Fake News Detection with Machine Learning Approaches

A PROJECT REPORT

Submitted by

Aaron Antony Noronha (21BAI10296)
Shambavi Pandey (21BAI10453)
Simran Namdev (21BAI10472)
Abhit Yadav (21BAI10397)
Anshu Kushwa (21BAI10141)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

NOV 2021

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **"Fake News Detection with Machine Learning Approaches"** is the bonafide work of "Aaron Antony Noronha(21BAI10296), **Shambavi Pandey (**21BAI10453), **Simran Namdev** (21BAI10472), Abhit Yadav (21BAI10397), Anshu Kushwa (21BAI10141)**"** Who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr.Suthir Sriram,Program Chair
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Dr.Saravanan S, Assistant Professor
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on _____

ACKNOWLEDGEMENT

First and foremost I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr. S. Poonkuntran, Head of the Department, School of Computer Science and Engineering for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide Dr.Saravanan S for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Aeronautical Science, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

ABBREVIATION	WORD
ML	Machine Learning
AI	Artificial Intelligence
KNN	K Nearest Neighbors
ANN	Artificial Neural Network
ROC Curve	Receiver Operating Characteristic Curve
DL	Deep Learning
SVM	Support Vector Machine
RNN	Recurrent Neural Network

LIST OF FIGURES AND GRAPHS

ABSTRACT

Fake news is a huge problem and is a genuine issue that can have a destructive impact on people's lives and this is done through the word that is spread around between people. This project is based on making a machine learning model that helps detect fake news. We use machine learning and deep learning approaches along with Python and various other training models like naive-bayes and recurrent neural networks to develop the model. By developing various models and including processes like vectorization, tokenization, and pre-cleaning the data, we develop the model with the highest accuracy to predict the data.

PURPOSE

Fake news is a huge real time problem as it leads to the spread of unreliable information being spread. The purpose of this project is to detect these fake news articles and prevent them from being spread thus leading people to get proper

information as listening to fake news reduces a person's credibility as basing your arguments based on fake information may lead to unwanted conflicts. Thus, having a proper fake news detection system leads to not having any of these conflicts and helps spread real news instead.

METHODOLOGY

In this project, a model is implemented to detect fake news. The first step in its development is the data collection phase in which a dataset consisting of both fake and real news is created which the help of web scraping using python. Further on preprocessing of data is done, feature selection is implemented and then classifiers like naïve bayes, random forest etc. are used on the training data to train the model. After the training is done the real data is passed through the model and the one with the most accuracy and precision is selected

FINDINGS

From this we realize fake news detection approaches in this paper are based on text analysis and use models based on speech characterization and predictive models. From our project we find out that the naïve bayes model has an accuracy of 97% in

predicting the fake news from the dataset which classifies the articles which the help of conditional probability. The combination of RNN neural network and the support vector machine has an accuracy of 99.9%.

TABLE OF CONTENTS

CHAPTER-1:

PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

Fake news is news that is false and leads to misleading information being spread. In this project we are trying to find a solution to prevent the spread of fake news and help people identify whether the news is fake or true. There have been several circumstances which has led to the rise of the spread of fake news between people such that the increased use of social media among people. Fake news leads to having large impacts on the lives of people and humans are extremely poor detectors of fake news. The objective of this project is to build a machine learning model that can differentiate between false and real news by detecting the pattern of language and sentence formation used in the text and title of the news.

1.2 Motivation for work

The motivation behind this project is to improve the intake and consumption of real news and also lead to the discarding of illegitimate news preventing the spread of rumors and helping them not to make proper decisions not based on the fake news. Another motivation is that manual classification of fake news take a lot of time and it requires lots of focus and concentration to go through various articles, this is why developing a machine learning model that can predict the news articles helps people by hosting the detector online thus allowing the access to everyone and helping them differentiate between the fake and real news.

1.3 Techniques

We have used two techniques that have been implemented in this project. That is the naïve bayes classifier which uses the multinomial model to train the predictor model. It uses conditional probability to classify the articles. The second one we have used is the recurrent neural network method which uses the back propagation method. It first takes the output of one layer and then feeds the output back into the input and then calculates the prediction of the data.

1.4 Problem Statement

Misinformation and disinformation spread within the media is becoming a serious social challenge. It is leading to the poisonous atmosphere on the web and causing riots and lynchings on the road. fake articles had outrageous headlines that were meant to draw in the greatest amount of engagement from users, also as pretending to be legitimate to gain credibility — at least at first glance. This led to exponential engagement to many users through mindless sharing on social media such as Facebook.

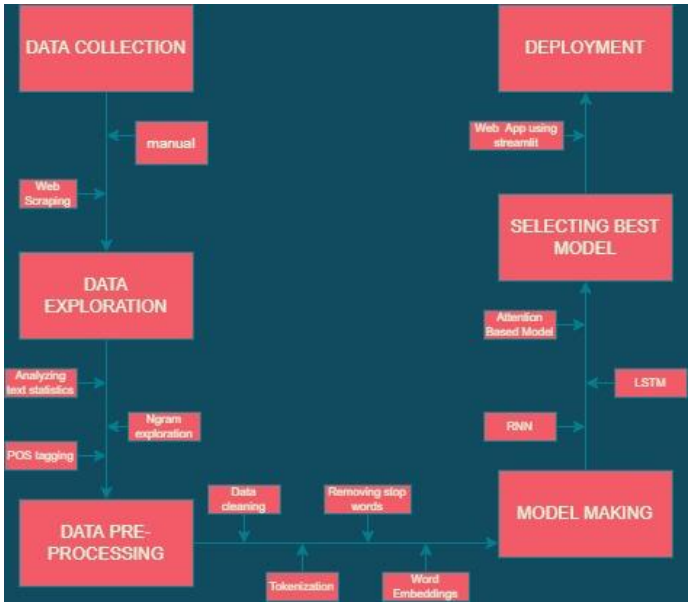
Fake News Damages: Popular Examples from India
1) GPS tracking nano chip in 2000 Rupee notes (Nov 2016).
2) Muzaffarnagar riots of 2013: fake video fueled communal passions.
3) UNESCO has declared ‘Jana Gana Mana' the best national anthem in the world (WhatsApp).
A solution to these problems is to use automated fake news detection software.

Thus, the aim of this project of this project is to develop a model that detects the language usage and sentence formation of previous fake news reports and then various machine learning techniques and approaches are used to train the developed model with the help of various training datasets. By training the model it helps the classifiers that are used in the model to automatically detect the fake news when the final dataset is passed through it.

1.5 Objective of the work

The objective of this model is to take the past news articles and use them to predict whether the test news article is real or fake. This is done with the help of various machine learning classifiers. Here the classifiers are first trained with the training data set. After it is trained the test data set is passed through and the accuracy of the model is calculated.

1.6 Organization of the project



The Organization of the project is such that first the data is collected from various websites with the help of web scrapping. Then the data goes through preprocessing which is the removal of URLs, numbers, making them lowercase etc. Further different models like LSTM, RNN, naive bayes etc are made and a training data set is passed through to select the one with the highest accuracy and then the model is deployed with the help of the python module streamlit in the form of a webapp.

1.8 Summary

Manual classification of fake news and real news will require people with great concentration and who are able to identity various text abnormalities. However, if manual classification is done it will take a lot of time to just identify one article at time. It is due to this that we have discussed the models developed in this paper. By using these models, it is possible to save more time and help people gain awareness whether the news is reliable or fake and thus it helps people make decisions based on real not fake news and help them not be manipulated.

CHAPTER-2:

RELATED WORK INVESTIGATION

2.1 Introduction

In this chapter we will be discussing about the related work investigation we have done on various other required topics which we have used to develop our model.

2.2 Core Area

The core tast of fake news detection involves the task of identifying the lanfuage i.e the set of words or languages which are used to trick the readers. The task of classifying fake news is a very challenging task as the process of manual classification of sentences is very difficult which is why we develop a machine learning model to do it.

2.3 Existing Approaches/Methods

2.3.1 Approaches/Methods -1

Used only the naive bayes classifier to detect the fake news using the bayes theorem. It was used as a software framework and it used past records from the Facebook.But it had a very low accuracy of 74%.

2.3.2 Approaches/Methods -2

Also used to naive bayes classifier to implement fake news discovery to different social media sites. They used social media sites like Facebook, Twitter as data sources for news.

2.3.3 Approaches/Methods -3

In this fake news was identified with the help of a language model due to which the accuracy of the classification is limited

2.4 Pros and Cons

2.4.1 Pros and cons of the stated Approaches/Methods -1

It was able to classify fake news properly but it ignored punctuation errors in the articles due to which it led to it having a very low accuracy score

2.4.2 Pros and cons of the stated Approaches/Methods -2

It was able to classify fake news properly that was posted on various social sites. But due to the reason that the site was not 100% credible the accuracy was very low.

2.4.3 Pros and cons of the stated Approaches/Methods -3

It was able to classify fake news properly based on only the language and the sentence formation of the news articles as it only used a language model. This is why the accuracy was even more it could have been improved if a prediction model was used.

2.5 Issues/observations from investigation

From the above investigation we came to know that we had to develop a model that not just read throught the sentence formation but also go through the pronunciation errors. And that we had do develop a prediction model instead of a language model to develop the fake news detector.The data for the dataset should be taken from a credible site.

CHAPTER-3

REQUIREMENT ARTIFACTS

3.1 Introduction

In this chapter we discuss about the various hardware and software requirements that we have used in our project.

3.2 Hardware and Software requirements

3.2.1 Hardware Requirements

Talking about the minimum Hardware Requirements which we will be using for our project are :
For this we will be using a PC with the following specifications:
→ It must have at least 16Gb of DDR4 RAM
→ It must have a NVIDIA GTX 1080 (4 GB RAM) Graphic Card
→ It must have a Intel Core i5-9300H or above Processor.
→ Hard disk: 100 GB (minimum) and above
→ We can use any Operating software like Windows, MacOS, or Linux (Ubuntu).

3.2.2 Software Requirements

Talking about the software requirements which we will be using for our project are:
→ Visual Studio Code (Visual Studio Code is a source-code editor)
→ Jupyter Labs (web-based interactive development environment)
→ PyCharm (integrated development environment used in computer programming)
→ Languages: Python and Web browsers: Chrome, Firefox
→ Microsoft Excel (For creating .csv dataset)

3.3 Specific Project requirements

3.3.1 Data requirement

For Data Requirement and extraction. We used Web Scraping with the help of the Beautiful Soup python library. The code extracts the data and the content of the webpage and stores the data extracted in a database.
Data collection is the collection of data from all relevant data from various websites and storing it in a database. Preprocessing of data includes converting all the text into lowercase, removing numbers, stop words like an, a, the, removing URLs from the dataset etc. This is usually done with the help of the NLTK module in python. The data is also normalized which is the process in which the data attributes are organized based on their entity types and helps reduce redundancy of data.

3.3.2 Functions requirement

In this the functions of the system are defined and the behavior of the model is evaluated when presented with specific inputs and various other conditions which includes data processing, manipulation. Functions that our model should be able to do:

- It should be able to read and preprocess the dataset
- Analyse the fake data and understand the pattern of the sentences
- Able to split data into training and testing sets.
- Must be able to train the model using training set
- Evaluate and validate the model using the testing set and thus being able to classify the data as real and fake.

3.4 Summary

In this Chapter we have discussed the Software and Hardware requirements which will be used in our project, then we have discussed some specific project requirements where we have discussed the Data

Requirements, Function Requirement, and Performance and security requirements. Many kinds of research show that news of different languages is a single tool.

CHAPTER-4:
DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Methodology and goal

Here a model is implemented to detect fake news. The first step in its development is the data collection phase in which a dataset consisting of both fake and real news is created which the help of web scraping using python. Further on preprocessing of data is done, feature selection is implemented and then classifiers like naïve bayes, random forest etc. are used on the training data to train the model. After the training is done the real data is passed through the model and the one with the most accuracy and precision is selected, which is the goal of the project.

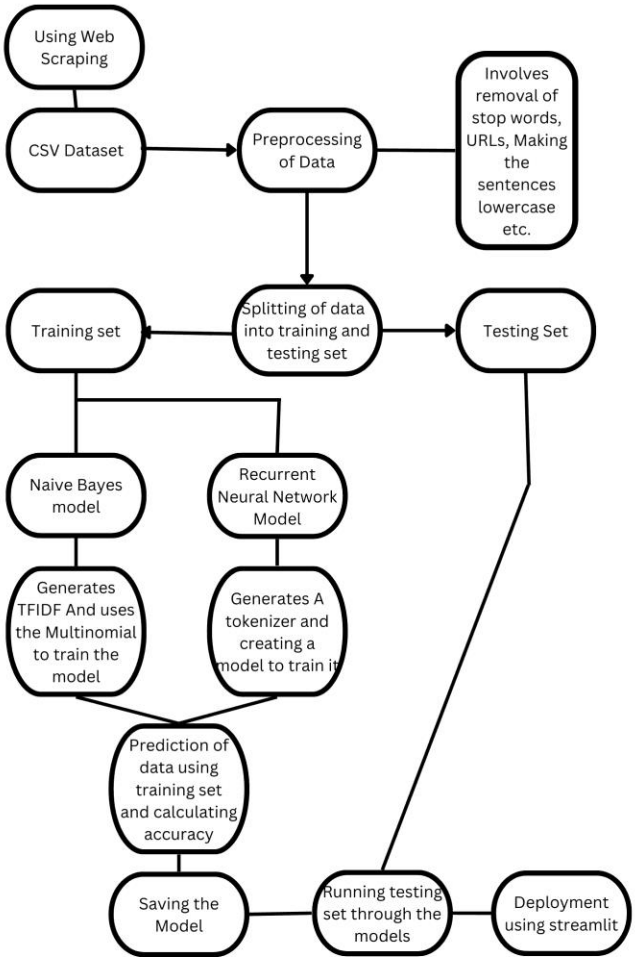
4.2 Functional modules design and analysis

Our Project is divided into 5 functional modules:

- **Data Collection -** For Data Requirement and extraction. We used Web Scraping with the help of the Beautiful Soup python library. The code extracts the data and the content of the webpage and stores the data extracted in a database.
- **Data Cleaning and Preprocessing -** Data collection is the collection of data from all relevant data from various websites and storing it in a database. Preprocessing of data includes converting all the text into lowercase, removing numbers, stop words like an, a, the, removing URLs from the dataset etc. This is usually done with the help of the NLTK module in python
- **Model Building -** The process of developing a machine learning model and training it using various training datasets is known as model building. We tried to build the machine learning model with many algorithms and choose the one with the best validation accuracy. We used libraries like sklearn and tensorflow for this.
- **Model Evaluation - It** is the use of various evaluation metrics to determine the accuracy of the prediction of the model. It is used to determine the efficiency of the model and helps monitor the model.
- **Deployment -** Deployment of a model refers to the usage of new data in the model and making sure the model gives the correct result during prediction. Here we have used streamlit to deploy the model in the form of a webapp.

4.3 Software Architectural designs

When designing the model first the dataset is created using web scraping which is then preprocessed by removing the stop words, URLS, Converting it into lowercase. Further on the data is split into a training set and testing set which is then passed through machine learning models and then the accuracy score is calculated. After that the models are saved and the testing set is passed thorugh it and the streamlit python module is used to deploy both the models as a webapp.



4.4 Novelty

The novelty of our project is that the time taken to create the dataset takes much less time as it uses a program to webscrap the data directly from the site. It has a high accuracy score and its runtime is very low. It also does not use a html file or a css file to implement a webpage so time is saved as we don't have to develop the html file.

CHAPTER-5:
TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Outline

This chapter will cover the coding of the project and the results and performance analysis of our project.

5.2 Technical coding and code solutions

```
df = pd.read_csv("C:\\Users\\Admin\\OneDrive\\Desktop\\Project\\Final DataFrame.csv") #
loaded csv(comman seperated values) files using pandas library
df.drop(['Unnamed: 0'],axis=1,inplace=True) # dropping unnamed column
df['Target'].value_counts() # 0-Fake , 1-True
def lowercase_text(title):
    return title.lower()
```

```
df["title"] = df["title"].apply(lowercase_text)
def remove_num(text):
    result = re.sub(r'\d+', '', text)
    return result

df["title"] = df["title"].apply(remove_num)
def rem_punct(title):
    translator = str.maketrans('', '', string.punctuation)
    return title.translate(translator)

df["title"] = df["title"].apply(rem_punct)
```

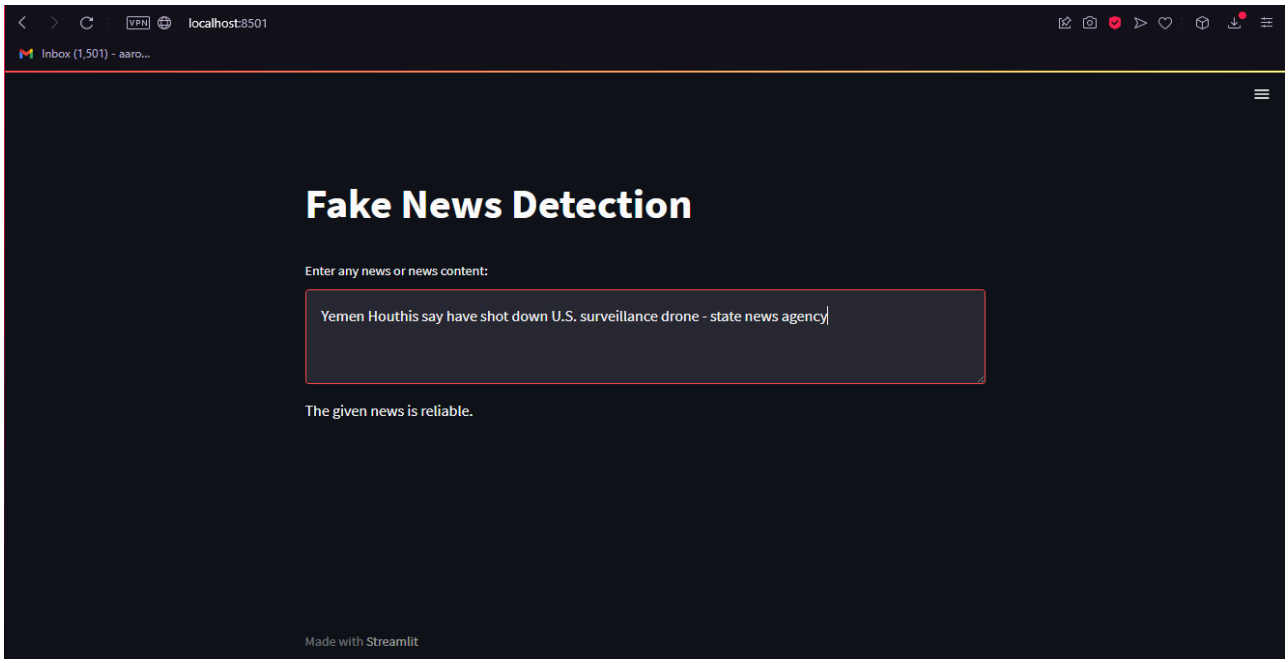
```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
# removing stop word
stop = set(stopwords.words("english"))
def remove_stopwords(text):
  filtered_words = [word.lower() for word in text.split() if word.lower() not in stop]
  return " ".join(filtered_words)
len(stop)
df["title"] = df.text.map(remove_stopwords)

def remove_html_tags(title):
    pattern = re.compile('<.*?>')  # used regular expression
    return pattern.sub(r'', title)
```

```
df['title'] = df['title'].apply(remove_html_tags)
def remove_url(title):
    pattern = re.compile(r'https?://\S+|www\.\S+') # used regular expression
    return pattern.sub(r'', title)
df['title'] = df['title'].apply(remove_url)

def counter_word(title_col):
    count = Counter()
    for title in title_col.values:
        for word in title.split():
            count[word] += 1
    return count


counter = counter_word(df.title)

num_unique_words = len(counter)
print(num_unique_words)

train_size = int(df.shape[0] * 0.9)

train_df = df[:train_size]
val_df = df[train_size:]

# split text and labels
train_sentences = train_df.title.to_numpy()
train_labels = train_df.Target.to_numpy()
val_sentences = val_df.title.to_numpy()
val_labels = val_df.Target.to_numpy()
# vectorize a text corpus by turning each text into a sequence of integers i.e assigning each
unqiue word(vocabulary) a unqiue number
tokenizer = Tokenizer(num_words=num_unique_words)
tokenizer.fit_on_texts(train_sentences) # fit only to training
# each word has unique index
word_index = tokenizer.word_index # dict- each word as key and value is unique indices

type(tokenizer)
tokenizer
```

```
len(word_index)
train_sentences[0]
train_sentences.shape
type(train_sentences)



train_sequences = tokenizer.texts_to_sequences(train_sentences)
val_sequences = tokenizer.texts_to_sequences(val_sentences)
print(train_sentences[10])
print(train_sequences[10])


import io
import json
tokenizer_json = tokenizer.to_json()
with io.open('tokenizer.json', 'w', encoding='utf-8') as f:

    f.write(json.dumps(tokenizer_json, ensure_ascii=False))
type(tokenizer_json)
max_length = 25
word_index["<PAD>"] = 0

train_padded = tf.keras.preprocessing.sequence.pad_sequences(sequences =
train_sequences,value=word_index["<PAD>"],padding="post",maxlen=max_length,truncating='post')
val_padded = tf.keras.preprocessing.sequence.pad_sequences(sequences =
val_sequences,value=word_index["<PAD>"],padding="post",maxlen=max_length,truncating='post')
train_padded.shape
val_padded.shape
model = keras.models.Sequential()
model.add(layers.Embedding(num_unique_words, 16, input_length=max_length))
"""The layer will take as input an integer matrix of size (batch, input_length),
and the largest integer (i.e. word index) in the input should be no larger than num_words
(vocabulary size).
Now model.output_shape is (None, input_length, 16), where `None` is the batch dimension.
"""
model.add(layers.SimpleRNN(32, dropout=0.9))
model.add(layers.Dense(1, activation="sigmoid"))
model.summary()
```

```
loss = keras.losses.BinaryCrossentropy(from_logits=False)
optim = keras.optimizers.Adam(lr=0.001)
metrics = ["accuracy"]

model.compile(loss=loss, optimizer=optim, metrics=metrics)



model.fit(train_padded, train_labels, epochs=5, validation_data=(val_padded, val_labels),
verbose=2)

model.save("model.h5")
predictions = model.predict(val_padded)
predictions = [1 if p > 0.5 else 0 for p in predictions]
# Check reversing the indices
# flip (key, value)
reverse_word_index = dict([(idx, word) for (word, idx) in word_index.items()])

def                                                                    decode(sequence):
    return    "    ".join([reverse_word_index.get(idx,    "?")    for    idx    in    sequence])
decoded_text                              =                          decode(train_sequences[10])
print(train_sequences[10])
print(decoded_text)
print(train_sentences[10:20])
print(train_labels[10:20])
print(predictions[10:20])
from                 sklearn.metrics                 import                 confusion_matrix
from                         sklearn                         import                         metrics
print('\t\t\t\tCLASSIFICATIION                                                    METRICS\n')
print(metrics.classification_report(predictions,                                    val_labels))
cm=metrics.confusion_matrix(val_labels,predictions)
print(cm)
```

5.3 Working Layout of Forms



In the webapp the text that needs to be checked is taken as inpput from the user.The input is then passed throught the model which predicts the data and gives the appropriate result.

5.5 Test and validation

CLASSIFICATION METRICS

	precision	recall	f1-score	support
0	0.99	1.00	0.99	2304
1	1.00	0.99	0.99	2186
accuracy			0.99	4490
macro avg	0.99	0.99	0.99	4490
weighted avg	0.99	0.99	0.99	4490

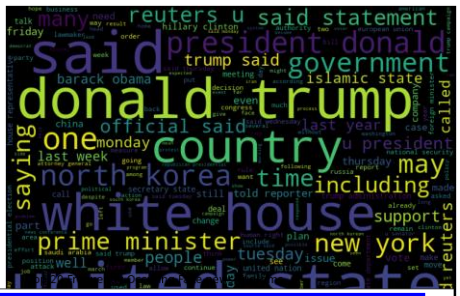
Accuracy of RNN Model = 99.9%

```
C:\Users\Admin\AppData\Loc
0.9525612472160356
```

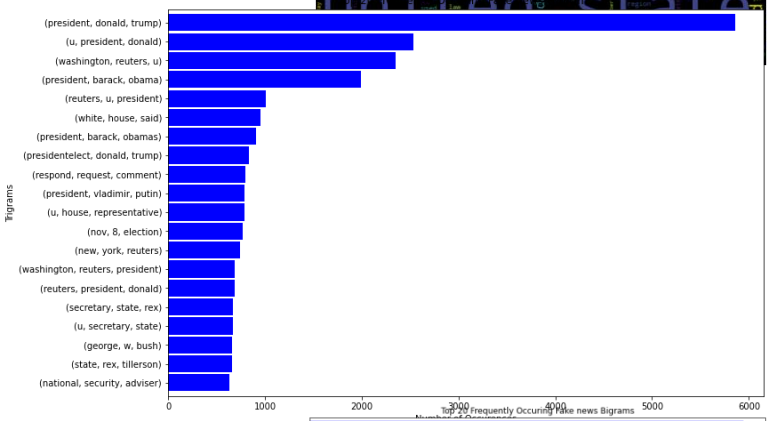
Accuracy of Naïve Bayes with Multinomial Model = 95%

5.6 Performance Analysis (Graphs/Charts)

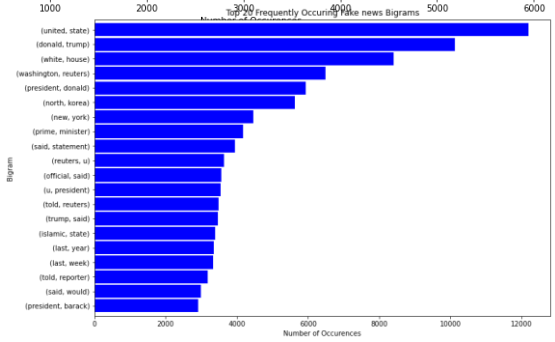
A word cloud is a collection, or cluster, of words depicted in different sizes. visualizing unstructured text data and getting insights on trends and patterns



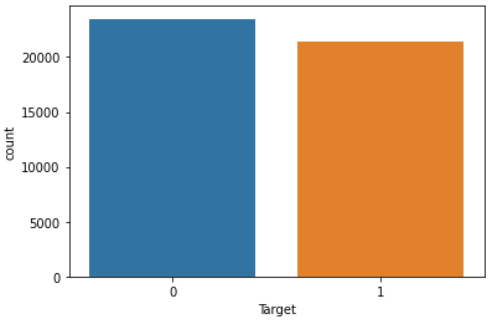
It used to show the common triplets of words that are used in fake news. The above graph shows the distribution of the triplets of word sequences in the dataset.



It is used to show the common pair of words that are used in fake news. The above graph shows the distribution of the pair of word sequences in the dataset.



The above bar graph shows the distribution of True and false data in the dataset.



Confusion matrix is used to compare the frequency of predicted classes to expected classes.Here it shows the comparion better the truth and false values

```
array([[4699,   30],
       [  25, 4226]])
```

5.7 Summary

Here we have shown our implemented code and the analysis of the performance of our model.

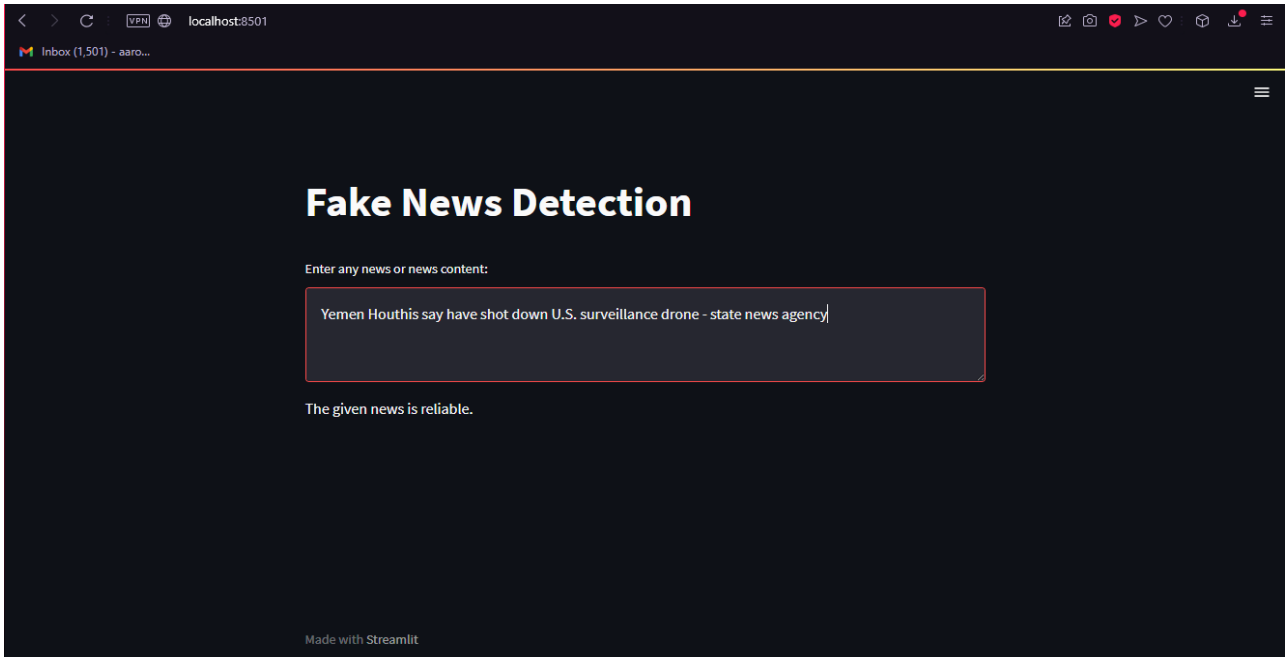
CHAPTER-6:

PROJECT OUTCOME AND APPLICABILITY

6.1 Key implementations outlines of the System

We have deployed a fake news detection model using streamlit which using recurrenet neural networks to predict whether the news is real or fake.

6.2 Significant project outcomes



In the webapp the text that needs to be checked is taken as inpput from the user.The input is then passed throught the model which predicts the data and gives the appropriate result.

6.3 Project applicability on Real-world applications

Manual classification of fake news and real news will require people with great concentration and who are able to identity various text abnormalities. However, if manual classification is done it will take a lot of time to just identify one article at time. It is due to this that we have discussed the models developed in this paper. By using these models, it is possible to save more time and help people gain awareness whether the news is reliable or fake and thus it helps people make decisions based on real not fake news and help them not be manipulated.

CHAPTER-7:

CONCLUSIONS AND RECOMMENDATION

7.1 Outline

In this chapter we discuss about the various constraints and limitations of our model and also the fute enhancements that are possible.

7.2 Limitation/Constraints of the System

The first model has a lower accuracy than the RNN model and the dataset has to be kept on being updated with the latest fake and real news.

7.3 Future Enhancements

This may include combining various machine learning models together to get a higher accuracy score. And also finding out a way to keep on updating the dataset.

7.4 Inference

From the project we have seen that there are various ways to develop a machine learning model and there exist numerous classifiers that are used in the model making and training process.

REFERENCES

1. Gilda, S. “Evaluating machine learning algorithms for fake news detection.” 15th Student Conference on Research and Development (SCOReD) (pp. 110-115). IEEE. 2017.

2. Yumeng Qin et al. “Predicting Future Rumours.” Chinese Journal of Electronics (Volume: 27 , Issue: 3 , 5 2018, 514 - 520.

3. Supanya Aphiwongsophon et al. “ Detecting Fake News with Machine Learning Method.” 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). Chiang Rai, Thailand, Thailand: IEEE . 2018.

4. ArushiGupta and RishabhKaushal. “Improving spam detection in Online Social Networks.” International Conference on Cognitive Computing and Information Processing (CCIP). Semanticscholar.org.2015.

5. Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., & Liu, Y. (2019). Combating fake news: A survey on identification and mitigation techniques. ACM Transactions on Intelligent Systems and Technology (TIST), 10(3), 1-42.

6. E. C. Tandoc Jr et al. “Defining fake news a typology of scholarly definitions”. Digital Journalism , 1–17. 2017