International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

FAKE NEWS DETECTION

ELECTRONICS AND TELECOMMUNICATION Department, Savitribai Phule Pune University, Pune

Dr. S.A. Bhisikar
JSPM's Rajarshi Shahu College of Engineering

Aishwarya Patil
JSPM's Rajarshi Shahu College of Engineering

Vaishnavi Kulkarni
JSPM's Rajarshi Shahu College of Engineering

Shweta Pawar
JSPM's Rajarshi Shahu College of Engineering

***Abstract*—** The fake news on social media is spreading extensively and is a matter of major concern due to its ability to cause a lot of social and national damage. A lot of research is already focused on identifying it. Analysis of research related to fake news detection is done in this paper and traditional methods have been explored. It chooses the best existing model, in order to create a new model with supervised machine learning algorithm, that can classify fake news as true or false, by using tools like python scikit-learn, NLP for textual analysis. NLP helps in resolving uncertainty in language and includes useful numeric structure to the data for various downstream applications. This method will result in feature extraction and vectorization; which also proposes the use of Python scikit-learn library to perform tokenization and feature extraction of text data, since this library contains useful tools, such as Count Vectorizer and Tiff Vectorizer. Then, feature selection methods were selected to experiment and choose the best fit features to obtain the highest precision.

***Keywords*— Natural Language Processing, machine learning, KNN, SVM, Random Forest**

I. INTRODUCTION

Some of the news articles may contain misleading information that needs to be checked. These news articles may contain false information about a particular statistic in a country or overestimated cost of certain services for a country, which may cause unrest for some countries. However, their scope is so limited because they depend on manual detection. In a globe with millions of articles being removed or being published every minute, this process cannot be feasible. A solution could be developing a system that provides a credible automated index scoring, or rating for credibility of different publishers, and news context. This paper suggests a methodology to generate a model that will find out if an article is real or fake based on its words, phrases, sources and titles, by using supervised machine learning algorithms on a labelled dataset, that are manually classified. Moreover, feature selection methods are experimented to choose the best fit attributes to achieve the highest precision, according to confusion matrix results. This paper puts forward the creation of a model using several classification algorithms. The product model will test the unseen data, the results will be plotted, and accordingly, the product will be a model that detects and classifies fake articles and can be used and integrated with any system for future use.

This paper presents a novel method for detecting fake news that uses:

- **Pre-processing:** process of removing null values or garbage values from the dataset.
- **Feature Extraction:** all sorts of symbolic data are removed from the data in this step.
- **Random Forest Classification:** a supervised machine learning algorithm which allows the classification of new information.

II. LITERATURE SURVEY

Authors in [1] have used a method for automating fake news detection on Twitter by learning to predict accuracy assessments in two credibility focused Twitter datasets: CREDBANK, a crowd sourced dataset of accuracy assessments for events in Twitter, and PHEME, a dataset of possible rumours in Twitter and journalistic assessments of their accuracies. A feature analysis identifies features that are most diagnostic for crowd sourced and journalistic accuracy assessments, results of which are consistent with previous work or study.

Authors of [2] presented LIAR, a new dataset for automatic fake news detection. This corpus can also be used for stance classification, argument mining, topic modelling, rumour detection, and political NLP research. However, it is well-known that this last is restricted to political information, whereas others have combined information from different fields.

Marco L. Della Vedova et. al proposed in [3] a novel Machine Learning based fake news detection method by combining social context features and news content, outperforms existing techniques in the literature, Furthermore, they implemented their method within a Facebook Messenger Chatbot and acknowledged it with a real-world application. They first distinguished the datasets they used for their test, then presented the content-based point of view they applied and the method they proposed to merge it with a social-based approach available in the literature.

In article [4], the author introduced the concept of the importance of NLP in stumbling across incorrect information. They have made use of time frequency-inverse document frequency (TF-IDF) of bigrams and probabilistic context-free grammar detection. They inspected the data set in more than one class of algorithms to discover a better model.

Mykhailo Granik et. al. in paper [5] exhibit a simple approach for fake news detection by applying Naive Bayes classifier. This perspective was executed as a software system and tested against a data set of Facebook news posts. They were gathered from three large Facebook pages, as well as three large mainstream political news. This may be caused by the skewness of the dataset: only 4.9% of it is fake news.

Authors of [6] propose a typology of several methods of truth assessment emerging from two main categories: linguistic cue approaches with machine learning and network analysis approaches, for detecting fake news.

In paper [7], the authors detect fake news on social media, a data mining perspective is presented that includes the characterization of fake news in psychology and social theories. It proposes a general two-phase data mining framework that includes 1) feature extraction and 2) analysing and modelling data sets, and confusion matrix for detecting fake news.

In paper [8], Shivam B. Parikh et. al. aims to present an understanding of characterization of news story in the modern diaspora merged with the differential content types of news story and its influence on readers. Afterwards, we dive into existing fake news detection methods that are entirely based on text- based analysis, and also report popular fake news datasets. It is a theoretical

proposal which gives illustrations of fake news detection by studying the psychological factors.

Authors in article [9] published a paper in the year 2018 in Bhopal, India Information preciseness on Internet, especially on social media, is an increasingly important concern. In this paper, authors put forward a system for "fake news" detection and different methods to apply it on Facebook. This technique uses Naive Bayes classification model.

In [10], the authors present an overall performance analysis of different approaches on three different datasets. This work focused on the text of the information and the feeling given by it, and ignores some features like the source, the author or the date of the publication that can have a considerable effect on the result. Besides, in our work, they showed that the integration of the feeling in the detection process does not bring any valuable information.

III. OBJECTIVES

The main objective is to generate a model that can discriminate between "fake" and "true" articles when it is trained with a LIAR fake news dataset. We have used three datasets: Test, Train and Valid.

IV. METHODOLOGY

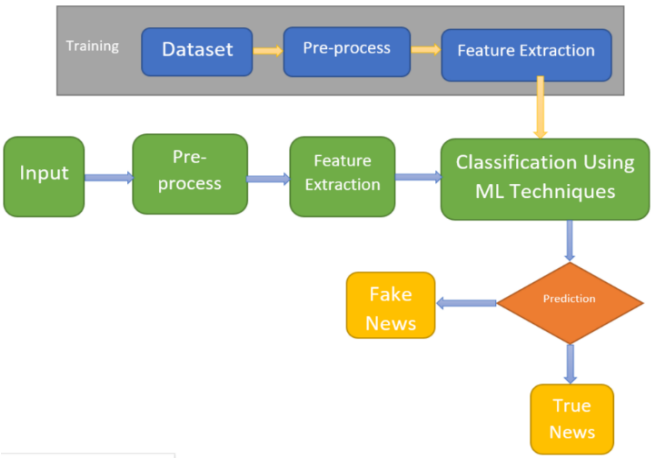
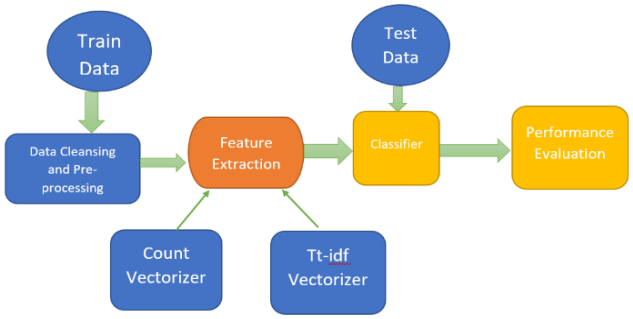


Fig: - System Architecture

The proposed fake news detection model is based on machine learning is shown in Figure above. Firstly, the news articles are pre-processed. A binary label is assigned to each news article as 1 for fake news and 0 for real news. The input news articles are changed to UTF-8 format and stop words and punctuations are taken off. The titles of news articles and content text are converted to space-separated padded sequences of words. These sequences are further divided into lists of tokens. Global Vectors for Word Representation (GloVe) embeddings is issued by Stanford NLP team. It is a supervised learning algorithm for obtaining vector representations for words. Pre-trained GloVe word embeddings are used to refer to the high dimensional news articles. The embedding layer will load the weights from GloVe instead of loading random weights. GloVe requests globally collected co-occurrence statistics across all words in the news article corpus. The resulting representations formalize important linear substructures of the word vector space. The converted vector represented data is partitioned into train, test and validation data. The training is taken place on the news article corpus. Validation data set

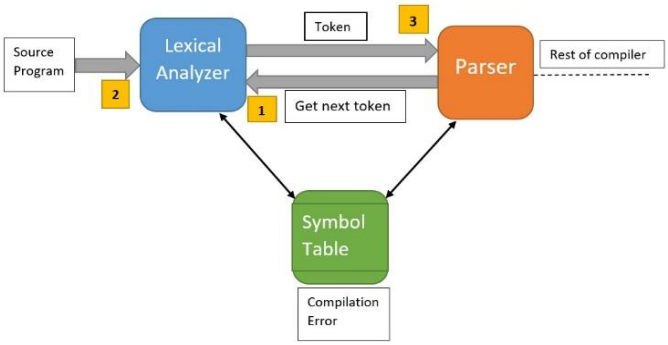
is utilized for fine-tuning the model. Further, the test data is implemented to know the stated label of news article based on trained model.



Data Acquisition: We used several datasets with various features to evaluate our proposed method: LIAR: This dataset was gathered by a website PolitiFact reality using its API. It includes 12836 brief statements with labels that were collected from different sources such as published news, TV and radio interviews, election speeches and, etc. These samples are classified as real, mostly real, semi-real and wrong classes.

Pre-processing: Furthermore, we will apply various pre-processing steps such as lexical analysis, stop word removal, stemming (Porter's algorithm), index term selection and data cleaning in order to make our dataset proper.

Lexical analysis: Lexical analysis segregates the input letters into, 1) Word characters (e.g., the alphabets a-z and 2) Word separators (e.g., space, newline, tab).
The main role of lexical analysis is to study the input attributes in the code and produce tokens. Lexical analyzer inspects the entire source code of the program. It points out each token gradually. Scanners are normally executed to produce tokens only when called for by a parser. Here is how identification of tokens in compiler design works-



Feature Extraction: Extracting the genuine feature has an ideal effect on the performance of the machine learning classifier algorithms. Count Vectorizer and TF-IDF Vectorizer (term frequency inverse document frequency) are used to withdraw features from our text before transferring it into the classification algorithms. Count Vectorizer produces a vector which has as various proportions as the specific word of corpora (collections of written texts). Every single word has a particular dimension and carries 1 in that specified dimension with 0 in others which simply retains the frequency of every word. TF-IDF vectorizer gives numerical depiction of the words whether they are present or not, instead of simply featuring a count. Words are measured by frequency, multiplied by the reverse document frequency of them. In simple terms, words that appear a good amount but everywhere should be provided hardly any significance or weighting don't offer a huge deal of interest. If a word occurs a smaller number of times, in that case such words are more relevant and should be carefully weighed. This would help in upgraded performance on classification. It is a technique intended to report the importance of a keyword inside a text. If a particular document is denoted as 'd', a term as 't' and the whole document as 'D', then the formula is given as,

$$tf - idf(t, d, D) = tf(t, d) * idf(t, D)$$

Here,
tf(t, d) is the frequency of 't' in 'd' and idf(t, D) is how 't' is common or rare across 'D'.

Stop word removal: Stop word removal deals with the removal of words that occur repeatedly in documents. Stop word removal is oftentimes used for pre-processing steps across distinct NLP applications. The idea is to simply separate the words that arise commonly across all the documents in the corpus. Stop words are usually considered to be a "single set of words". It really can have different meanings to different applications. For example, in some applications, removal of all stop words right from determiners (e.g., the, a, an) to prepositions (e.g., above, across, before) to some adjectives (e.g., good, nice) can be a suitable stop word list. To some applications nonetheless, this step can be detrimental. For instance, in sentiment analysis discarding adjective terms like 'good' and 'nice' as well as negations like 'not' can throw algorithms off their tracks. In such cases, a user can choose to use a minimal stop list that includes only determiners or determiners with prepositions or just coordinating conjunctions depending on the requirements of the application.

Stemming: Stemming substitutes all the variants of a word with a single stem word. Variants comprise of plurals, gerund forms (ing forms), third person suffixes, past tense suffixes, etc.). Stemming is a method used to withdraw the base form of the words by extracting affixes from them. It is exactly like cutting down the branches of a tree to its stems. For example, the stem of the words ***writing, writes, written*** is ***write***. Search engines make use of stemming for indexing the words. As a result, rather than saving all forms of a word, a search engine can store only the stems. In this manner, stemming decreases the size of the index and increases retrieval accuracy.

Train Classification: We compile artificial as well as real time using online review data and provide training with any machine learning classifier. Training of the classifiers of machine learning is a crucial task. This plays a vital role for the accuracy of results of these classifiers. First, unnecessary words have been removed and the words are transformed to its single form. So that the training dataset that is given to these classifiers should only have the valuable data.

Test Classification: we detect the fake review using any machine learning classifier, weight calculator for real time or synthetic input data accordingly. We evaluate the performance of algorithms for fake news detection problem; various evaluation metrics have been used. In this subsection, the most widely used metrics for fake news detection have been reviewed. Most existing researches assess the fake news problem as a classification problem that anticipates whether a news article is real or fake.

Barely True: when predicted fake news pieces are actually classified as fake news;

Mostly True: when predicted true news pieces are actually classified as true news;

Pants fire: when predicted true news pieces are actually classified as fake news;

Half True: when predicted fake news pieces are actually classified as fake news.

```
Label1= []

Statement= []

for i in traindata.label:

if i=='false':

    la='false'

  if i=='half-true':

    la='false'

  if i=='barely-true':

    la='false'

  if i=='mostly-true':
```

```
    la='true'
if i=='pants-fire':
    la='false'
if i=='true':
    la='true'
if i=='original':
    la='true'
Label1.append(la)
```

Analysis: We demonstrate the accuracy of preferred system and analyse with other existing systems.

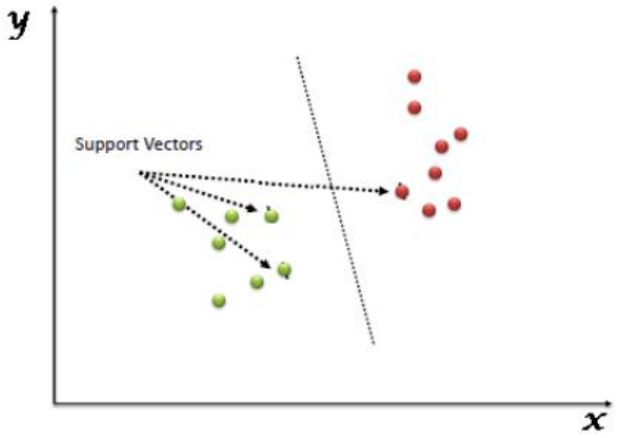
V. RESULT

The scope of this project is to cover the fake news data, of a dataset known as Liar-dataset, it is a New Benchmark Dataset for Fake News Detection and labelled by fake or trust news. We have performed analysis on "Liar" dataset. The results of the analysis of the datasets using the three algorithms have been depicted using the confusion matrix.
The three algorithms used for the detection are as:

- Support Vector Machine (SVM)
- Random Forests.
- K-Nearest Neighbours (KNN).

Support Vector Machine (SVM) The SVM algorithm is based on the layout of each data item in the form of a point in a range of dimensions n (the number of available properties), and the value of a given property is the number of specified coordinates. Given a set of n features, SVM algorithm uses n dimensional space to plot the data item with the coordinates representing the value of each feature. The hyper-plane obtained to separate the two classes is used for classifying the data.

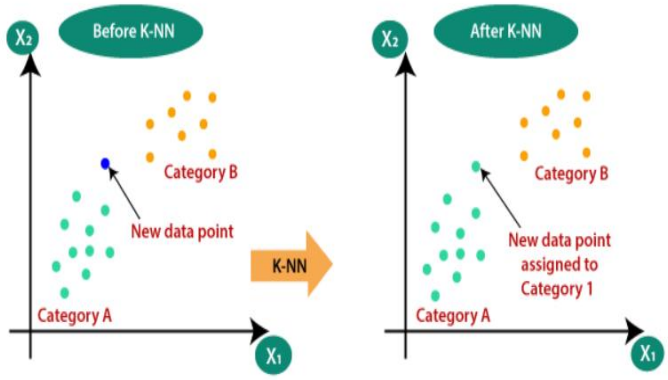
```
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
clf=
make_pipeline(StandardScaler(with_mean=False),
SVC())
clf.fit(X_train, y_train)
```



	precision	recall	F1-score	support
false	0.98	0.68	0.80	1244
true	0.04	0.50	0.08	36
accuracy			0.67	1280
Macro average	0.51	0.59	0.44	1280
Weighted average	0.96	0.67	0.78	1280

KNN classifies new positions based on most of the sounds from the neighbouring k with respect to them. The position assigned in the class is highly mutually exclusive between the nearest neighbours K, as measured by the role of the distance KNN falls in the category of supervised learning and its main applications are intrusion detection, pattern recognition.

```
clf = KNeighborsClassifier(n_neighbors=5)
```



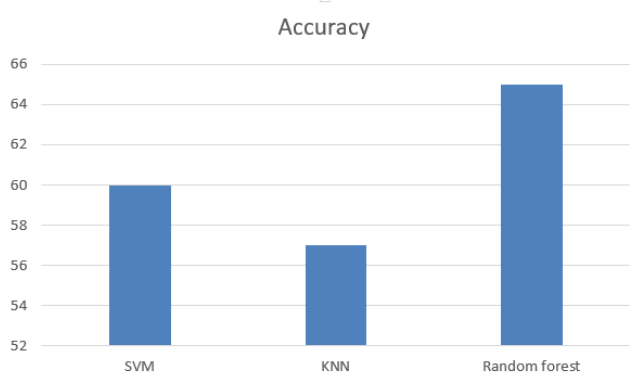
	precision	recall	F1-score	support
false	0.86	0.71	0.78	1053
true	0.26	0.47	0.33	227
accuracy			0.66	1280
Macro average	0.56	0.59	0.55	1280
Weighted average	0.75	0.66	0.70	1280

Random Forest are built on the concept of building many decision tree algorithms, after which the decision trees get a separate result. The results, that are estimated by a huge number of decision trees, are taken up by the random forest. To ensure a variation of the decision trees, the random forest randomly selects a subcategory of properties from each group The applicability of Random Forest is best when used on uncorrelated decision trees. If applied on similar trees, the overall result will be more or less similar to a single decision tree. Uncorrelated decision trees can be obtained by bootstrapping and feature randomness.

clf = RandomForestClassifier(n_estimators = 100)

	precision	recall	F1-score	support
false	0.95	0.69	0.80	1191
true	0.12	0.56	0.20	89
accuracy			0.68	1280
Macro average	0.54	0.63	0.50	1280
Weighted average	0.90	0.68	0.76	1280

Algorithm	Accuracy (%)
SVM	67
KNN	66
Random forest	68



VI. CONCLUSION

In this paper we present a system that is able to detect whether the news is fake or not. We have performed machine learning algorithms of fake news detection. We have addressed the benefits and shortcomings of the proposed model. We overcame the shortcomings of these models through our project. We made use of two datasets, a full training dataset and a testing training data set, associated with news articles. These datasets are obtained from Liar. Kaggle helps users to join with other users, find and produce datasets, use GPU integrated notebooks, and participate with other data scientists to resolve data science challenges. We have used natural language

processing to detect fake news directly, on the basis of the text content of the articles.

This paper presents a method of detecting fake news using random forest classifier, trying to determine the best features and techniques to detect fake news. We started by studying the field of fake news, its impact and its detection methods. We then designed and implemented a solution that uses a dataset of news pre- processed using cleaning techniques, steaming, bag of words and TF-IDF to extract a set of features allowing to detect fake news. We the applied Random Forest Classifier on our features dataset to build a model allowing the classification of the new information. Through the research carried out during this study, we obtained the following results:

• the best features to detect fake news are in order: text, author, source, date and sentiment.

• the analysis of the sentiment given by the text is interesting, however it would be more influential in the case of opinion mining.

• the random forest classifier seems the best algorithm to detect fake news, because it gave a better recognition rate, and allowed to give for each information a degree of confidence for its classification.

The work we have done could be completed and continued in different aspects. It would be relevant to extend this study with a larger dataset, and to evolve its supervised learning by another online for a continuous update and automatic integration of new fake news.

VII. ACKNOWLEDGMENT

We are greatly indebted to our guide Dr. Swati Bhisikar, Head of the Department and Principal for their unconditional support, and for sharing their profound technical knowledge, without which our work would not have seen the light of the day.

VIII. REFERENCES

[1] Cody Buntain and Jennifer Golbeck “Automatically identifying Fake News in popular Twitter Threads: 2017 IEEE International Conference on Smart Cloud (SmartCloud).

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

[3] 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), Jyvaskyla, 2018, pp. 272-279.

[4] S. Gilda, "Notice of Violation of IEEE Publication Principles: Evaluating machine learning algorithms for fake news detection," 2017 IEEE 15th Student Conference on Research and Development (SCOReD), 2017, pp. 110-115, DOI: 10.1109/SCORED.2017.8305411.

[5] Mykhailo Granik and Volodymyr Mesyura. Fake news detection using naive bayes classifier. In 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), pages 900–903. IEEE, 2017.

[6] Niall J Conroy, Victoria L Rubin, and Yimin Chen. Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology, 52(1):1–4, 2015.

[7] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (n.d.)."fake news detection on social media: A data Mining Perspective".

[8] 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

[9] Fake News Detection by Akshay Jain, Amey Kasbe, Maulana Azad National Institute of Technology, Bhopal, India, IEEE 2018 International Students' Conference on Electrical, Electronics, and Computer Sciences

[10] Junaed Younus Khan, Md Khondaker, Tawkat Islam, Anindya Iqbal, and Sadia Afroz. A benchmark study on machine learning methods for fake news detection. arXiv preprint arXiv:1905.04749, 2019.
[11] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detection of online fake news using n-gram analysis and machine learning techniques. In Inter-national Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, pages 127–138. Springer, 2017.
[12] Chih-Chung Chang and Chih-Jen Lin. LIBSVM – A Library for Support Vector Machines, July 15, 2018.
[13] Chris Faloutsos. Access methods for text. ACM Computing Surveys (CSUR), 17(1):49–74, 1985.
[14]Javatpoint_k Nearest Neighbor Algorithm for Machine Learning
[15] Educba SVM Algorithm