# A Taxonomy Of Fake News Classification Techniques :

## Survey and Implementation of Aspects .

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING ( CSE )**


Submitted by

**ABDUL RAHMAN – 22081A0506**

**AHMED ABDUL AZIZ – 22081A0512**

**ANSAR PATEL – 22081A0517**

**MOHAMMED ABDUL WAHED YOSUF ZAI – 22081A0560**

**CSE ( 2ND YEAR – 2ND SEMESTER )**

**SHADAN COLLEGE OF ENGINEERING & TECHNOLOGY**

**(A NAAC A+ & NBA Accredited and ISO 9001:2015 certified institution)**

**(AN AUTONOMOUS INSTITUTION)**

**PEERANCHERU, HYDERABAD-500086**

**(Affiliated to JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD)**

**HYDERABAD-500087**

**(2023-2024)**

# A Taxonomy Of Fake News Classification Techniques : Survery And Implementation Aspects

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING (CSE)**


**UNDER THE GUIDANCE**

**OF**

**Dr. G SRIDHAR**

Submitted by –

**ABDUL RAHMAN – 22081A0506**

**AHMED ABDUL AZIZ – 22081A0512**

**ANSAR PATEL – 22081A0517**

**MOHAMMED ABDUL WAHED YOSUF ZAI – 22081A0560**

**CSE ( 2ND YEAR – 2ND SEMESTER ) (2023-2024)**

**DEPARTMENT OF COMPUTER SCIENCE**

**SHADAN COLLEGE OF ENGINEERING & TECHNOLOGY**

**(A NAAC A+ & NBA Accredited and ISO 9001:2015 certified institution)**

**(AN AUTONOMOUS INSTITUTION)**

**PEERANCHERU, HYDERABAD-500086**

**(Affiliated to JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD)**

**HYDERABAD-500087**

Ref. No.: Date: _____

# CERTIFICATE

This is to certify that the dissertation entitled "**A TAXOMONY OF FAKE NEWS CLASSIFICATION TECHNIQUES : SURVEY AND IMPLEMENTATION ASPECTS** " is a bonafide work done and

The results embodied in this report have not been submitted to any other University for the award of any degree.

Submitted by

**ABDUL RAHMAN - 22081A0506**

**AHMED ABDUL AZIZ - 22081A0512**

**ANSAR PATEL – 22081A0517**

**MOHAMMED ABDUL WAHED YOSUF ZAI – 22081A0560**

**INTERNAL EXAMINER** **HEAD OF THE DEPARTMENT**

**PRINCIPAL**

**DECLARATION :**

We hereby declare that the Major Project Report entitled "A Taxonomy of Fake News Classification techniques : Survey And Implementation Aspects" is being submitted by

**ABDUL RAHMAN** (22081A05**06**) ,

**AHMED ABDUL AZIZ** (22081A05**12**) ,

**ANSAR PATEL** (22081A05**17**) and

**MOHAMMED ABDUL WAHED YOSUF ZAI** (22081A0**60**)

in **II** B.Tech **II** Semester Computer Science & Engineering(**CSE**)

is a record bona-fide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of any degree.

| | | |
|---|---|---|
| ABDUL RAHMAN | - | 22081A0506 |
| AHMED ABDUL AZIZ | - | 22081A012 |
| ANSAR PATEL | - | 22081A0517 |
| MOHAMMED ABDUL WAHED YOSUF ZAI | - | 22081A0560 |

DATE :

# ACKNOWLEDGEMENT

I thank GOD almighty for guiding us throughout the project.

I would like to thank our parents without whose blessings, we would not have been able to accomplish our goal.

I would like to thank our internal guide Mr. **Dr. G SRIDHAR**, for all help and support he extended to us.

I am extremely grateful to **Dr. G. SRIDHAR**, HEAD OF THE DEPARTMENT OF **CSE & IT**, for providing us with Best faculties and the atmosphere for the creative work, guidance and encouragement.

I would thank **Dr. ATEEQ UR RAHMAN,** PRINCIPAL and all staff members of our college and friends for extending their cooperation during our project.

I would also like to thank all those who have contributed to the completion of the project and help us with valuable suggestions and improvement.

# CONTENTS

## ABSTRACT

In the present era, social media platforms such as Facebook, WhatsApp, Twitter, and Telegram are significant sources of information distribution, and people believe it without knowing their origin and genuineness. Social media has fascinated people worldwide in spreading fake news due to its easy availability, cost-effectiveness, and ease of information sharing. Fake news can be generated to mislead the community for personal or commercial gains. It can also be used for other personal benefits such as defaming eminent personalities, amendment of government policies, etc. Thus, to mitigate the awful consequences of fake news, several research types have been conducted for its detection with high accuracy to prevent its fatal outcome. Motivated by the aforementioned concerns, we present a comprehensive survey of the existing fake news identification techniques in this paper. Then, we select Machine Learning (ML) models such as Long-Short Term Memory (LSTM), Passive Aggressive Algorithm, Random Forest (RF), and Naive Bayes (NB) and train them to detect fake news articles on the self-aggregated dataset. Later, we implemented these models by hyper tuning various parameters such as smoothing, drop out factor, and batch size, which has shown promising results in accuracy and other evaluation metrics such as F1-score, recall,

precision, and AUC score. The model is trained on 6335 news articles, with LSTM showing the highest accuracy of 92.34% in predicting fake news and NB were showing the highest recall. Based on these results, we propose a hybrid fake news detection technique using NB and LSTM.

## LIST OF FIGURES

# 1.INTRODUCTION

## 1.1 Problem statement:

Intentionally deceptive content presented under the guise of legitimate journalism (or 'fake news,' as it is commonly known) is a worldwide information accuracy and integrity problem that affects opinion forming, decision making, and voting patterns. Most fake news is initially distributed over social media conduits like Facebook and Twitter and later finds its way onto mainstream media platforms such as traditional television and radio news. The fake news stories that are initially seeded over social media platforms share key linguistic characteristics such as excessive use of unsubstantiated hyperbole and non-attributed quoted content. The results of a fake news identification study that documents the performance of a fake news classifier are presented and discussed in this paper.

## 1.2 Motivation:

In this paper, the research process, technical analysis, technical linguistics work, and classifier performance and results are presented. The paper concludes with a discussion of how the current system will evolve into an influence mining system. The fake news stories that are initially seeded over social media platforms share key linguistic characteristics such as excessive use of unsubstantiated hyperbole and non-attributed quoted content. The results of a fake news identification study that documents the performance of a fake news classifier are presented and discussed in this paper.

## 1.3 Objective:

Fake news has been demonstrated to be problematic in multiple ways. It has been shown to have real influence on public perception and the ability to shape regional and national dialogue . It has harmed businesses and individuals and even resulted in death, when an individual responded to a hoax . It has caused some teenagers to reject the concept of media objectivity and many students can't reliably tell the difference between real and faked articles . It is even thought to have influenced the 2016 United States elections . Fake news can be spread deliberately by humans or indiscriminately by bot armies , with the latter giving a nefarious article

significant reach. Not just articles are faked, in many cases fake, mislabeled or deceptive images are also used to maximize impact . Some contend that fake news is a “plague” on society’s digital infrastructure . Many are working to combat it. Farajtabar, et al. , for example, has proposed a system based on points, while Haigh, Haigh and Kozakhave suggested the use of “peer-topeer counter propaganda

### 1.4 Proposed System:

In this paper author is describing concept to detect fake news from social media or document corpus using Natural Language Processing and attribution supervised learning estimator. News documents or articles will be uploaded to application and then by using Natural Language Processing to extract quotes, verbs and name entity recognition (extracting organizations or person names) from documents to compute score, verbs, quotes and name entity also called as attribution. Using supervised learning estimator we will calculate score between sum of verbs, sum of name entity and sum of quotes divided by total sentence length. If score greater than 0 then news will be consider as REAL and if less than 0 then new will be consider as FAKE.

### 1.5 Advantages of proposed system:

- It is desirable to use COX data for phylogenetic exploration.
- We use the data of COX experimental values.
- Security

## 2.TECHNOLOGIES LEARNT

### 2.1 Python Development Steps:

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
- Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

  Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors.

## 2.2 Modules Used in Project :

**Tensorflow:**

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a highperformance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

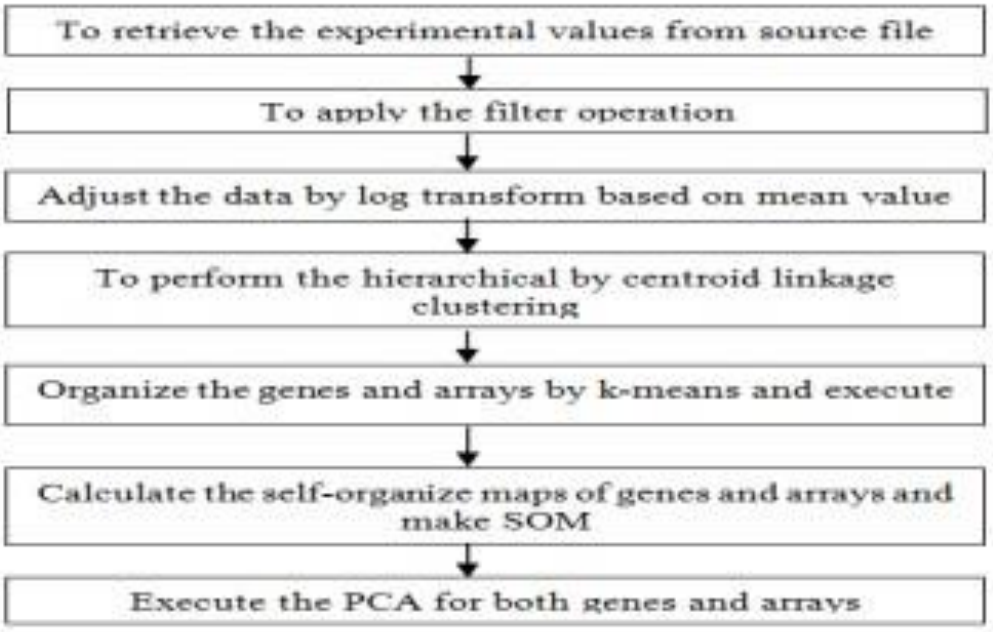
Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits.Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

## 3. SYSTEM DESIGN

### 3.1 System Architecture:




### 3.2 Module description:

**Tensorflow**

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities Besides its obvious scientific uses, Numpy can also be used as an efficient

multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics,

Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits.Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. The library is

built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

- **NumPy**: Base n-dimensional array package
- **SciPy**: Fundamental library for scientific computing
- **Matplotlib**: Comprehensive 2D/3D plotting
- **IPython**: Enhanced interactive console
- **Sympy**: Symbolic mathematics
- **Pandas**: Data structures and analysis
- Extensions or modules for SciPy care conventionally named SciKits. As such, the module

## 3.3 System Specification:

### 3.3.1 Software Requirements

Functional requirements for a secure cloud storage service are straightforward:

1. The service should be able to store the user'sdata;
2. The data should be accessible through any devices connected to theInternet;
3. The service should be capable to synchronize the user's data between multiple devices (notebooks, smart phones,etc.);
4. The service should preserve all historical changes(versioning);
5. Data should be shareable with otherusers;
6. The service should support SSO;and
7. The service should be interoperable with other cloud storage services, enabling data migration from one CSP toanother.

• **Operating System:**Windows • **Coding Language**: Python 3.7

• **Script:**

• **Database :**

### 3.3.2 Hardware Requirements:

• **Processor** - Pentium–III

• **Speed** – 2.4GHz

• **RAM** - 512 MB(min)

• **Hard Disk** - 20 GB

• **Floppy Drive** - 1.44MB

• **Key Board** - Standard Keyboard

• **Monitor** – 15 VGAColour

Cloud computing has three fundamental models, these are:

### 3.4 Detailed Design

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on **diagrammatic representations** of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningfulmodels.
2. Provide extendibility and specialization mechanisms to extend the coreconcepts.

3. Be independent of particular programming languages and developmentprocess.
4. Provide a formal basis for understanding the modelinglanguage.
5. Encourage the growth of OO toolsmarket.

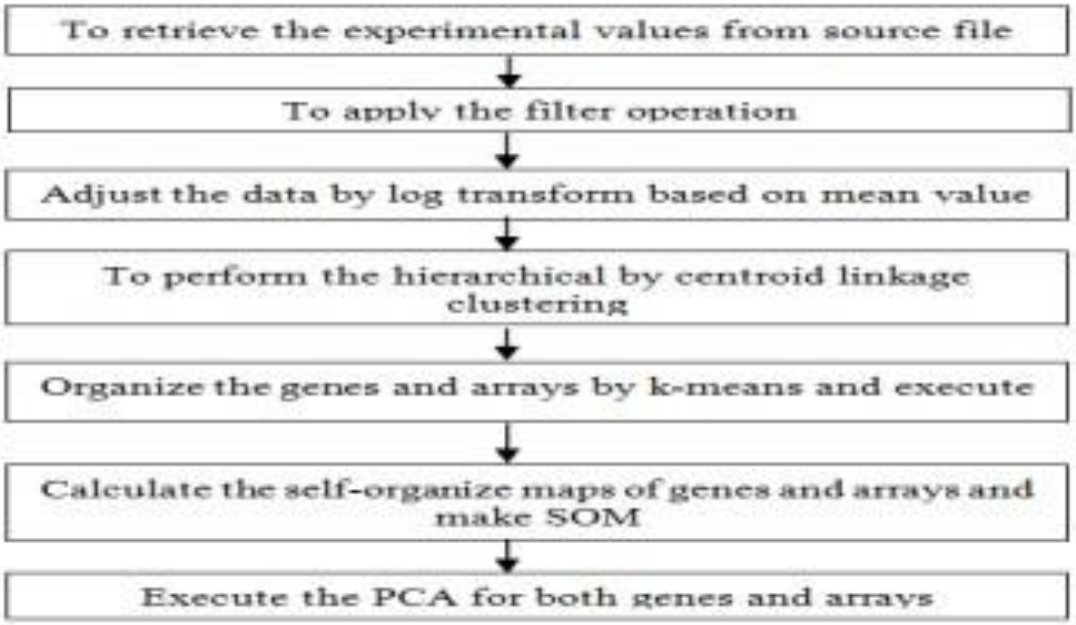
6 Support higher level development concepts such as collaborations, frameworks, patterns and components.

7 Integrate best practices.

### 3.5 Algorithms used in this project :-

**CLUSTERING ALGORITHM :**

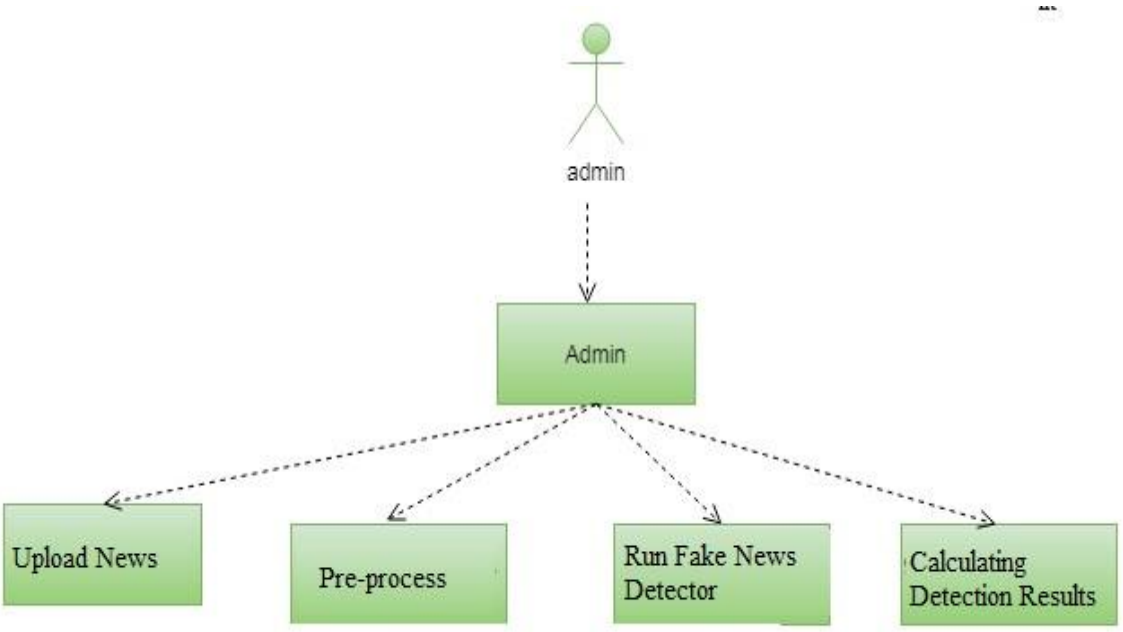
- Clustering is an unendorsed culture algorithm that finds the concealed arrangement in the unlabeled data. In this work, we used the filter the values, adjust the data values, then apply the hierarchical method, k-means algorithm, self- organizing maps (SOM), and finally apply the Principal Component Analysis (PCA) for avoid the unwanted values, adjust the data with help of log transform, for clustering genes and arrays with hierarchical clustering by centroid linkage.


To retrieve the experimental values from source file
To apply the filter operation
Adjust the data by log transform based on mean value
To perform the hierarchical by centroid linkage clustering
Organize the genes and arrays by k-means and execute
Calculate the self-organize maps of genes and arrays and make SOM
Execute the PCA for both genes and arrays

### 3.6 Diagrams:

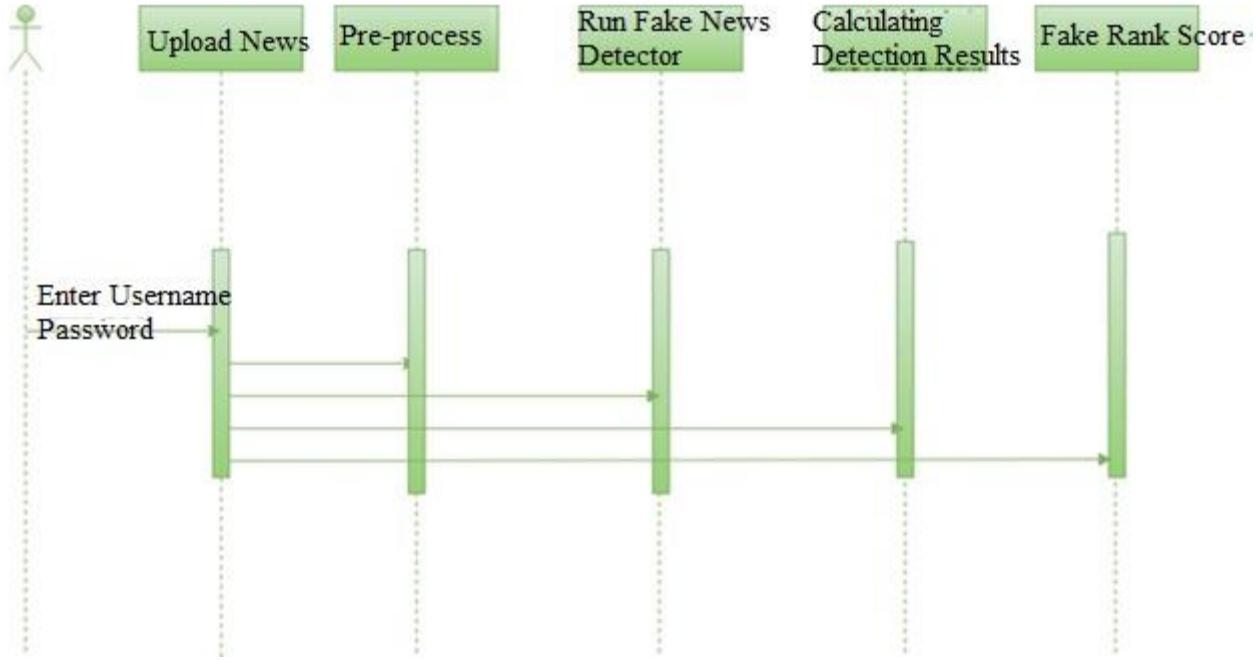
**Use Case Diagram:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



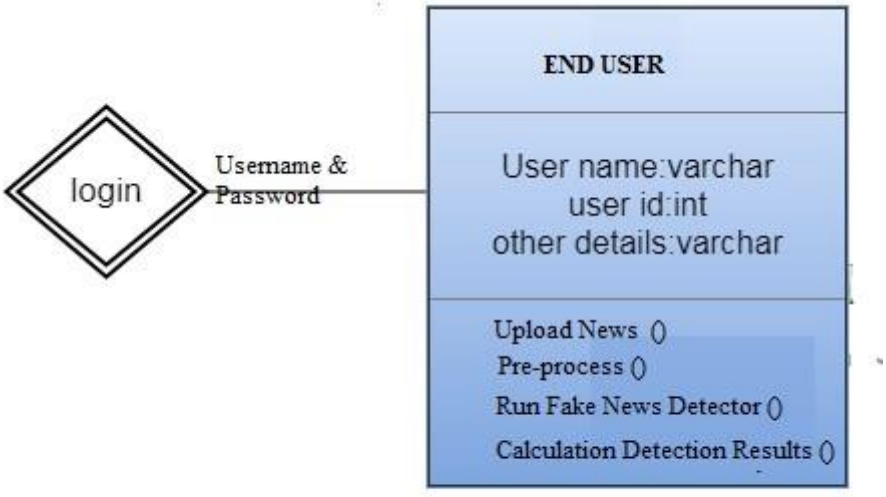

**Sequence Diagram:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



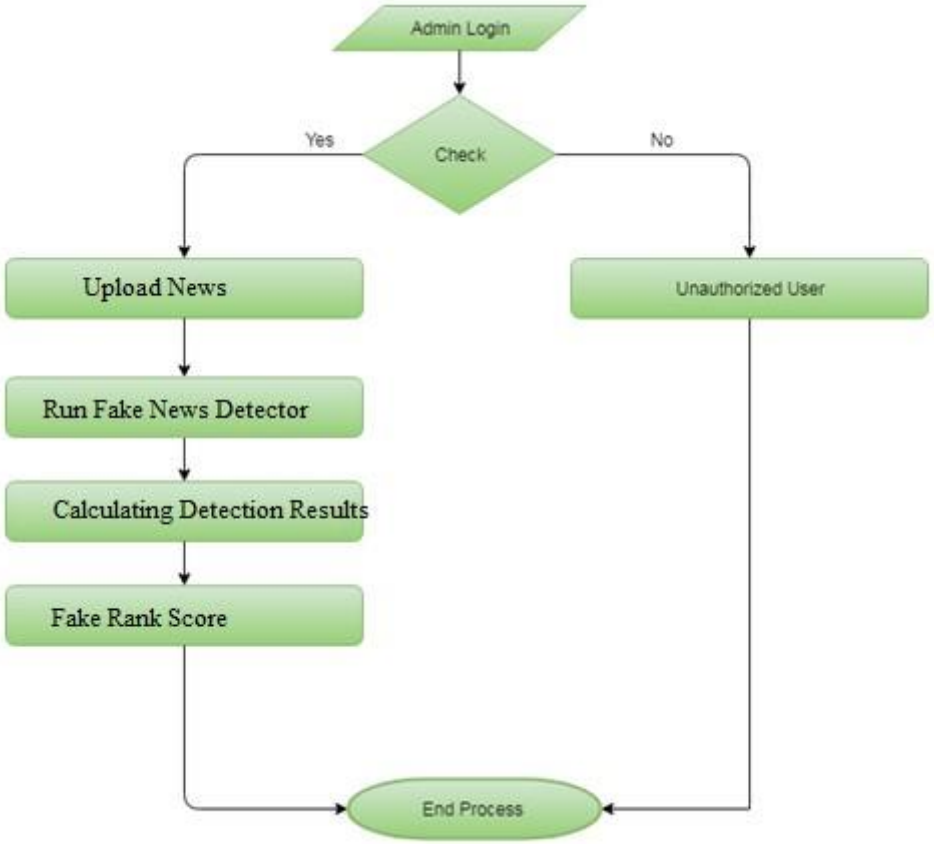

**Class Diagram:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.




**Data Flow diagram :**

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.. DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:



**Component Diagram :**

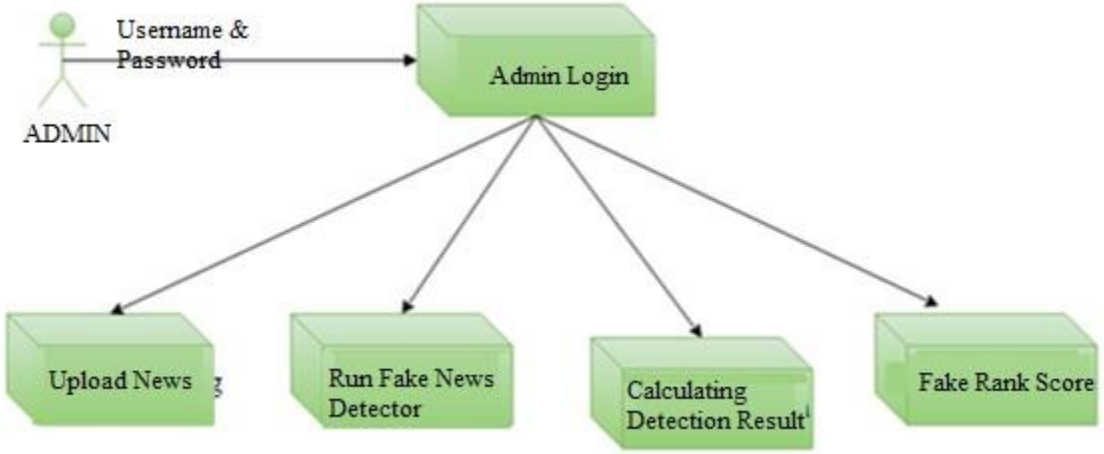
Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

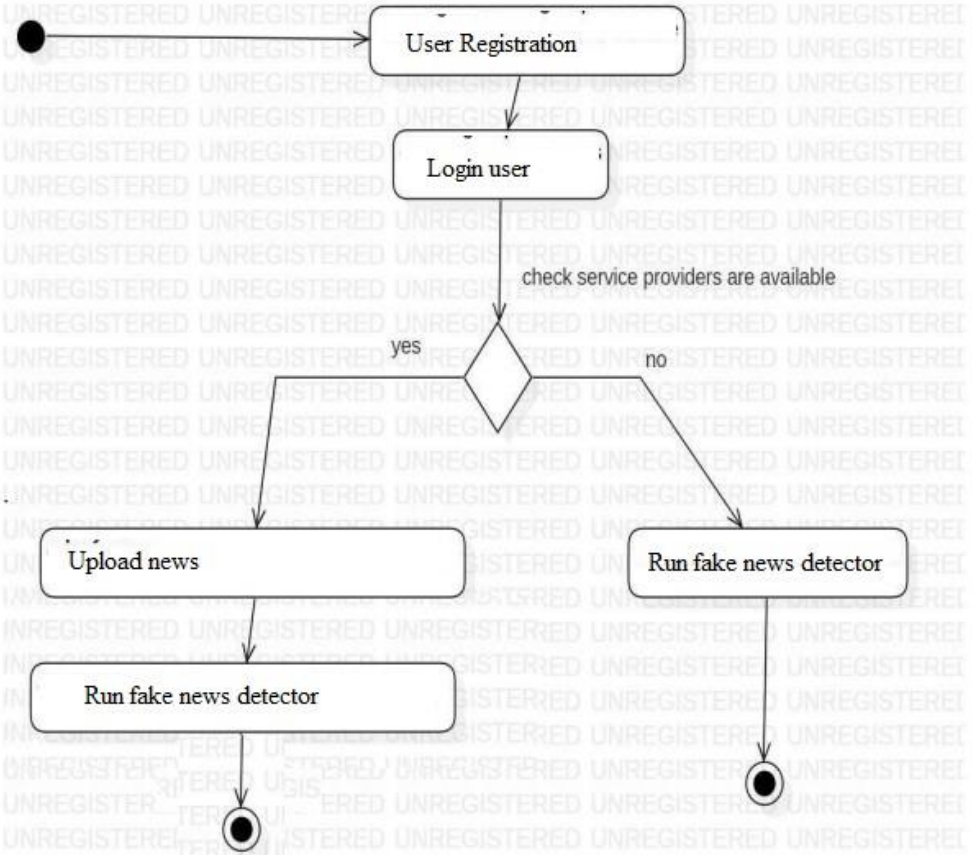
A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

UML Component diagrams are used in modeling the physical aspects of objectoriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.



**Activity Diagram:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



## 4 .IMPLEMENTATION

```
from __future__ import absolute_import
from __future__ import division from
__future__ import print_function

importargparse import
collections fromdatetime
import datetime
importhashlib importos.path
import random import re
import sys importtarfile

importnumpy as np fromsix.moves
import urllib importtensorflow as
tf

fromtensorflow.python.framework import graph_util
fromtensorflow.python.framework import tensor_shape
fromtensorflow.python.platform import gfile
fromtensorflow.python.util import compat

FLAGS = None
MAX_NUM_IMAGES_PER_CLASS = 2 ** 27 - 1  # ~134M


defcreate_image_lists(image_dir, testing_percentage, validation_percentage): if
not gfile.Exists(image_dir):
tf.logging.error("Image directory '" + image_dir + "' not found.") return
None
result = collections.OrderedDict() sub_dirs
= [ os.path.join(image_dir,item) for item
in gfile.ListDirectory(image_dir)]
```

```
sub_dirs = sorted(item for item in sub_dirs
ifgfile.IsDirectory(item)) forsub_dir in
sub_dirs:
extensions = ['jpg', 'jpeg', 'JPG', 'JPEG'] file_list
= []
dir_name = os.path.basename(sub_dir) ifdir_name
== image_dir:
continue tf.logging.info("Looking for images in '" +
dir_name + "'") for extension in extensions:
file_glob = os.path.join(image_dir, dir_name, '*.' + extension)
file_list.extend(gfile.Glob(file_glob)) if not file_list:
tf.logging.warning('No files found')
continue iflen(file_list) < 20:
tf.logging.warning(
    'WARNING: Folder has less than 20 images, which may cause issues.')
eliflen(file_list) > MAX_NUM_IMAGES_PER_CLASS: tf.logging.warning(
    'WARNING: Folder {} has more than {} images. Some images will '
'never be selected.'.format(dir_name, MAX_NUM_IMAGES_PER_CLASS))
label_name = re.sub(r'[^a-z0-9]+', ' ', dir_name.lower()) training_images = []
testing_images = [] validation_images = [] forfile_name in file_list:
base_name = os.path.basename(file_name) hash_name =
re.sub(r'_nohash_.*$', '', file_name) hash_name_hashed =
hashlib.sha1(compat.as_bytes(hash_name)).hexdigest() percentage_hash =
((int(hash_name_hashed, 16) %
(MAX_NUM_IMAGES_PER_CLASS + 1)) *                    (100.0 /
MAX_NUM_IMAGES_PER_CLASS))
ifpercentage_hash<validation_percentage:
validation_images.append(base_name) elifpercentage_hash<
(testing_percentage + validation_percentage):
testing_images.append(base_name) else:
training_images.append(base_name) result[label_name]
= {
    'dir': dir_name,
    'training': training_images,
```

```
        'testing': testing_images,
        'validation': validation_images,
    }
return result defget_image_path(image_lists, label_name, index,
image_dir, category): iflabel_name not in image_lists:
tf.logging.fatal('Label does not exist %s.', label_name)
label_lists = image_lists[label_name] if category not
in label_lists:
tf.logging.fatal('Category does not exist %s.', category)
category_list = label_lists[category] if not category_list:
tf.logging.fatal('Label %s has no images in the category %s.',
label_name, category) mod_index = index %
len(category_list) base_name = category_list[mod_index]
sub_dir = label_lists['dir'] full_path = os.path.join(image_dir,
sub_dir, base_name) returnfull_path


defget_bottleneck_path(image_lists, label_name, index, bottleneck_dir, category,
architecture):
returnget_image_path(image_lists, label_name, index, bottleneck_dir,
category) + '_' + architecture + '.txt'
defcreate_model_graph(model_info): withtf.Graph().as_default() as
graph:
model_path = os.path.join(FLAGS.model_dir, model_info['model_file_name'])
withgfile.FastGFile(model_path, 'rb') as f:
graph_def = tf.GraphDef() graph_def.ParseFromString(f.read())
bottleneck_tensor, resized_input_tensor = (tf.import_graph_def(
graph_def, name='', return_elements=[
model_info['bottleneck_tensor_name'],
model_info['resized_input_tensor_name'],
        ]))
return graph, bottleneck_tensor, resized_input_tensor
defrun_bottleneck_on_image(sess, image_data, image_data_tensor,
decoded_image_tensor, resized_input_tensor, bottleneck_tensor):
```

```
resized_input_values = sess.run(decoded_image_tensor,
{image_data_tensor: image_data}) bottleneck_values =
sess.run(bottleneck_tensor,
                    {resized_input_tensor: resized_input_values})
bottleneck_values = np.squeeze(bottleneck_values)
returnbottleneck_values defmaybe_download_and_extract(data_url):
dest_directory = FLAGS.model_dir if not
os.path.exists(dest_directory): os.makedirs(dest_directory) filename =
data_url.split('/')[-1] filepath = os.path.join(dest_directory, filename)
return (flip_left_right or (random_crop != 0) or (random_scale != 0) or
    (random_brightness != 0))
defadd_input_distortions(flip_left_right, random_crop, random_scale,
random_brightness, input_width, input_height, input_depth,
input_mean, input_std):
jpeg_data = tf.placeholder(tf.string, name='DistortJPGInput')
decoded_image = tf.image.decode_jpeg(jpeg_data, channels=input_depth)
decoded_image_as_float = tf.cast(decoded_image, dtype=tf.float32)
decoded_image_4d = tf.expand_dims(decoded_image_as_float, 0)
margin_scale = 1.0 + (random_crop / 100.0) resize_scale = 1.0 +
(random_scale / 100.0) margin_scale_value = tf.constant(margin_scale)
resize_scale_value = tf.random_uniform(tensor_shape.scalar(), minval=1.0,
maxval=resize_scale) scale_value = tf.multiply(margin_scale_value,
resize_scale_value) precrop_width = tf.multiply(scale_value, input_width)
precrop_height = tf.multiply(scale_value, input_height) precrop_shape =
tf.stack([precrop_height, precrop_width]) precrop_shape_as_int =
tf.cast(precrop_shape, dtype=tf.int32) precropped_image =
tf.image.resize_bilinear(decoded_image_4d, precrop_shape_as_int)
precropped_image_3d = tf.squeeze(precropped_image, squeeze_dims=[0])
cropped_image = tf.random_crop(precropped_image_3d,
                    [input_height, input_width, input_depth]) ifflip_left_right:
flipped_image = tf.image.random_flip_left_right(cropped_image) else:
flipped_image = cropped_image brightness_min = 1.0 -
(random_brightness / 100.0) brightness_max = 1.0 +
(random_brightness / 100.0) brightness_value =
```

tf.random_uniform(tensor_shape.scalar(),
minval=brightness_min, maxval=brightness_max)
brightened_image = tf.multiply(flipped_image, brightness_value)
offset_image = tf.subtract(brightened_image, input_mean)
mul_image = tf.multiply(offset_image, 1.0 / input_std)
distort_result = tf.expand_dims(mul_image, 0,
name='DistortResult') returnjpeg_data, distort_result
defvariable_summaries(var): withtf.name_scope('summaries'):
mean = tf.reduce_mean(var) tf.summary.scalar('mean', mean)
withtf.name_scope('stddev'):
stddev = tf.sqrt(tf.reduce_mean(tf.square(var - mean)))
tf.summary.scalar('stddev', stddev) tf.summary.scalar('max',
tf.reduce_max(var)) tf.summary.scalar('min', tf.reduce_min(var))
tf.summary.histogram('histogram', var)
defadd_final_training_ops(class_count, final_tensor_name, bottleneck_tensor,
bottleneck_tensor_size): withtf.name_scope('input'):
bottleneck_input = tf.placeholder_with_default(

but found '%s' for architecture '%s'""",
size_string, architecture) return None
iflen(parts) == 3: is_quantized = False
else: if parts[3] != 'quantized':
tf.logging.error(
"Couldn't understand architecture suffix '%s' for '%s'", parts[3],
architecture) return None is_quantized = True data_url =
'http://download.tensorflow.org/models/mobilenet_v1_' data_url +=
version_string + '_' + size_string + '_frozen.tgz'
bottleneck_tensor_name = 'MobilenetV1/Predictions/Reshape:0'
bottleneck_tensor_size = 1001
input_width = int(size_string) input_height
= int(size_string) input_depth = 3
resized_input_tensor_name = 'input:0'
ifis_quantized:
model_base_name = 'quantized_graph.pb' else:

model_base_name = 'frozen_graph.pb' model_dir_name =
'mobilenet_v1_' + version_string + '_' + size_string model_file_name
= os.path.join(model_dir_name, model_base_name) input_mean =
127.5 input_std = 127.5 else:
tf.logging.error("Couldn't understand architecture name '%s'", architecture)
raiseValueError('Unknown architecture', architecture)

```
return {
    'data_url': data_url,
    'bottleneck_tensor_name': bottleneck_tensor_name,
    'bottleneck_tensor_size': bottleneck_tensor_size,
    'input_width': input_width,
    'input_height': input_height,
    'input_depth': input_depth,
    'resized_input_tensor_name': resized_input_tensor_name,
    'model_file_name': model_file_name,
    'input_mean': input_mean,
    'input_std': input_std,
  }
```

defadd_jpeg_decoding(input_width, input_height, input_depth, input_mean,
input_std):
jpeg_data = tf.placeholder(tf.string, name='DecodeJPGInput') decoded_image
= tf.image.decode_jpeg(jpeg_data, channels=input_depth)
decoded_image_as_float = tf.cast(decoded_image, dtype=tf.float32)
decoded_image_4d = tf.expand_dims(decoded_image_as_float, 0)
resize_shape = tf.stack([input_height, input_width]) resize_shape_as_int =
tf.cast(resize_shape, dtype=tf.int32) resized_image =
tf.image.resize_bilinear(decoded_image_4d, resize_shape_as_int)
offset_image = tf.subtract(resized_image, input_mean) mul_image =
tf.multiply(offset_image, 1.0 / input_std) returnjpeg_data, mul_image

def main(_):

```
  # Needed to make sure the logging output is visible.   # See
https://github.com/tensorflow/tensorflow/issues/3047
tf.logging.set_verbosity(tf.logging.INFO)

  # Prepare necessary directories  that can be used during training
prepare_file_system()

  # Gather information about the model architecture we'll be using.
model_info = create_model_info(FLAGS.architecture) if
not model_info:
tf.logging.error('Did not recognize architecture flag') return
-1
  # Set up the pre-trained graph.
maybe_download_and_extract(model_info['data_url']) graph,
bottleneck_tensor, resized_image_tensor = (
create_model_graph(model_info))

  # Look at the folder structure, and create lists of all the images.
image_lists = create_image_lists(FLAGS.image_dir, FLAGS.testing_percentage,
FLAGS.validation_percentage) class_count = len(image_lists.keys())
ifclass_count == 0:
tf.logging.error('No valid folders of images found at ' + FLAGS.image_dir) return
-1
ifclass_count == 1: tf.logging.error('Only one valid folder of
images found at ' +
FLAGS.image_dir +
                ' - multiple classes are needed for classification.') return
-1
  # See if the command-line flags mean we're applying any distortions.
do_distort_images = should_distort_images(
FLAGS.flip_left_right, FLAGS.random_crop, FLAGS.random_scale,
FLAGS.random_brightness) withtf.Session(graph=graph) as sess:
# Set up the image decoding sub-graph.
```

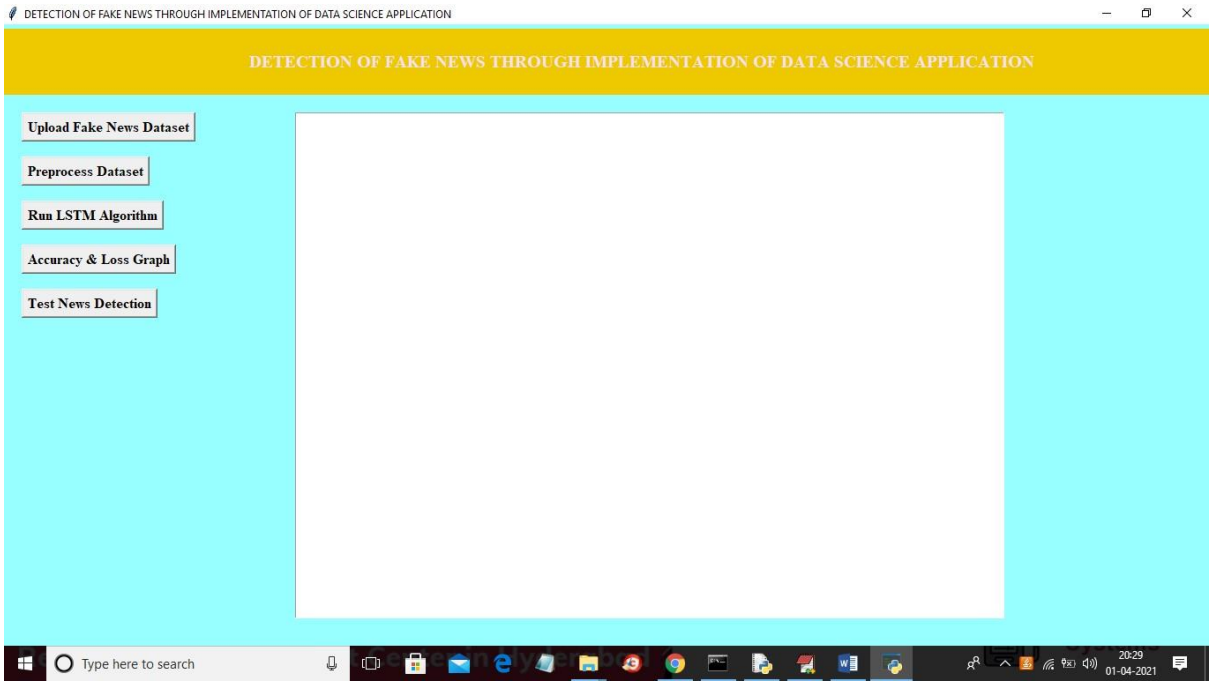
jpeg_data_tensor, decoded_image_tensor = add_jpeg_decoding(
model_info['input_width'], model_info['input_height'],
model_info['input_depth'], model_info['input_mean'],
model_info['input_std'])

ifdo_distort_images:
    # We will be applying distortions, so setup the operations we'll need.
    (distorted_jpeg_data_tensor, distorted_image_tensor)
= add_input_distortions(
FLAGS.flip_left_right, FLAGS.random_crop, FLAGS.random_scale,
FLAGS.random_brightness, model_info['input_width'],
model_info['input_height'], model_info['input_depth'],
model_info['input_mean'], model_info['input_std'])

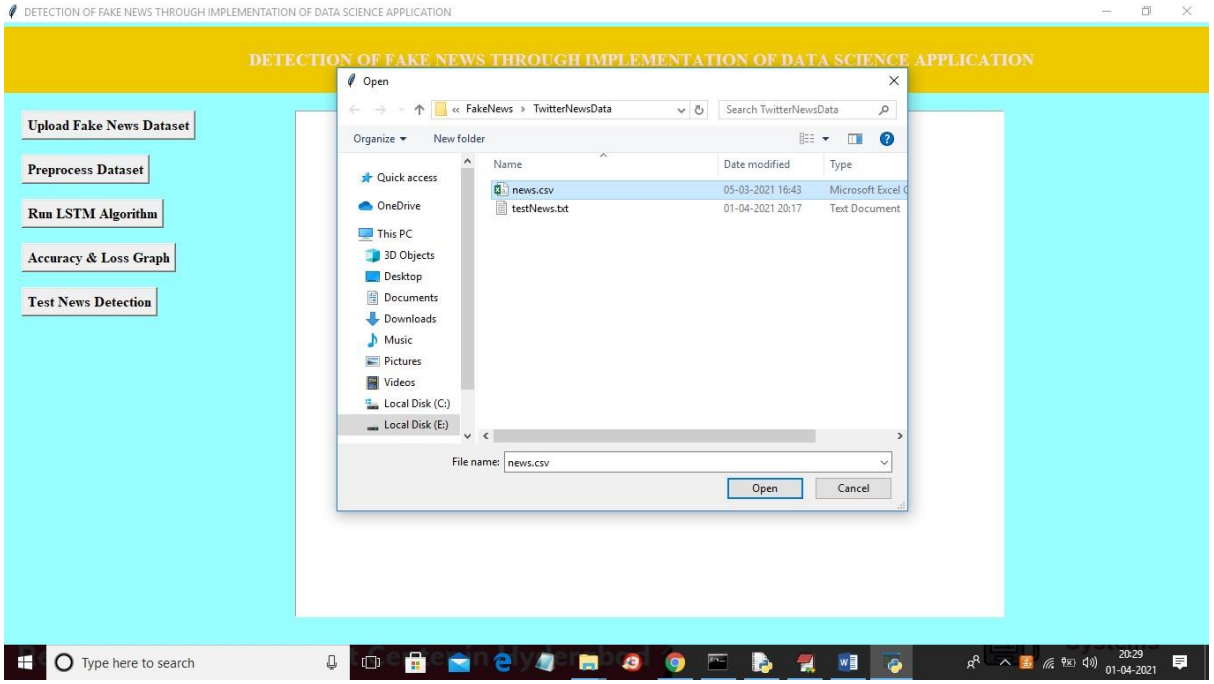
## 5.RESULTS

### 5.1 Execution:

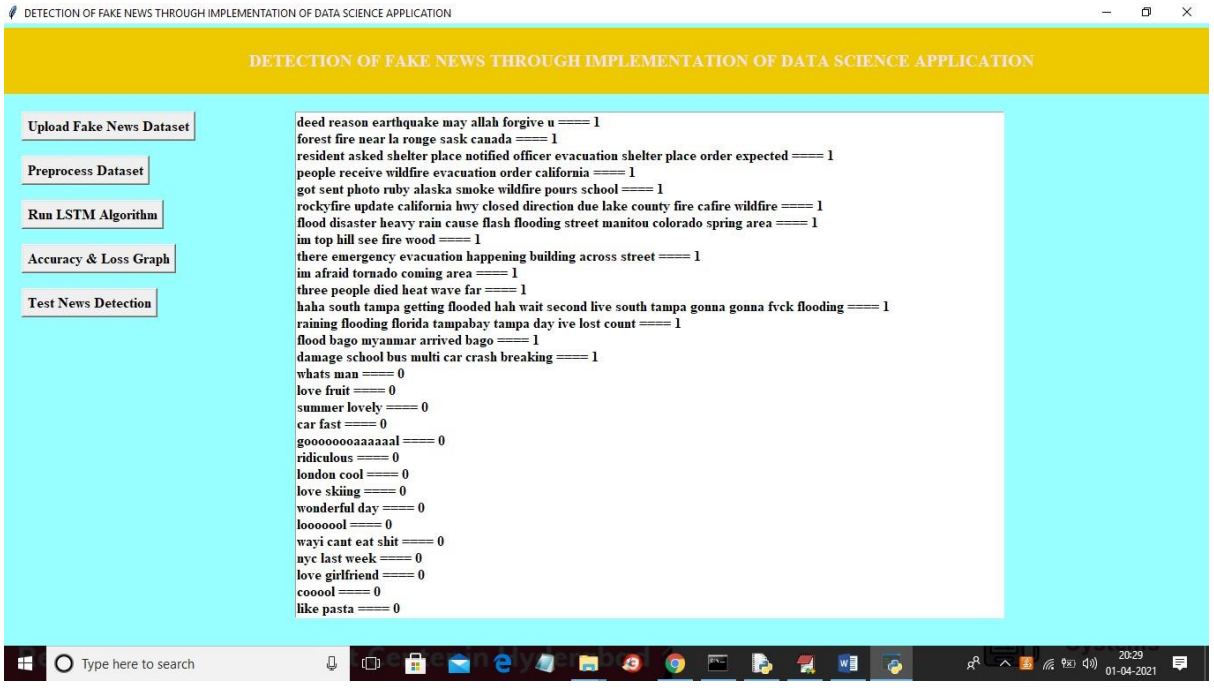
To run project double click on‘run.bat’ file to get below screen



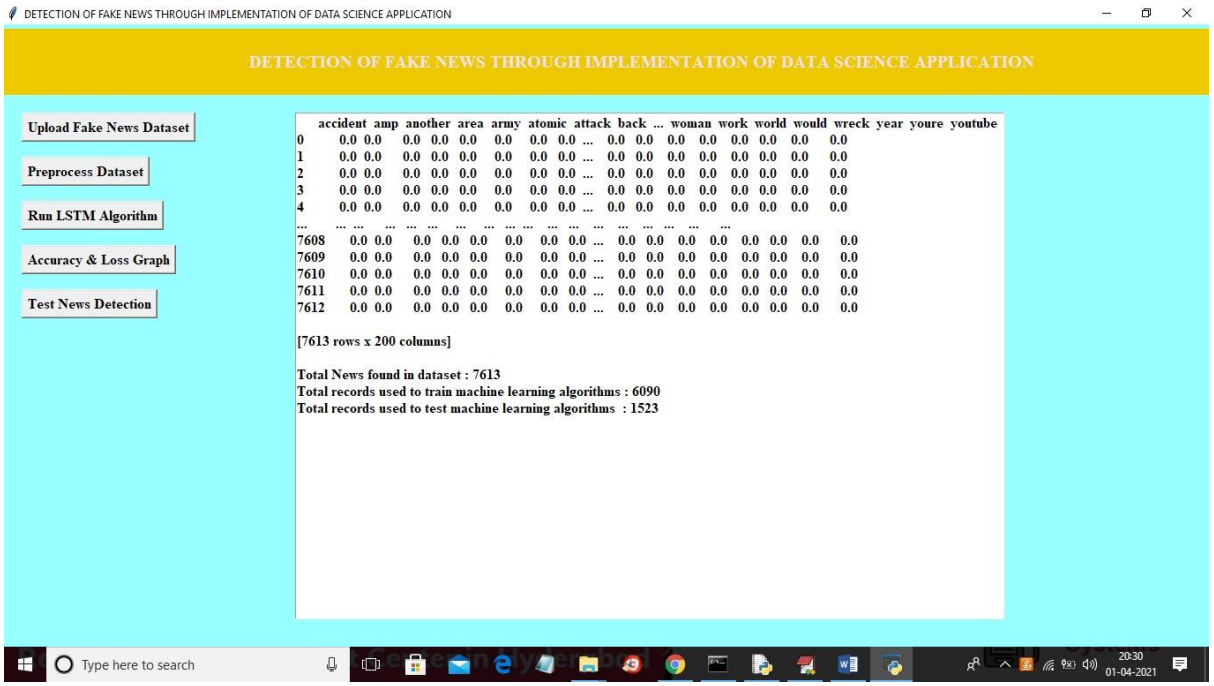

In above screen click on ‘Upload Fake News Dataset’ button to upload dataset



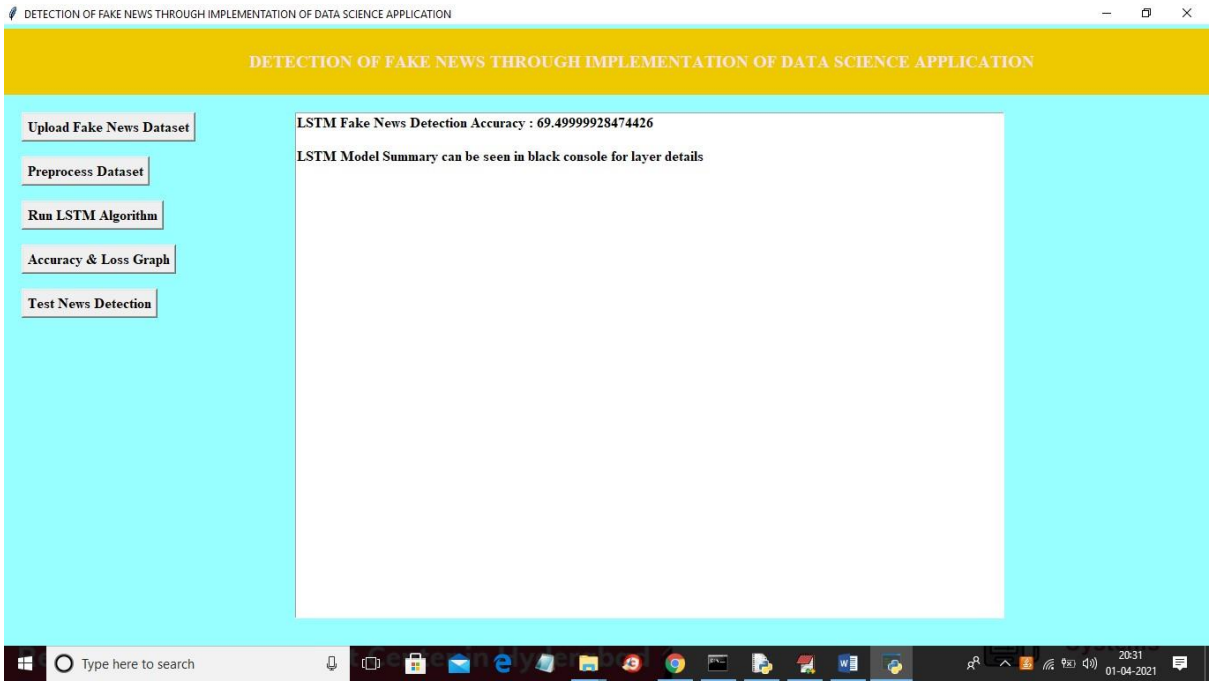

In above screen selecting and uploading 'news.csv' file and then click on 'Open' button to load dataset and to get below screen



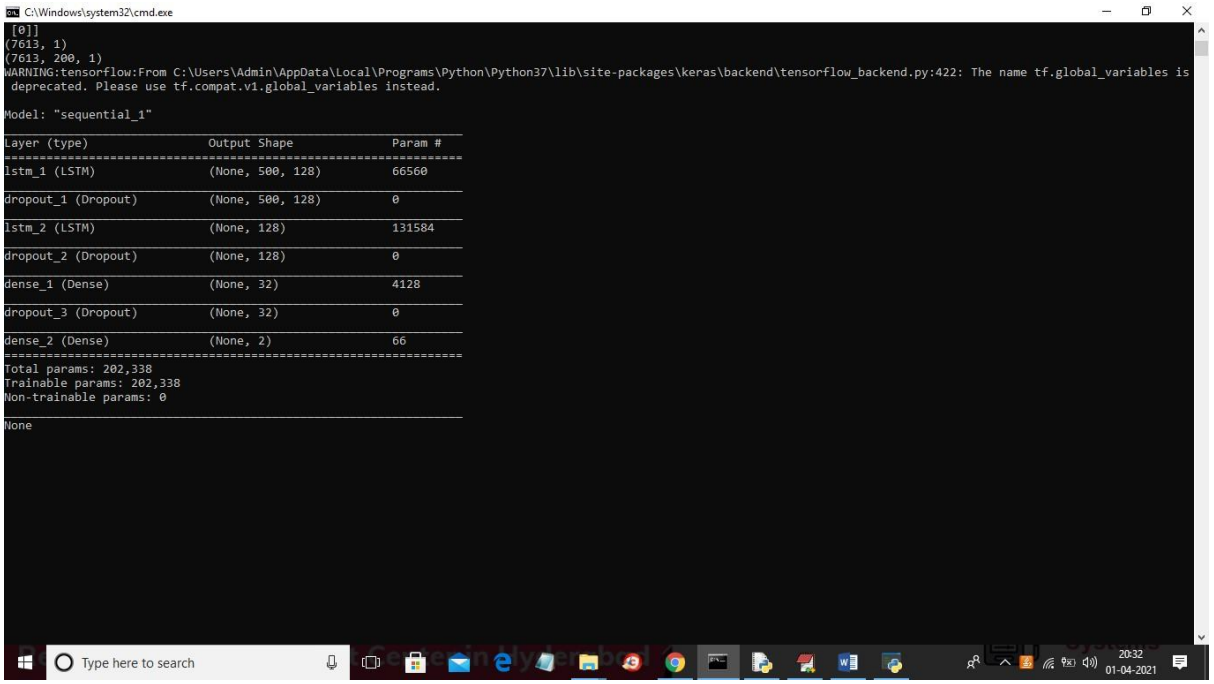

In above screen dataset loaded and then in text area we can see all news text with the class label as 0 or 1 and now click on 'Preprocess Dataset & Apply NGram' button to convert above string data to numeric vector and to get below screen




In above screen all news words put in column header and if that word appear in any row then that rows column will be change with word count and if not appear then 0 will be put in column. In above screen showing some records from total 7612 news records and in bottom lines we can see dataset contains total 7613 records and then application using 80% (6090 news records) for training and then using 20% (1523 news records) for testing and now dataset is ready with numeric record and now click on 'Run LSTM Algorithm' button to train above dataset with LSTM and then build LSTM model and then calculate accuracy and error rate

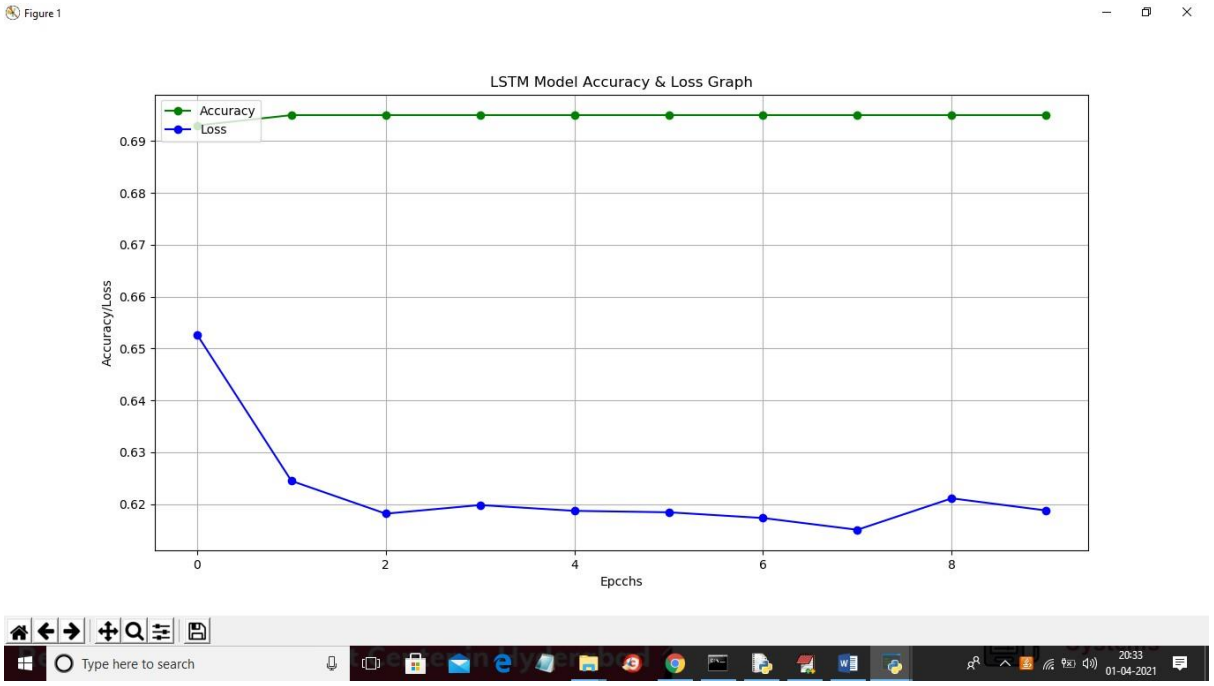


In above screen LSTM model is generated and we got its prediction accuracy as 69.49% and we can see below console to see LSTM layer details.



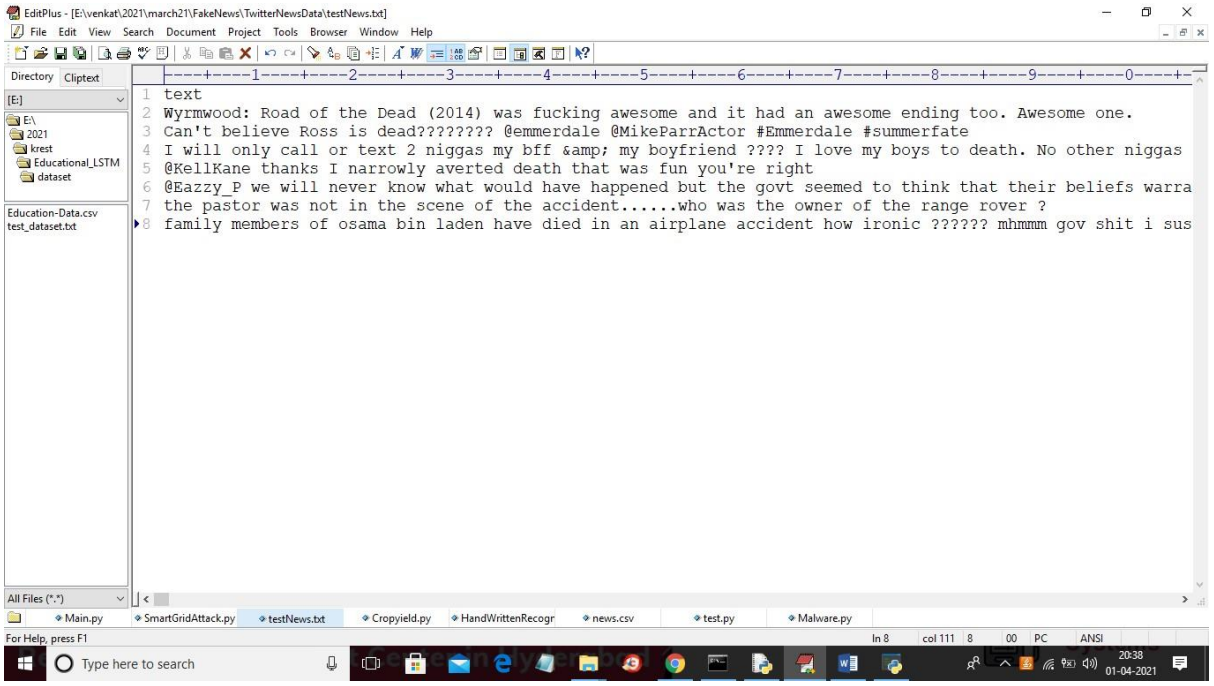

In above screen different LSTM layers are created to filter input data to get efficient features for prediction. Now click on 'Accuracy & Loss Graph' button to get LSTM graph.

**5.2 Analysis:**

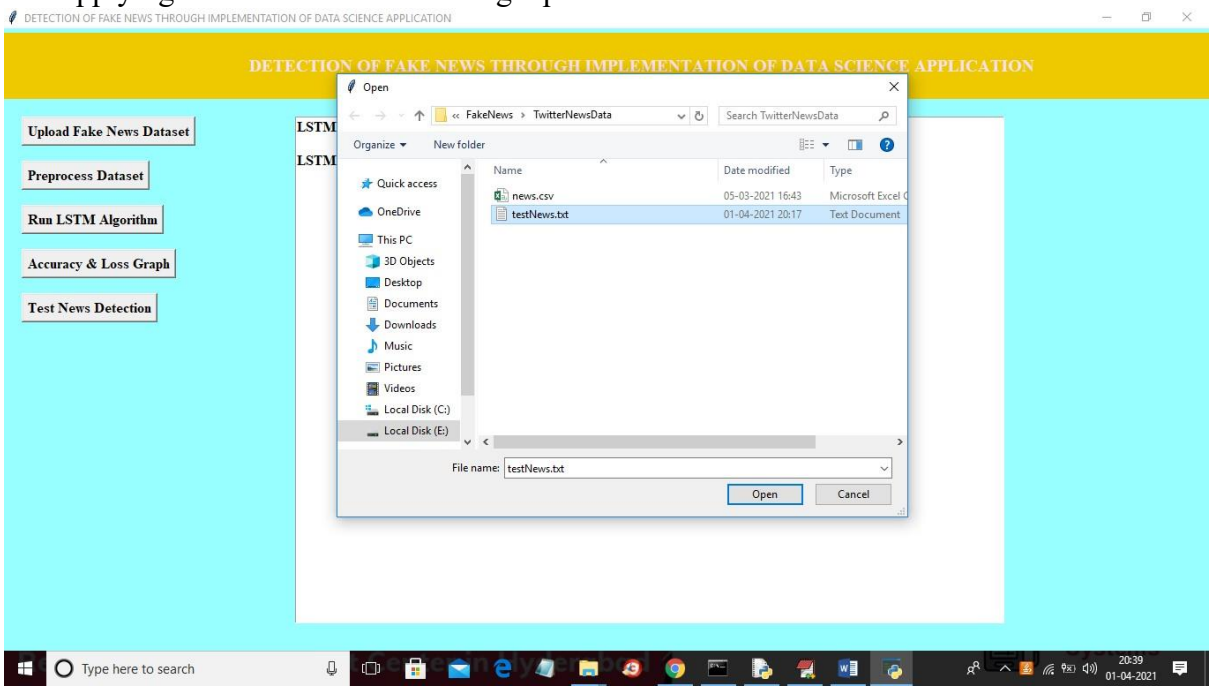



In above graph x-axis represents epoch/iterations and y-axis represents accuracy and loss value and green line represents accuracy and blue line represents loss value and at

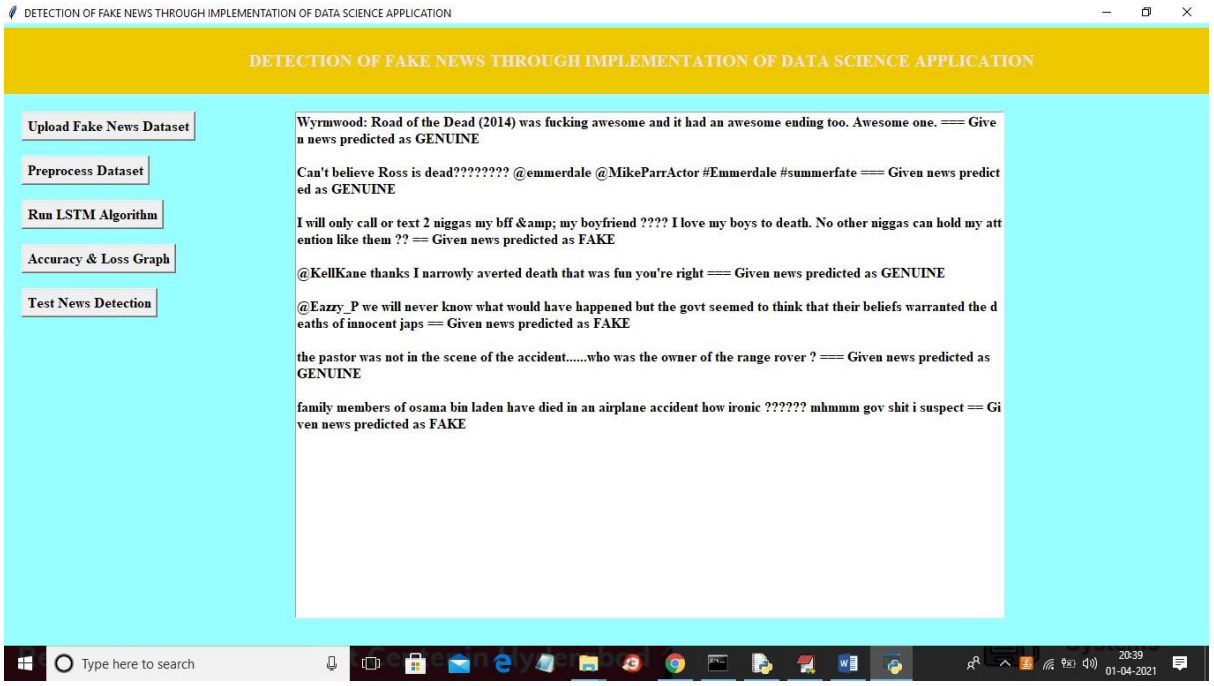
each increasing epoch loss values get decrease and accuracy reached to 70%. Now click on 'Test News Detection' button to upload some test news sentences and then application predict whether that news is genuine or fake. In below test news dataset we can see only TEXT data no class label and LSTM will predict class label for that test news




In above screen in test news we have only one column which contains only news 'TEXT' and after applying above test news we will get prediction result




In above screen selecting and uploading 'testNews.txt' file and then click on 'Open' button to load data and to get below prediction result




**5.3 Summary:**

In above screen before dashed symbols we have news text and after dashed symbol application predict news as 'FAKE or GENUINE'. After building model when we gave any news text then LSTM will check whether more words belongs to genuine or fake category and whatever category get more matching percentage then application will predict that class label.

# 6.TESTING AND VALIDATION

## 6.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTS

### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application

.it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expectedresults.

**Integration testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination ofcomponents **Functional test:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and usermanuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. InvalidInput : identified classes of invalid input must be rejected. Functions : identified functions must beexercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must beinvoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software. It is used to test areas that cannot be reached from a black boxlevel.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, asablackbox.youcannot—seeintoit.Thetestprovidesinputsandrespondsto outputswithout considering how the softwareworks.

**6.2 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

- All field entries must workproperly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not bedelayed.

**Features to be tested**

- Verify that the entries are of the correctformat
- No duplicate entries should beallowed
- All links should take the user to the correctpage.

### 6.3 Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.
The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### 6.4 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

**References** :

[1] M. Balmas, “When Fake News Becomes Real: Combined Exposure to Multiple News Sources and Political Attitudes of Inefficacy, Alienation, and Cynicism,” Communic. Res., vol. 41, no. 3, pp. 430–454, 2014.

[2] C. Silverman and J. Singer-Vine, “Most Americans Who See Fake News Believe It, New Survey Says,” BuzzFeed News, 06-Dec-2016.

[3] P. R. Brewer, D. G. Young, and M. Morreale, “The Impact of Real News about ‘“Fake News”’: Intertextual Processes and Political Satire,” Int. J. Public Opin. Res., vol. 25, no. 3, 2013.

D. Berkowitz and D. A. Schwartz, “Miley, CNN and The Onion,” Journal. Pract., vol. 10, no. 1, pp. 1–17, Jan. 2016

[4] C. Kang, “Fake News Onslaught Targets Pizzeria as Nest of Child-Trafficking,” New York Times, 21-Nov-2016.

C. Kang and A. Goldman, “In Washington Pizzeria Attack, Fake News Brought Real Guns,” New York Times, 05-Dec-2016