COVID FAKE NEWS CLASSIFICATION WITH NATURAL LANGUAGE PROCESSING

Raj Sanjay Kulkarni[1], S Vinod[2], Beatriz Lucia Salvador Bizotto[3], Maria Emilia Camargo[4], Mithileysh Sathiyanarayanan[5]
[1,2]Department of Computer Science Engineering, SRM Institute of Science and Technology, Ramapuram Campus, Chennai, India
[3,4]University Center Unifacvest, Brazil
[5]MIT Square Services Private Limited, London

Abstract—we are already dealing with a pandemic of fake news in today's rapidly expanding world of technology and journalism. It is difficult to predict if a news story is fake or authentic because there are thousands of sites on the internet which suggest different things. This type of misleading information causes a lot of violence and chaos among people. To avoid this, we have worked on "COVID Fake News Classification". The main aim of this paper is to develop a classifier that segregates the news into fake and authentic with an accuracy around 90%. We collected a public dataset consisting of news and used simple NLP technique for the pilot study; compared and validated our trained model using three machine learning techniques – Multinomial Naive Bayes, Passive Aggressive and Logistic Regression (LR) where Logistic Regression showed the maximum accuracy (~93%).

Keywords—fake news detection, natural language processing, machine learning

I. INTRODUCTION

Nowadays the internet is an integral part of our everyday lives and, at the same time, one of the most important data sources for users. However, as a result of social media and other forms of media, we are constantly bombarded with fake news on the Internet. In reality, the entire fake news model necessitates a substantial amount of effort and a complex dataset. These days, there are an increasing number of new messages and articles appearing on the internet. The fake news may be a phenomenon that relates to varied topics, which is continuously read by many users. This effect is very favorable for those who wrote this fake news. Fake news shatters the equilibrium of the news ecosystem.

Fake news has become a major threat causing panic, especially during difficult times such as the covid pandemic. Despite the fact that the Internet is one of the most important sources of information, social media has allowed for misinformation which is a potential cause for widespread panic. There have been many such instances where fake news had caused a lot of protests all around. For example, when the Rs. 500 and Rs. 1000 notes were banned in India to combat black money, bogus news arose in the market claiming that "the Reserve Bank of India (RBI) has implanted chips in every note it has created to combat corruption and black money." This sparked a lot of public outrage because the fake news led them to feel that they were being watched and listened to continually by the government, resulting in a lot of discussions and protests.

In this era of pandemic, we are surrounded by a lot of fake news which does not only affect our mental health but also our physical health. News emerged saying that to fight coronavirus, we are required to take steam more than 10-15 times a day. For a long time, many people believed this rumor until it was reported on the news and debunked by one of the doctors on a major news television channel, who stated that "by inhaling too much steam, one may start to burn their windpipe due to too much heat taken for a longer period". Anybody with access to the internet can start fights in any part of the world. WhatsApp, Facebook and all other social media sites are a hub for fake news. Amplification of such fights/chaos is amplified on these social media sites. The main aim of the paper is to develop a classifier to classify fake and authentic news using machine learning algorithms.

For establishing a viable fake news detection model, extensive dataset is required. Although good accuracy can be achieved, considering the scale at which fake news is being spread, models need to be trained on huge datasets, and should classify correctly and in real time [6]. The entire process of detecting fake news is a multi-step process. Previous research [Jozef Kapusta and Juraj Obonya, 2019] aimed to analyse speech characteristics to classify fake and real news. Using a decision tree with a maximum depth of 9, the previous authors were able to attain a 75 percent accuracy. In the following research, publicly gathered (Kaggle) and custom dataset from misleading content is used. We use logistic regression as a classifier and apply upon processed data. Results from studies [5] note that logistic regression shows good performance in classification. Current project produces similar results as observed in previous research. An accuracy of 93% is achieved, applied on nearly 18000 news articles. The contributions of this paper are:

1. The survey of related works gave us an idea about various methodologies and approaches used for fake news detection. We have ensured that the reader understands each and every definition in a detailed way. We have broken colossal topics of countvectorization, stemming, etc., into simple tables and examples.
2. Implementation using the base models; the accuracy achieved is higher when compared to other works

which have used ensembling, bagging, etc. With lesser complexity, we have achieved higher accuracies.

3. The results and conclusions with the help of confusion matrix will help readers understand the statistics of classified data (fake and authenticity). Using confusion matrix, as a visualization tool to analyze, gives a clear distinction and accuracy between the different models used.

II. LITERATURE REVIEW

Literature review is done in order to understand what progress/advancements have been made in the topic/field so far. Many researchers have made a lot of conclusions on fake news detection. Many of them have used simple approaches whilst others have used complex structures or algorithms like neural network, deep learning, etc. These papers introduce us to the fundamental concepts of Artificial Intelligence and Machine Learning which comprises topics like Natural Language Processing, Ensembling, etc.

In paper [1], to deal with the multidimensional nature of fake news, the authors used a combination of Naive Bayes classifiers, Support Vector Machines, and semantic investigation. They utilized SVM with Naive Bayes to extract the binary class based on the data provided to the model because SVM is best suited for binary classification. Their survey discusses proper explanation of methodologies highlighting 3 main modules, Aggregator - to obtain source for news, News Authenticator - to compare news given by them with different news sources and Recommendation system - this displays related news that the user has provided as part of the authentication process. The implementation of SVM and Naïve Bayes classifiers with NLP gives 93.6% accuracy. Despite good accuracy, they stated that considerably more input and training was required, and that determining the category of news was difficult owing to the nature of fake news.

Z Khanam et. al. [2] propose a model based on words, phrases, sources, and titles, and they train it using supervised machine learning methods on annotated (labelled) datasets. The authors use the Python scikit-learn module for feature extraction and vectorization because it provides handy functions like CountVectorizer and Tiff Vectorizer. They use the common benchmark dataset called Liar-dataset, for Fake News Detection. In their extensive research of many papers which focus on characteristics of fake news and sentimental analysis, they notice that most of them use Naïve bayes for qualitative analysis. Owing to their findings, POS textual analysis, a quantitative approach using random forest classifiers was used in the process to increase the accuracy along with other mentioned methodologies. Among the six algorithms, (Naïve bayes, Support Vector Machine, Random Forest, KNN, Linear Regression, XGBoost), they were able to get the highest accuracy of over 75% with the XGBoost algorithm. By using all the different algorithms, the execution cost of individual classifiers being high, their overall cost was high.

In paper [3], the focus of literature on specific datasets or domains, most notably the politics domain, is highlighted in their survey. On four large real-world publicly available datasets, the authors apply a combination of different machine learning methods. The utilisation of ensemble methodologies, as well as the Linguistic Inquiry and Word Count (LIWC) feature set, is a key aspect of their approach. From their ensemble approach, they achieved an accuracy of 99% with the random forest algorithm and Perez-LSVM on ISOT Fake News Dataset. Ensemble approaches such as bagging, boosting, and vote classifier were utilised to evaluate the performance across many datasets, according to the methodology. They also use four performance metrics to verify their findings: accuracy, precision, recall, and F-1 score. They note that Wang-Bi-LSTM (accuracy 64.25%) had the lowest performance. Their model analyzes data in a specific domain. This causes the algorithm to be trained on a specific type of article's domain and to perform poorly when applied to articles from different domains.

Uma Sharma et. al. [4], using Natural Language Processing and Machine Learning, demonstrate a basic strategy for detecting fake news. They allow the user to classify news as fake or real, as well as verify the legitimacy of the website that published the news. The authors take the fake news detection approaches that are heavily based on text- based analysis. Their methodology involves classifying fake news based on several characteristics such as grammatical mistakes, attention seeking words and click bait. They analyse based on psychological factors. K-fold cross validation technique is used to improve the effectiveness of the models. They achieved a 75 percent accuracy using grid search parameter optimization to improve the performance of logistic regression.

According to the authors [5], Logistic Regression is used for fake news detection on a dataset focused on the Malay language. The authors used Logistic Regression as they observed that the classifier performed well in classification tasks. Furthermore, because of its predictive strength in probability values, Logistic Regression has proven to be highly useful for writers in addressing binary classifications. Along with the stance detection approach, they were able to increase the accuracy using the Term Frequency–Inversed Document Frequency (TF-IDF) feature applied to text pre-processing. They note the distinction between content based and stance-based model with stance based outperforming content based. They also mention the limitations in developing the model, stating that the data collected is insufficient and not quality enough to fit the model for data training. It also has inaccuracies with fresh data, wherein it predicts the news label wrongly.

As far as paper [6] is concerned, Jozef Kapusta and Juraj Obonya worked on floating languages, and by making use of approaches based on morphological group analysis, they analyze the authenticity. Their main aim being improving the methods of dataset preparation. They used their own unique dataset in the Slovak language (a floating type) from different verified publishers. They concentrate on exploiting part of speech characteristics to classify fake news messages.

In flective languages like Slovak , emphasize the relevance of parts of speech and word shape characteristics. The authors use decision trees over many depths to analyse results and were able to achieve 75% accuracy with depth of 9. Similar to paper [5], the authors discuss their limitations, that low accuracies are observed due to low quality dataset and limited examples.

III. METHODOLOGY

A. Material

For this project, we used a dataset which was available on Kaggle. The dataset consists of id, title, author, text, label. The ‘label’ column is the dependent feature in our dataset. It is the final output [1 for real and 0 for fake news].

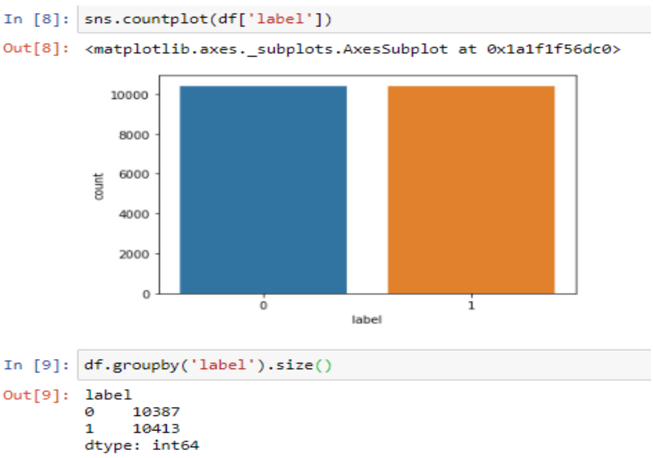

Fig.1. Representation of the ‘label’ column which consists of fake (0) and real (1)

B. Implementation

In an overview, our project is based on three simple parts – Data Management, Model Training, Model Evaluation.

- Data Management - In this phase, data/dataset is collected. Once we collect it, we clean the dataset and pre-process it. Example, we drop the 'Nan' (Not a number) values, reset the index, etc.
- Model Training - After the data is cleaned and pre-processed, we train the model. In this, we train the model for prediction tasks. Features are studied and then are fit or fit-transformed for machine readable data.
- Model Evaluation - We then evaluate the model and feed it to our algorithms and predict the accuracy of our individual models. We then plot the confusion matrix based on the correctly and incorrectly classified samples for better understanding.

We can also do Model Integration and Model Deployment by using ‘pickle’ after we complete our code in Jupyter Notebook. After pickling the model, which has given the highest accuracy, we can import it in our IDE, like VS Code, PyCharm, Spyder, etc., and then deploy it via flask, Django, etc. In model integration, API Services can also be used which can act as web service for the portal we have created.

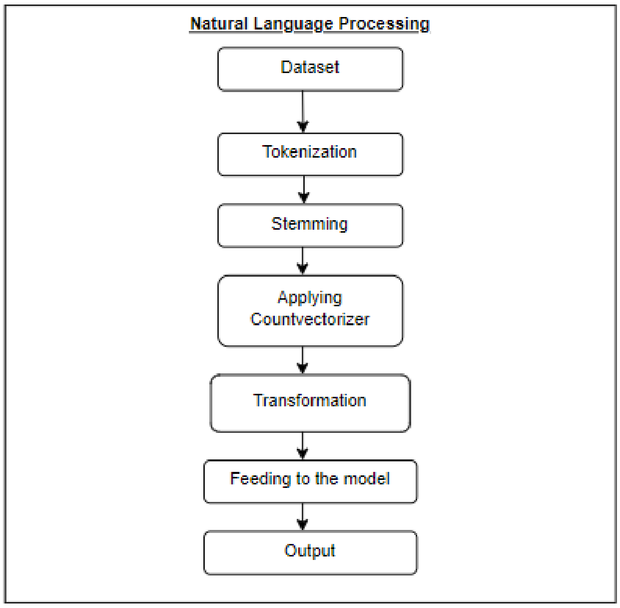

Fig.2. Flowchart of methodology implemented in this study using NLP

Before understanding the model or the methodology, we need to understand the main idea. Natural Language Processing (NLP) is a branch of machine learning in which a computer understands, interprets and sometimes even manipulates data or human language for further use. In an NLP model, when we work with textual data, we need to do a lot of processing to the textual data and then feed it to the model.

As we know that the machine/model does not understand textual data, we convert the textual data into numbers for feeding into our model. The textual data is vectorized and this process is called vectorization. Vectorization or Feature-extraction is the process of converting textual format into machine-readable format. These words are represented as vectors. For this we use Countvectorization or Bag-of-words (BoW). The fundamental goal of BoW is to turn a sentence or paragraph into a meaningless jumble of words. After this comes the part of Countvectorizer. To understand CountVectorizer, we must understand the term ‘tokenization’.

Tokenization is the process of converting or breaking down a sentence into words which have more importance. In simple words, it means to convert each word into a separate token. In this, we remove the stopwords. Stopwords are the words which occur more frequently in the document and are not important. Examples of such words are he, she, it, and, the, etc.

In this process, we convert the words to lowercase. We do this because there can be various words which are similar but will not match because one word can be uppercase while the other might be lowercase (python being case-sensitive language). Example – “He is a Good boy whereas she is not a good girl”. In this example, ‘Good’ and ‘good’ will not be treated the same because of their ASCII value. Due to this, we convert everything to lowercase and then remove all the stopwords. Hence, most of the punctuations and stopwords have been removed.

Now, countvectorizer is used to convert a collection of text documents or corpus into a vector of terms. In simple words, it will look at a sentence in a corpus or text document and assign it values which indicate the number of times it has repeated in a line or a text (in our case, title). Let us take an example to understand countvectorizer.
Consider the following example–
“Out of all the countries of the world, some countries are poor, some countries are rich, but no country is perfect “

Table.1. Countvectorizing a example sentence

	out	of	all	the	countries	…
doc	1	2	1	2	3	

Table.2. Indexing Table 1

index	0	1	2	3	4	…
doc	1	2	1	2	3	…

Table 1 is to represent it in a basic way to understand countvectorizer visually whereas Table 2 is the actual representation of how a countvectorizer will work. The numbers assigned in each column, under each word, represent how many times the word has been repeated in the text (after consideration of root words).
Instead of countvectorizer, we can also use the Term Frequency-Inverse document frequency (Tf-Idf) vectorizer [4].
CountVectorizer will choose 5000 words/features that occur most frequently in our corpus.
When these features are converted to array, the result will look like the following (specifying the result of how countvectorizer has worked) –

```
In [42]: X
Out[42]: array([[0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                ...,
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0]], dtype=int64)

In [23]: X.shape
Out[23]: (18285, 5000)
```

Fig.3. Result of applying countvectorizer to our corpus

Fig.4 is representation of the same corpus after applying countvectorizer.

```
In [29]: count_df.head()
Out[29]:
```

	abandon	abc	abc news	abduct	abe	abedin	abl	abort	abroad
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0

5 rows × 5000 columns

Fig.4. Representation of X in detail

After doing all operations on our corpus, we feed it to the model to acquire our best possible result. To do so, we use three primary algorithms– Multinomial Naïve Bayes, Passive-Aggressive and Logistic Regression. Our aim is to inspect the algorithms separately, without ensembling, and find out how accurate the results are after applying countvectorizer.

- Multinomial Naïve Bayes Algorithm -

It is a probabilistic learning algorithm which is mostly used in NLP. It is based on Bayes Theorem and is used for predicting the tag of a text such as a newspaper article or a part of an email. It calculates probability for each tag and returns the tag with the highest probability as output. The Bayes theorem is a mathematical formula for estimating posterior probability - P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

$$\mathbf{P(c|x) = \frac{P(x|c)\,P(c)}{P(x)}}$$

Above,

- P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of the predictor given class.
- P(x) is the prior probability of the predictor.

- Passive-Aggressive Algorithm –

In simple terms, it's an online-learning method that gradually trains a system. After that, the programmer feeds it instances in a sequential order, either individually or in small groups known as mini-batches. Basically, it will remain passive to correct predictions and aggressive to incorrect predictions. It is generally used for large amounts of data or datasets like Twitter’s tweet analysis.
Passive: Keep the model and don't make any changes if the

prediction is right. i.e., the data in the case is insufficient to generate any model adjustments.
Aggressive: Make modifications to the model if the prediction is inaccurate. i.e., a change to the model could correct the problem.

- Logistic Regression -

Despite its name, logistic regression is a classification model rather than a regression model. For binary and linear classification problems, logistic regression is a simple and more efficient method. It's a classification model that's simple to implement and delivers excellent results with linearly separable classes. In the industrial world, it is a widely used categorization method. It is a classifier algorithm which classifies the output of models into '0' or '1'. For example, the model classifies if the mail received is spam or not. If the mail is spam, it classifies it as '0' and if it is not, it is classified as '1'.

A binary logistic regression model will have only 2 outputs, which can be true/false, yes/no, 0/1, and so on. The Multinomial Logistic Regression model will have more than two outputs. Logistic regression is a handy analysis tool for determining if a fresh sample fits best into a category in classification tasks. Because components of cyber security, such as threat detection, are classification problems, logistic regression is a valuable analytic tool.

IV. RESULTS

After feeding the input to our models, the following accuracy was achieved. Accuracy is defined by the percentage of correct predictions. In mathematical way, accuracy is:

Accuracy = number of correct predictions
total number of predictions

While performing classification, the following are four types of outcomes that can occur.

- True Positive – prediction is true and in actual it is true
- False Positive - prediction is true and in actual it is false
- True Negative - prediction is false and in actual it is false
- False Negative - prediction is false and in actual it is true

When the four outcomes are plotted on confusion matrix,

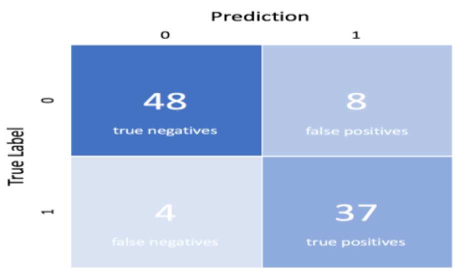

Fig.5. Representation of a confusion matrix

We can also find out the Precision, Recall and F1 score.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{F1} = \frac{2\ X\ Precision\ X\ Recall}{Precision + Recall}$$

For our project, we restricted ourselves with 'Accuracy' of the three algorithms used.

Confusion Matrix for Multinomial Naïve Bayes Classifier using Countvectorizer–

There were 5439 correctly classified samples and the accuracy was 90.12%

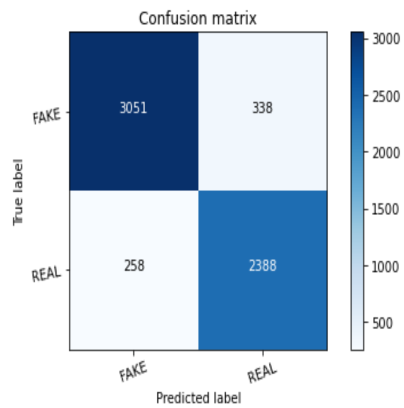

Fig.6. Confusion Matrix for Multinomial Naïve Bayes

Confusion Matrix for Passive-Aggressive Classifier using Countvectorizer–

There were 5555 correctly classified samples and the accuracy was 92.04%

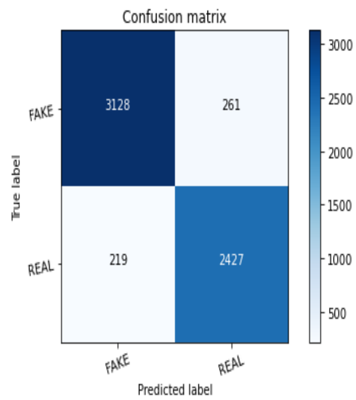

Fig.7. Confusion Matrix for Passive Aggressive

Confusion Matrix for Logistic Regression using Countvectorizer–

There were 5635 correctly classified sample and the accuracy was 93.37%

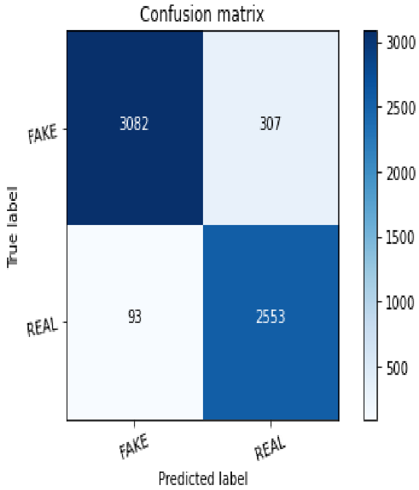

Fig.8. Confusion Matrix for Logistic Regression

Table.3. Accuracy of each algorithm

Sl. No	Algorithm	Accuracy
1	Multinomial Naïve Bayes	90.12%
2	Passive-Aggressive Classifier	92.04%
3	Logistic Regression	93.37%

V. CONCLUSION AND DISCUSSION

To tackle the pandemic of fake news, we discussed the classification of fake news with three machine learning algorithms to minimise the spread of misinformation namely- Logistic Regression, Multinomial Naïve Bayes and Passive-Aggressive. These models help us classify more than thousands of news stories in a very short amount of time with utmost precision. Our primary aim is to train models to improve classification. We used countvectorizer to format input data and the algorithms were applied. Logistic Regression gives higher accuracy (93.37%) when compared to Multinomial Naïve Bayes (90.12%) and Passive Aggressive (92.04%) in terms of simple classification problems. We used a confusion matrix to compare the results of each model. As discussed,

1. The survey of related works gave us an idea about various methodologies and approaches used for fake news detection. We have ensured that the reader understands each and every definition in a detailed way. We have broken colossal topics of countvectorization, stemming, etc., into simple tables and examples.
2. With respect to the implementation using the base models; the accuracy achieved is higher when compared to other works which have used ensembling, bagging, etc. With lesser complexity, we have achieved higher accuracies.
3. The results and conclusions with the help of confusion matrix will help readers understand the statistics of classified data (fake and authenticity). Using confusion matrix, as a visualization tool to analyze, gives a clear distinction and accuracy between the different models used.

Fake news detection and classification requires a lot more data and processing to get good accuracy. A major problem with the datasets present is the format in which they are given. There are many rows in the dataset which consist of symbols and random words which are a result of formatting of the csv or excel file (dataset). Considering the problems from a research perspective, no news can be found as fake or true until they are not present in the dataset which is used for classification (unless web-scraping is not used). When trying to enter a new input, which is not present in the dataset, we find that the model decides how genuine the news is according to the countvectorizer assigned to the tokenized value of the input. The referred research works use many different algorithms which are different from our approach. As for [6], they have used morphological tags and have processed their data on that logic whereas we have used CountVectorizer to process our corpus.

In [4], to improve the models' effectiveness, they implemented the K-fold cross validation technique. Their approach is very similar to the approach we followed in this project. The authors have used Tf-Idf whereas we have used countvectorizer. In most of the cases, Tf-Idf stands out to be better than countvectorizer. In [1], various results of various papers were made and their results were tabulated. Support Vector Machine with an accuracy of 76%, Naïve Bayes with an accuracy of 74% and Natural Language Processing with an accuracy of 86.65% were performed separately and then were ensembled with the respective base models with an accuracy of 93.50% whereas even without ensembling, our model had the lowest accuracy of 90.12% with Multinomial Naïve Bayes and the highest accuracy of 93.37% with Logistic Regression.

Focusing on this article, even if there is no higher achieving algorithm or ensembling applied, we have achieved one of the highest accuracies amongst all the existing research works only with the base models (without Bagging or Boosting). With lesser complexity, we have processed, trained and tested the data up to an appreciable level. These implemented classification base models don’t require much training and give promising results.

The efficacy of these model(s) can be increased with the help of ensembling. Another method which can be used in the future is web-scraping. Anybody could type in a news item and the backend would automatically web-scrape a site on the internet and then would cross-verify by applying advanced machine learning models in determining if that is fake news or not. Another technique which could be implemented is Lemmatization followed by Tf-Idf vectorizer. Tf-Idf would not only focus on the frequency of the words but also give importance of the words in the corpus and Lemmatization would find the root words but with meaning unlike stemming. Various visualisation techniques [7-12] will be considered in the extended research.

REFERENCES

[1] Jain, Anjali, Avinash Shakya, Harsh Khatter, and Amit Kumar Gupta. "A smart system for fake news detection using machine learning." In 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), vol. 1, pp. 1-4. IEEE, 2019.

[2] Khanam, Z., B. N. Alwasel, H. Sirafi, and M. Rashid. "Fake News Detection Using Machine Learning Approaches." In IOP Conference Series: Materials Science and Engineering, vol. 1099, no. 1, p. 012040. IOP Publishing, 2021.

[3] Ahmad, Iftikhar, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad. "Fake news detection using machine learning ensemble methods." Complexity 2020 (2020).

[4] Sharma, Uma, Sidarth Saran, and Shankar M. Patil. "Fake news detection using machine learning algorithms.” International Journal of Engineering Research & Tchnology (IJERT) NTASU 9, no. 03 (2020).

[5] Mokhtar, Muhammad Syahmi, Yusmadi Yah Jusoh, Novia Admodisastro, N. Pa, and Amru Yusrin Amruddin. "Fakebuster: Fake news detection system using logistic regression technique in machine learning." International Journal of Engineering and Advanced Technology 9, no. 1 (2019): 2407-2410.

[6] Kapusta, Jozef, and Juraj Obonya. "Improvement of misleading and fake news classification for flective languages by morphological group analysis." In Informatics, vol. 7, no. 1, p. 4. Multidisciplinary Digital Publishing Institute, 2020.

[7] M. Sathiyanarayanan, C. Turkay, and O. Fadahunsi. "Design of Small Multiples Matrix-based Visualisation to Understand E-mail Socio-organisational Relationships." In 2018 10th International Conference on Communication Systems & Networks (COMSNETS). 2017.

[8] M. Sathiyanarayanan,, and N. Burlutskiy. "Design and evaluation of euler diagram and treemap for social network visualisation." In 2015 7th international conference on communication systems and networks (COMSNETS), pp. 1-6. IEEE, 2015.

[9] M. Sathiyanarayanan,, and D. Pirozzi. "Spherule diagrams with graph for social network visualization." In 2016 8th International Conference on Communication Systems and Networks, pp. 1-6. IEEE, 2016.

[10] M. Sathiyanarayanan,, and D. Pirozzi. "Social network visualization: Does partial edges affect user comprehension?" In 2017 9th international conference on communication systems and networks (COMSNETS), pp. 570-575. IEEE, 2017.

[11] M. Sathiyanarayanan, and D. Pirozzi. "Linear-time diagram: A set visualisation technique for personal visualisation to understand social interactions over time." In 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), pp. 259-264. IEEE, 2016.

[12] M. Sathiyanarayanan, AK. Junejo, and O. Fadahunsi. "Visual Analysis of Predictive Policing to Improve Crime Investigation." In 2019 International Conference on contemporary Computing and Informatics (IC3I), pp. 197-203. IEEE, 2019.

