International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 03, 2020
ISSN: 1475-7192

Analysis of Machine Learning Algorithms via Detection of Fake News

[1]G.Manoj Kumar, [2]Chinmay Misra, [3]Achint Singh Rawat

Abstract-*In the modern political climate, fake news is a growing and legitimate threat to our institutions and all voters. Fake news articles are those which are "intentionally and verifiably false". This project is aimed at implementing combinations of various feature extraction techniques along with various Machine Learning algorithms from distinct categories and for the purpose of detecting fake news articles through their content. The results of this supervised binary text- classification problem will be compared and ranked. Kaggle which is owned by Google LLC and is a community of data scientists and machine learning engineers which will fulfill our requirement of a reliable source that provides us with a verifiable dataset of real and fake news. To mirror the real-world environment, the quantity of fake news articles in the dataset, will be substantially less than the amount of real news articles. A data set with approximately 85: 15 ratio will be used.*

Keywords: *Fake, News, Classifiers, Vectorizers, N-gram s, F1-score, Precision, Recall, Faux*

I. INTRODUCTION

The label of 'Fake News' firstly requires a reputed source which can verify and assign it. Datasets containing thousands of Fake and Real news items will be obtained in CSV format as it is well suited for the operations of Machine Learning Pandas library. The problem is a binary text classification problem. The classification labels 'Fake' and 'Real' will be based on the ontext text itself. Comparing scores from different categories of classifiers will provide a better insight on how variation works therefore for the purpose of this project classifiers from the same category will be not be executed or compared. These scores will be tabulated to provide for easier reading comprehension and analysis. Fake news has significantly different features from 'Sentiment analysis' or 'Spam detection' which are the more common text classification. Therefore, the expectation is that the scores of different classifier- vectorizer permutations will not exhibit traits similar to those in other binary text classification problems. Ultimately, classifiers can be ranked based on their performance in partnership with the corresponding vectorizer variations. Future enhancements will then be considered.

II. STATE OF THE ART

Paper1: This paper was intended by the rise in deceptive info in everyday media retailers and social

[1] Assistant Professor, Department of Computer Science and Engineering, Faculty of Engineering and Technology, SRMIST, Kattankulathur, Tamil Nadu India
[2] Department of Computer Science and Engineering, Faculty of Engineering and Technology, SRMIST,Kattankulathur, Tamil Nadu, India
[3] Department of Computer Science and Engineering, Faculty of Engineering and Technology, SRMIST,Kattankulathur, Tamil Nadu, India
Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

media feeds, news blogs, and e-papers. These have created it difficult to spot trustworthy news sources, therefore increasing the requirement for tools which might give insights into the dependability of consumed content. This paper centred on the automated identification of faux news. The contribution of this paper is twofold. First, it introduced 2 datasets in acceptable format for the specified goal of sleuthing pretend news. It covered seven totally different news domains.

Paper2: This paper talks regarding how the difficulty of the dependability of data on the web has emerged as a vital issue of contemporary society. Social networking websites like Facebook have revolutionized the way by which information is spread by permitting all of its users to share content freely and easily. As a result of which such websites are progressively being used as vectors for the diffusion of faux news and hoaxes.

As a contribution towards this objective, it showed that Facebook posts are often classified with high accuracy as faux or real on the premise of the users which "liked" them. It bestowed 2 totally different classification techniques, one was using Logistic regression, the second one involved use of Boolean crowdsourcing algorithms.

III. PROPOSDED METHODOLOGY PANDAS

Pandas is a Python library which can be used for the tasks of analysing and manipulating data in an easy manner.

We can do the following:

Create DataFrame objects for manipulating data with automatic indexing. Read and write from different formats of files and data structures. Easily handle missing data and clean it efficiently. Reshape and pivot datasets readily. Label-based slicing, fancy indexing. The library is highly optimized for performance, as it is written in C it is very fast.

SCIKIT LEARN

Scikit-learn is a library use for the tasks of Machine Learning created for the programming language Python. It can be used to implement various algorithms like classification, clustering, regression, SVMs etc Eg: Random Forests, Logistic Regression, Multilayer Perceptron. It can readily interoperate with the unique python libraries used for numerical and scientific analysis called NumPy and SciPy.

MATPLOT LIB

Matplotlib is a Python 2D plotting library which can work with various interactive environments and is also cross platform. Matplotlib is available to be used inside scripts of Python and I Python shells like the Jupyter notebook and also be used in web servers.It can be used to generate: Normal plots, Bar charts, Histograms, Violin Plots, Scatterplots etc

NLTK

NLTK or Natural Language Tool Kit is used for:- classification,tokenization,stemming,tagging,parsin g,semantic reasoning

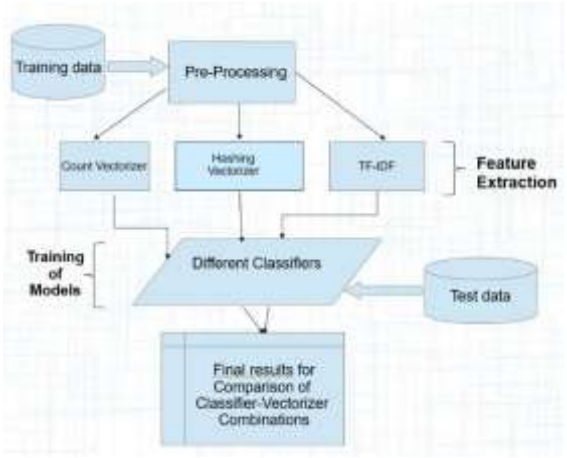

Figure 1 : Architecture

A.DATA PRE-PROCESSING

The first step was to download a Fake News dataset form Kaggle.com. The CSV (Comma Separated Val ues) file obtained had 12,999 fake news items. Anot her CSV file was obtained which had 3171 real new s articles. Using the Pandas library for Machine Lear ning in Python, both of these were put into 'DataFra mes'. After this the DataFrames were edited and cle aned up to get rid of unnecessary fluff columns such

as 'author', 'published-date' etc. A 'Label' column

was added to each separate DataFrame with the strin g values "REAL" and "FAKE" for the entire frame. This will later be used to append Boolean True and F alse values which are mandatory requirements for an y classifier in a Binary Text Classification problem s uch as this one. After all the operations the 2 datafra mes where merged into a single data frame. We chos e to keep the percentage of Fake News items in the t otal corpus to approximately 15%. This Dataframe n ow needed an Index. Random numbers were generat ed and appended to a list then using the Insert() meth od of DataFrame, this list served to provide the Inde x values for our new DataFrame. Then the DataFram e was sorted based on the Index. Therefore the FAK E and REAL news articles were now randomly distr ibuted in the DataFrame. From these very small articles, articles with junk (g arbage scraped while collecting data) were removed and cleaned manually using custom built functions f rom the NLTK library. Finally the Test+Train datase t size was 3406. Training data size = 2554

Testing data size = 852.

B. FEATURE GENERATION

The challenge in text classification is that text as a sequence of words cannot be used as input in machine learning algorithms directly. The textual information of fake news provided in the content of their articles are the best source for determining their credibility. We will extract this text and represent it as fixed-length vectors. The "method of representing text as vectors is commonly referred to as text vectorization. For the project we decided to use the n-gram model." In this model, the sentences are split on whitespaces or punctuation as separators and represented as a multiset of their words. The value of

'N' can be tweaked as desired. This will change the

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

size of the multiset. A 'N' value = 1, 2 and 3 will be used in this project. This model removes some information about the structure of the text, the "grammar and word order, but keeps the multiplicity of each word in the text. The idea is to use the number of occurrences as a feature in training the classifier."

Consider the following document "Ram enjoys to go out. Amit also enjoys to go out", "Ram also enjoys to go swimming". With N=1 splitting after each whitespace and punctuation, leads to the following multiset of words: ["Ram", "enjoys", "to", "go", "out", "Amit", "too", "also", swimming"]. In the example, the two sentences would result in these two vectors:

(1) [1, 2, 2, 2, 2, 1, 1, 0, 0] (2) [1, 1, 1, 1, 0, 0, 0, 1, 1] C. FEATURE EXTRACTION

A.COUNT VECTORIZER

The Count Vectorizer provides the simplest way to tokenise a collection of text documents. It converts a collection of text documents to a matrix of token counts. The fit () method "learns a vocabulary dictionary of all tokens in the raw documents.

The transform () method was used to "transform documents to a document-term matrix.". This matrix was a "sparse matrix", meaning that most of its elements are 0. Using sparse matrices reduces computing time drastically and requires less storage.

For e.g. with N =1 the output was

<2970x43245 sparse matrix of type with 792499 stored elements in Compressed Sparse Row format >

Here the first value (2970) corresponds to the number of documents in the matrix and the seconds value is the number of words in the vocabulary, which is essentially the total number of features. As the value of N will be increased the total number of elements in the sparse matrix will be increased.B. TF-IDF VECTORIZER

TF-IDF itself is short for "term frequency-inverse document frequency. Term Frequency TF (t, d) is defined as the frequency of the Term (t) in the Document (d). It is the raw count of the term. There are other variations of the method of calculating term frequency also, but we will not consider those for the purpose of this project.

Inverse Document Frequency is a good measure of how valuable a term is in the larger context. Small Inverse Document Frequency means that the terms are rare" e.g. Proper Nouns like names. Large Inverse Document Frequency mean means the word is very common, this includes words which are articles or prepositions e.g. 'A', 'An', 'The', 'on, 'over',

'under' , 'near' etc .To calculate Inverse Document

Frequency, the formula is

$$ID\text{}(\text{}, \text{}) = l\text{} \frac{N}{|\{\text{} \in D : \text{} \in \text{}\}|}$$

Here, N = Total number of documents in the corpus

(D),The denominator is the number of documents (d) where the term (t) occurs,TF-IDF is then calculated as

$$\text{hash}(w, \text{sign}) = \text{hash}(w) \cdot \text{sign}(w, \text{sign})$$

C. HASHING VECTORIZER

This one is designed to be as memory efficient as possible. Instead of storing the tokens as strings, the vectorizer applies the hashing trick to encode them as numerical indexes. The downside of this method is that once vectorized, the features' names can no longer be retrieved

- *John likes to watch movies.*
- *Mary likes movies too.*
- *John also likes football.*

can be converted, using the dictionary

Term	Index
John	1
likes	2
to	3
watch	4
movies	5
Mary	6
too	7
also	8
football	9

to the term-document matrix

$$\begin{pmatrix} \text{John} & \text{likes} & \text{to} & \text{watch} & \text{movies} & \text{Mary} & \text{too} & \text{also} & \text{football} \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Figure 2: Hashing Vectorizer

D. CLASSIFIER SELECTION AND IMPLEMENTATION

To ensure sufficient variation in the nature of the Machine Learning algorithms. The available choices were classified into categories These were

-Functional Classifiers

-Tree based Classifiers

-Probability Based Classifier

We selected one algorithm from each. Before beginning implementation of the classifier, we needed to split the total dataset into training set and a test set.

A. LOGISTIC REGRESSION

Logistic regression is "a predictive analysis." We are using it to "describe data and to explain the relationship between one of our dependent binary variables and one or more of our independent variables." Since it requires a dependent variable which is dichotomous in nature, it is well suited for this project where news articles will be

labelled as

'FAKE' vs 'REAL'. We will be using Binary logistic regression because this is a binary text classification problem. It the maximum likelihood estimation (MLE). The method's resulting values are between 0 and 1 and its general scheme is as follows

$$P(y = 1)$$

1

form a loop. Artificial Neural Networks (ANN) try to mimic the brain and are based on units and connections between them. Each connection transmits a signal from one unit to another. The signal is a real number. Each unit has inputs and output that is calculated by a non-linear function using all input values. It has "an input layer, an output layer and 1 or more hidden layers." The model itself is a supervised learning technique and uses backpropagation, which is short for backward propagation for errors. This is useful for weights calculation. Layers after "the input layer are called hidden layers because that are not directly exposed to the input. The simplest conceivable network is to have a only one neuron in the hidden layer that will directly output the value."

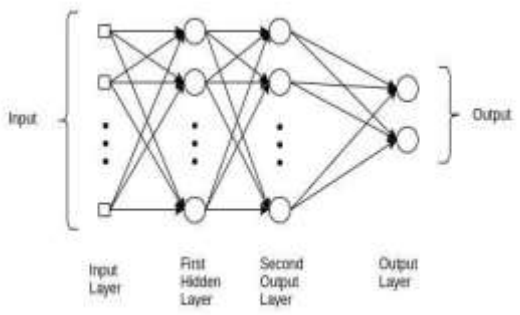

Figure 3: Neural Network structure

$$= \frac{}{1 + e^{-z}}$$ [illegible]

IV. EXPLORATORY DATA ANALYSIS

The linear function 'z' consists of dependent and independent variables along with error bias value

B. RANDOM FOREST CLASSIFIER

Random Forest Classifier, a tree-based classifier, is "an ensemble algorithm. Ensembled algorithms are those type of algorithms which combine more than one algorithm of same or different kind for classifying item. This classifier creates a set of decision trees from randomly selected subset of our training set. Then it aggregates the votes from different decision trees to decide the final class of our item in the test set." Random Forest classifiers use tree-based classifiers as a base for their algorithm. These decision trees are not correlated and grow randomly during learning. For every classification, the class that most trees assign to the input, decides the final classification. We are using Random Forest classifiers because they are considerably fast during training and the evaluation is parallelized. Thus, it is very efficient for our large datasets.

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

C.MULTILAYER PERCEPTRON

It is a class of feedforward artificial neural networks. Feed forward network means that the nodes do not

A. CONFUSION MATRIX

After all the classifier-vectorizer combinations were executed we wanted to measure the performance the format of the matrix in a binary classification is shown above. Here we can see there are 4 values of concern.

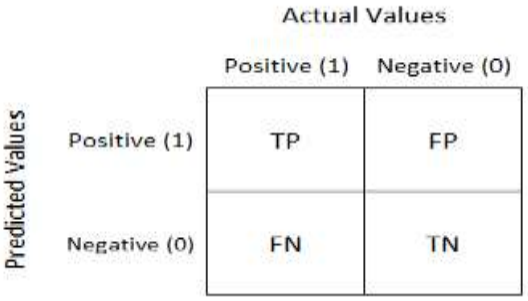

Figure 4: Confusion Matrix

True Positive (TP): Item is of positive label and the predicted label was also positive.

False Positive (FP): Item was falsely predicted as positive

False Negative (FN): Item was falsely predicted as negative.

True Negative (TN): Item is of negative label; the predicted label was also negative.

We generated a confusion matrix for each of the vectorizer-classifier combinations. Therefore 36 total confusion matrices were generated. One point of distinction in our project was that we had assigned “TRUE” Boolean value to the “FAKE” label therefore the TN value was the largest for most matrices we generated in our experiment

B. METRICS: F1-SCORE, PRECISION AND RECALL

PRECISION: The formula for precision is given below:

$$\text{PRECISION} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

By measuring precision, we are able to measure how

well a particular classifier-vectorizer combinationlabelled a class positively, meaning it did not label a class negatively when it was positive

A higher precision is considered better

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

RECALL: The formula for recall is given below

$$R\text{✿✿✿✿✿✿}l = \frac{\text{✿✿✿✿✿ ✿✿✿✿✿✿✿}i\text{✿✿✿}}{(\text{✿✿✿✿✿ ✿✿✿✿✿}i\text{✿✿✿} + Fa\text{✿✿✿✿ }N\text{✿✿}a\text{✿✿✿✿✿✿✿})}$$

As we can understand from the formula, recall

measures how well a classifier labels a class positive relative to the total number of positive class item in the total dataset. It's useful in our fake news detection project because we have given positive label to FAKE label. The range is between 0 and 1

F1-SCORE: The F-1 score may be defined as the H.M (harmonic mean) of precision and recall. This is how it has been traditionally defined It may be interpreted by the user as an average (taking weight into consideration) of precision and recall.

C. PERFORMANCE METRICS

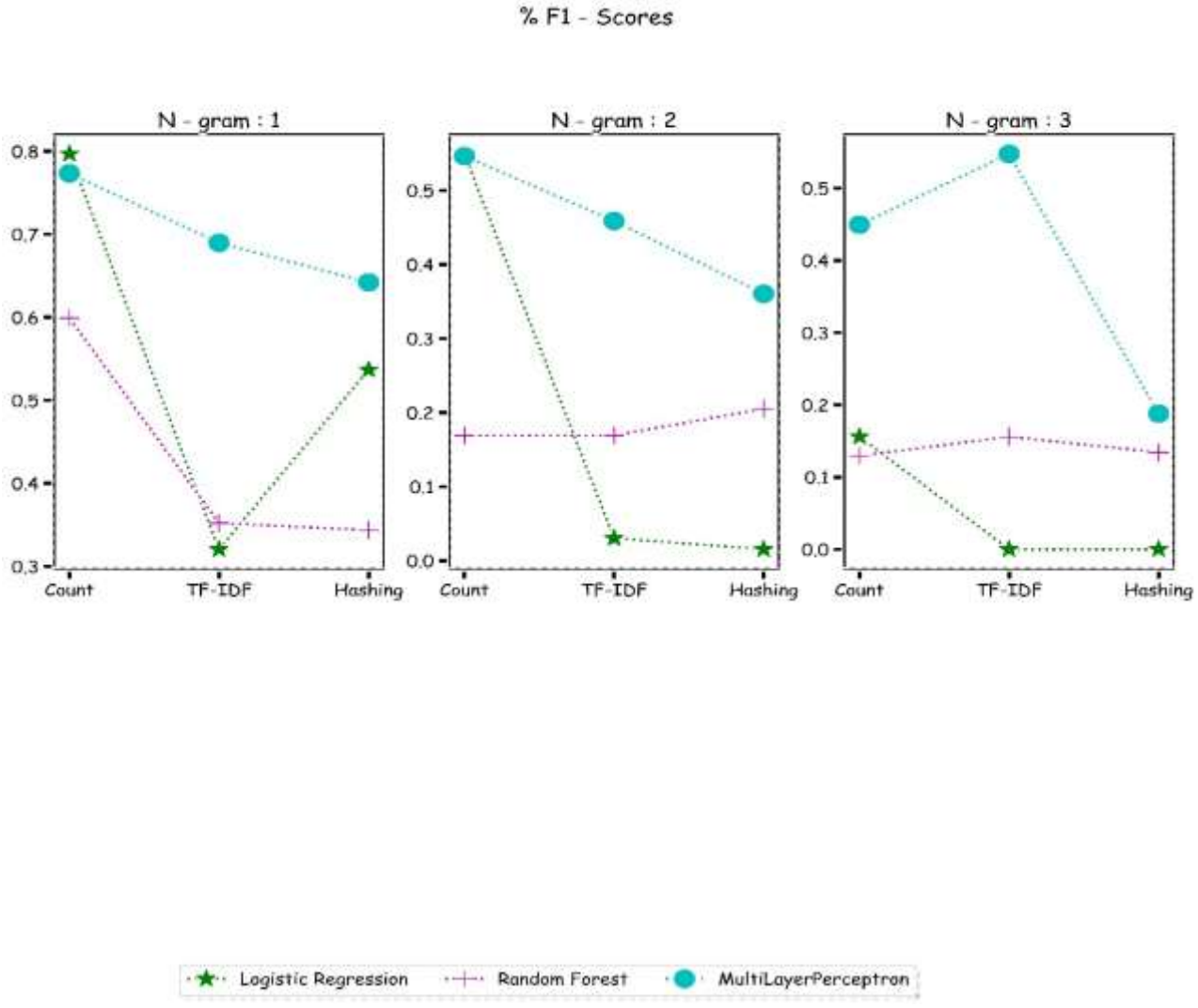

Figure 5: F1 scores

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

CLASSIFIER/ VECTORIZER	LOGISTIC	RANDOM FOREST	MULTILAYER
COUNT (N=1)	F: 0.7967479674796748	F: 0.329268292682926	F: 0.7733333333333332
COUNT (N=2)	F: 0.5494505494505495	F: 0.169014084507042	F: 0.546448087431694
COUNT (N=3)	F: 0.1560283687943262	F: 0.129496402877697	F: 0.44943820224719094
TF-IDF (N=1)	F: 0.32051282051282054	F: 0.352201257861635	F: 0.689655172413793
TF-IDF (N=2)	F: 0.0303030303030303	F: 0.169014084507042	F: 0.4588235294117647
TF-IDF (N=3)	F: 0.0	F: 0.156028368794326	F: 0.547486033519553
HASHING (N=1)	F: 0.5368421052631578	F: 0.34355828220858897	F: 0.6419753086419754
HASHING (N=2)	F: 0.015267175572519085	F: 0.2051282051282051	F: 0.3605150214592275

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

HASHING (N=3)	F: 0.0	F: 0.1342281879194631	F: 0.18779342723004697

Table1:Performance Metrics

MODEL PERFORMANCE

After seeing all the F-1 score we concluded that the Multi-Layer Perceptron model had the good and acceptable performances. Logistic Regression had average performance Random forest had extremely poor performance. This was not an entirely unexpected result , Multilayer Perceptron is the preferred choice of model in many other text classification tasks such as 'spam-filtering' and

'sentiment analysis' so to see the model performing

well in this context of fake news detection is a trend that was expected and is a positive

EFFECT OF VECTORIZERS

Unfortunately, as per the results the different vectorizers did not manage to increase the accuracy enough to have any form of substantial effect on the scores. Hashing Vectorizer performed poorly in places where other vectorizers did well for Simple Layer perceptron.

EFFECT OF N-GRAM SIZES

Increase in N-gram size did not bring substantial increase performance of classifiers or F-scores. It instead brought a decrease in the performance of the classifiers and also lead to an increase in time required for computation greatly as well as RAM space required. The n-gram 2 and 3 took more than an hour to execute for Multi-Layer Perceptron model with Counting and TF-IDF vectorizer.

Ultimately, we can see that adding features did not help in fake news detection problem like it does in other text classification problems.

V. CONCLUSION

After implementing and executing 3 categorically different classifiers, 3 different and unique text vectorizers, a total of 27 different combinations by tweaking the n-gram size of each vectorizer thrice in the range of n = 1 to 3 and obtaining 81 unique(F,P,R) metric values we built our conclusion.We concluded that "Fake news" detection using Machine Learning will be best performed by a Neural Network. This was justified by the performance of the Multilayer Perceptron classifier outperforming Logistic Regression especially in the case of Count Vectorizer and n-gram size equal to 1. We found that increasing the n-gram size to add features to the

classification process did not bring about desired or appreciable improvements in the performance metrics of precision, recall and F1-score. We found that changing the vectorizers to TF-IDF and Hashing vectorizer improved the computation time, especially in the case of hashing vectorizer the performance improved greatly but the trade-off was in the performance. The Random Forest classifier performed the worst of all the classifiers. It is not suited for this task of binary text classification. This was in part due to the fact that it could not handle larger dataset. An interpretation of this project can be that Multilayer Perceptron can accurately identify 8 out of 10 fake news articles from a set of larger news articles provided to it. This is proving that it is the best classifier among those tested for Fake News Detection

VI. FURTHER WORK

From our results it can be confidently said that looking into Neural Networks for this Fake News Classification problem will be fruitful and may even give us better results than the Multilayer Perceptron.After optimizing a Model it can deployed on a webserver along with a good web scraper in order to detect fake news in real time by taking URL as input from the user and hence drastically in minimising the damage caused by Fake News.

REFERENCES

[1] “Pérez-Rosas, Verónica & Kleinberg, Bennett & Lefevre, Alexandra & Mihalcea, Rada (2017). Automatic Detection of Fake News.” (https://arxiv.org/abs/1708.07104v1)

[2] “E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro, Some Like it Hoax: Automated Fake News Detection in Social Networks.” (http://arxiv.org/pdf/1704.07506.)

[3] “Thota, Aswini; Tilak, Priyanka; Ahluwalia, Simrat; and Lohia, Nibrat (2018) "Fake News Detection: A Deep Learning Approach," SMU Data Science Review: Vol. 1: No. 3, Article 10 https://scholar.smu.edu/datasciencereview/vol1/iss3/

10”

[4] “Getting Real about Fake News [Online]

https://www.kaggle.com/mrisdal/fake-news/data”

[5] “Detecting Fake News with Scikit-learn” “https://www.datacamp.com/community/tutorials/sc ikit-learn-fake-news”

[6] W. Y. Wang, "Liar, Liar Pants on Fire": “A New Benchmark Dataset for Fake News Detection. Available: http://arxiv.org/pdf/1705.00648.”

[7] Anaconda distribution https://www.anaconda.com/distribution/

[8] “scikit-learn: machine learning in Python — scikit-learn 0.20.2 documentation. [Online] Available: http://scikit-learn.org/stable/.”

[9] “Text Classification. A Comprehensive Guide to Classifying Text with Machine Learning https://monkeylearn.com/text-classification/”

[10] “Natural Language Processing course of National Research University Higher School of Economics https://www.hse.ru/en/edu/courses/219930752”

[11] “How Fake News Goes Viral: A Case Study https://www.nytimes.com/2016/11/20/business/med ia/how-fake-news-spreads.html”

[12] “Fake News Is Not the Only Problem https://points.datasociety.net/fake-news-is-not-the- problem-f00ec8cdfcb”

[13] “We Tracked Down A Fake-News Creator In The Suburbs. Here's What We Learned https://www.npr.org/sections/alltechconsidered/201

6/11/23/503146770/npr-finds-the-head-of-a-covert- fake-news-operation-in-the-suburbs”

[14] “Johnson, J.: 2016. The Five Types of Fake News. https://www.huffpost.com/entry/the-five-types-of-fake-ne_b_13609562”

[15] “F1 Score documentation available with Scikit- learn module.

[16] NLTK Documentation available with nltk module