probability of not event occurrence

$\ln(\text{odds}) = \ln(p/(1-p))$

$\text{logit}(p) = \ln(p/(1-p)) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_kX_k$

4. Passive Aggressive Classifier:

The Passive Aggressive Algorithm is an online algorithm; ideal for classifying massive streams of data (e.g. twitter). It is easy to implement and very fast. It works by taking an example, learning from it and then throwing it away [24]. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector [25].

4.3 IMPLEMENTATION STEPS

A. Static Search Implementation-

In static part, we have trained and used 3 out of 4 algorithms for classification. They are Naïve Bayes, Random Forest and Logistic Regression.

Step 1: In first step, we have extracted features from the already pre-processed dataset. These features are; Bag-of-words, Tf-Idf Features and N-grams.

Step 2: Here, we have built all the classifiers for predicting the fake news detection. The extracted features are fed into different classifiers. We have used Naive-bayes, Logistic Regression, and Random forest classifiers from sklearn. Each of the extracted features was used in all of the classifiers.

Step 3: Once fitting the model, we compared the f1 score and checked the confusion matrix.

Step 4: After fitting all the classifiers, 2 best performing models were selected as candidate models for fake news classification.

Step 5: We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier.

Step 6: Finally selected model was used for fake news detection with the probability of truth.

Step 7: Our finally selected and best performing classifier was Logistic Regression which was then saved on disk. It will be used to classify the fake news.

It takes a news article as input from user then model is used for final classification output that is shown to user along with probability of truth.

B. Dynamic Search Implementation-

Our dynamic implementation contains 3 search fields which are-

- 1) Search by article content.
- 2) Search using key terms.
- 3) Search for website in database.

In the first search field we have used Natural Language Processing for the first search field to come up with a proper solution for the problem, and hence we have attempted to create a model which can classify fake news according to the terms used in the newspaper articles. Our application uses NLP techniques like CountVectorization and TF-IDF Vectorization before passing it through a Passive Aggressive Classifier to output the authenticity as a percentage probability of an article.

The second search field of the site asks for specific keywords to be searched on the net upon which it provides a suitable output for the percentage probability of that term actually being present in an article or a similar article with those keyword references in it.

The third search field of the site accepts a specific website domain name upon which the implementation looks for the site in our true sites database or the blacklisted sites database. The true sites database holds the domain names which regularly provide proper and authentic news and vice versa. If the site isn't found in either of the databases then the implementation doesn't classify the domain it simply states that the news aggregator does not exist.

Working-

The problem can be broken down into 3 statements-

- 1) Use NLP to check the authenticity of a news article.
- 2) If the user has a query about the authenticity of a search query then we he/she can directly search on our platform and using our custom algorithm we output a confidence score.
- 3) Check the authenticity of a news source.

These sections have been produced as search fields to take inputs in 3 different forms in our implementation of the problem statement.

4.4 EVALUATION MATRICES

Evaluate the performance of algorithms for fake news detection problem; various evaluation metrics have

been used. In this subsection, we review the most widely used metrics for fake news detection. Most existing approaches consider the fake news problem as a classification problem that predicts whether a news article is fake or not:

True Positive (TP): when predicted fake news pieces are actually classified as fake news;

True Negative (TN): when predicted true news pieces are actually classified as true news;

False Negative (FN): when predicted true news pieces are actually classified as fake news;

False Positive (FP): when predicted fake news pieces are actually classified as true news.

Confusion Matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made [26].

Table 1: Confusion Matrix

Total	Class 1 (Predicted)	Class 2 (Predicted)
Class 1 (Actual)	TP	FN
Class 2 (Actual)	FP	TN

By formulating this as a classification problem, we can define following metrics-

$$1. \text{ Precision} = \frac{|T P|}{|T P| + |F P|}$$

$$2. \text{ Recall} = \frac{|T P|}{|T P| + |F N|}$$

$$3. \text{ F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$4. \text{ Accuracy} = \frac{|T P| + |T N|}{|T P| + |T N| + |F P| + |F N|}$$

These metrics are commonly used in the machine learning community and enable us to evaluate the performance of a classifier from different perspectives. Specifically, accuracy measures the similarity between predicted fake news and real fake news.

4.5 SNAPSHOTS OF SYSTEM WORKING

A. Static System-

```
UserWarning)
The given statement is True
The truth probability score is 0.6202405257600963
(base) C:\Users\HP\Desktop\fake news detection\Fake_News_Detection>
```

Figure 3: Static output (True)

```
The given statement is False
The truth probability score is 0.3221557972557687
(base) C:\Users\HP\Desktop\fake news detection\Fake_News_Detection>
```

Figure 4: Static Output (False)

B. Dynamic System-

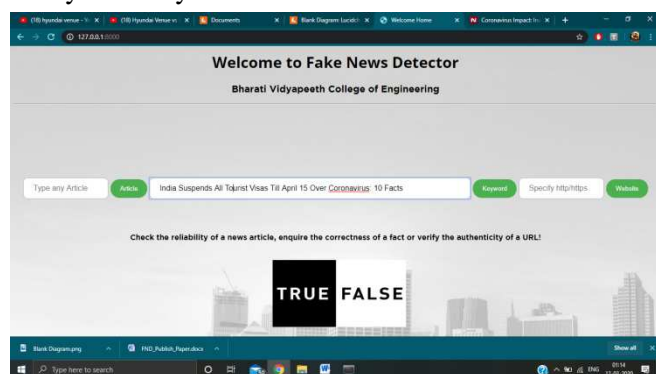


Figure 5: Fake News Detector (Home Screen)

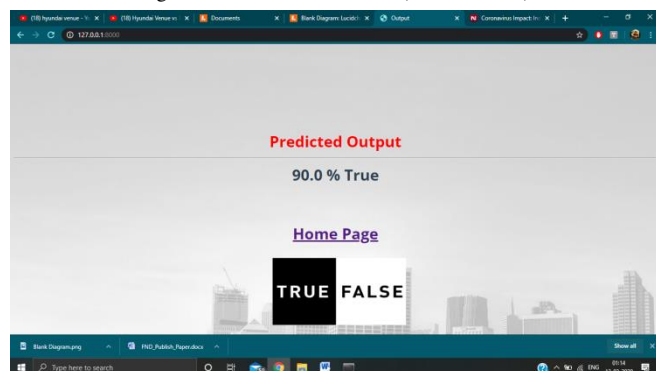


Figure 6: Fake News Detector (Output page)

V. RESULTS

Implementation was done using the above algorithms with Vector features- Count Vectors and Tf-Idf vectors at Word level and Ngram-level. Accuracy was noted for all models. We used K-fold cross validation technique to improve the effectiveness of the models.

A. Dataset split using K-fold cross validation

This cross-validation technique was used for splitting the dataset randomly into k-folds. (k-1) folds were used for building the model while kth fold was used to check the effectiveness of the model. This was repeated until each of the k-folds served as the test set. I used 3-fold cross validation for this experiment where 67% of the data is used for training the model and remaining 33% for testing.

B. Confusion Matrices for Static System

After applying various extracted features (Bag-of-words, Tf-Idf, N-grams) on three different classifiers (Naïve bayes, Logistic Regression and Random Forest), their confusion matrix showing actual set and predicted sets are mentioned below:

Table 2: Confusion Matrix for Naïve Bayes Classifier using Tf-Idf features-

Total= 10240	Naïve Bayes Classifier	
	<i>Fake (Predicted)</i>	<i>True (Predicted)</i>
Fake (Actual)	841	3647
True (Actual)	427	5325

Table 3: Confusion Matrix for Logistic Regression using Tf-Idf features-

Total= 10240	Logistic Regression	
	<i>Fake (Predicted)</i>	<i>True (Predicted)</i>
Fake (Actual)	1617	2871
True (Actual)	1097	4655

Table 4: Confusion Matrix for Random Forest Classifier using Tf-Idf features-

Total= 10240	Random Forest	
	<i>Fake (Predicted)</i>	<i>True (Predicted)</i>
Fake (Actual)	1979	2509
True (Actual)	1630	4122

Table 5: Comparison of Precision, Recall, F1-scores and Accuracy for all three classifiers-

Classifiers	Precision	Recall	F1-Score	Accuracy
Naïve Bayes	0.59	0.92	0.72	0.60
Random Forest	0.62	0.71	0.67	0.59
Logistic Regression	0.69	0.83	0.75	0.65

As evident above our best model came out to be Logistic Regression with an accuracy of 65%. Hence we then used grid search parameter optimization to increase the performance of logistic regression which then gave us the accuracy of 80%.

Hence we can say that if a user feed a particular news article or its headline in our model, there are 80% chances that it will be classified to its true nature.

C. Confusion Matrix for Dynamic System

We used `real_or_fake.csv` with passive aggressive classifier and obtained the following confusion matrix-

Table 6: Confusion Matrix for passive aggressive classifier-

Total= 1267	Passive Aggressive Classifier	
	<i>Fake (Predicted)</i>	<i>True (Predicted)</i>
Fake (Actual)	588	50
True (Actual)	42	587

Table 7: Performance measures-

Classifier	Precision	Recall	F1-Score	Accuracy
PAC	0.93	0.9216	0.9257	0.9273

VI. CONCLUSION

In the 21st century, the majority of the tasks are done online. Newspapers that were earlier preferred as hard-copies are now being substituted by applications like Facebook, Twitter, and news articles to be read online. Whatsapp's forwards are also a major source. The growing problem of fake news only makes things more complicated and tries to change or hamper the opinion and attitude of people towards use of digital technology. When a person is deceived by the real news two possible things happen- People start believing that their perceptions about a particular topic are true as assumed. Thus, in order to curb the phenomenon, we have developed our Fake news Detection system that takes input from the user and classify it to be true or fake. To implement this, various NLP and Machine Learning Techniques have to be used. The model is trained using an appropriate dataset and performance evaluation is also done using various performance measures. The best model, i.e. the model with highest accuracy is used to classify the news headlines or articles. As evident above for static

search, our best model came out to be Logistic Regression with an accuracy of 65%. Hence we then used grid search parameter optimization to increase the performance of logistic regression which then gave us the accuracy of 75%. Hence we can say that if a user feed a particular news article or its headline in our model, there are 75% chances that it will be classified to its true nature.

The user can check the news article or keywords online; he can also check the authenticity of the website. The accuracy for dynamic system is 93% and it increases with every iteration.

We intent to build our own dataset which will be kept up to date according to the latest news. All the live news and latest data will be kept in a database using Web Crawler and online database.

VII. REFERENCES

- [1] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, "Fake News Detection on Social Media: A Data Mining Perspective" arXiv:1708.01967v3 [cs.SI], 3 Sep 2017
- [2] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, "Fake News Detection on Social Media: A Data Mining Perspective" arXiv:1708.01967v3 [cs.SI], 3 Sep 2017
- [3] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, 2017, pp. 900-903.
- [4] Fake news websites. (n.d.) Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Fake_news_website. Accessed Feb. 6, 2017
- [5] Cade Metz. (2016, Dec. 16). The bittersweet sweepstakes to build an AI that destroys fake news.
- [6] Conroy, N., Rubin, V. and Chen, Y. (2015). "Automatic deception detection: Methods for finding fake news" at Proceedings of the Association for Information Science and Technology, 52(1), pp.1-4.
- [7] Markines, B., Cattuto, C., & Menczer, F. (2009, April). "Social spam detection". In Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (pp. 41-48)
- [8] Rada Mihalcea, Carlo Strapparava, The lie detector: explorations in the automatic recognition of deceptive language, Proceedings of the ACL-IJCNLP
- [9] Kushal Agarwalla, Shubham Nandan, Varun Anil Nair, D. Deva Hema, "Fake News Detection using Machine Learning and Natural Language Processing," International Journal of Recent Technology and

Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6, March 2019

[10] H. Gupta, M. S. Jamal, S. Madisetty and M. S. Desarkar, "A framework for real-time spam detection in Twitter," 2018 10th International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, 2018, pp. 380-383

[11] M. L. Della Vedova, E. Tacchini, S. Moret, G. Ballarin, M. DiPierro and L. de Alfaro, "Automatic Online Fake News Detection Combining Content and Social Signals," 2018 22nd Conference of Open Innovations Association (FRUCT), Jyväskylä, 2018, pp. 272-279.

[12] C. Buntain and J. Golbeck, "Automatically Identifying Fake News in Popular Twitter Threads," 2017 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, 2017, pp. 208-215.

[13] S. B. Parikh and P. K. Atrey, "Media-Rich Fake News Detection: A Survey," 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, 2018, pp. 436-441

[14] Scikit-Learn- Machine Learning In Python

[15] Dataset- Fake News detection William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. arXiv preprint arXiv:1705.00648, 2017.

[16] Shankar M. Patil, Dr. Praveen Kumar, "Data mining model for effective data analysis of higher education students using MapReduce" IJERMT, April 2017 (Volume-6, Issue-4).

[17] Aayush Ranjan, "Fake News Detection Using Machine Learning", Department Of Computer Science & Engineering Delhi Technological University, July 2018.

[18] Patil S.M., Malik A.K. (2019) Correlation Based Real-Time Data Analysis of Graduate Students Behaviour. In: Santosh K., Hegadi R. (eds) Recent Trends in Image Processing and Pattern Recognition. RTIP2R 2018. Communications in Computer and Information Science, vol 1037. Springer, Singapore.

[19] Badreesh Shetty, "Natural Language Processing (NLP) for machine learning" at towardsdatascience, Medium.

[20] NLTK 3.5b1 documentation, Nltk generate n gram

[21] Ultimate guide to deal with Text Data (using Python) – for Data Scientists and Engineers by Shubham Jain, February 27, 2018

[22] Understanding the random forest by Anirudh Palaparthi, Jan 28, at analytics vidya.

[23] Understanding the random forest by Anirudh Palaparthi, Jan 28, at analytics vidya.

[24] Shailesh-Dhama, "Detecting-Fake-News-with-Python", Github, 2019

[25] Aayush Ranjan, "Fake News Detection Using Machine Learning", Department Of Computer Science & Engineering Delhi Technological University, July 2018.

[26] What is a Confusion Matrix in Machine Learning by Jason Brownlee on November 18, 2016 in Code Algorithms From Scratch