# Fake News Classification Using LSTM

**A Project Report submitted in partial fulfillment of the requirements for the award of the degree of,**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted by:

| | |
|---|---|
| **SUSHMA SWARAJ K** | **321810307011** |
| **HARISH KUMAR M L** | **321810307016** |
| **CHANDRA SEKHAR** | **321810307034** |
| **KARTHIK M P** | **321810307050** |

**Under the esteemed guidance of**

**Mrs. KAMALA L**

**ASSISTANT PROFESSOR, GST**

**Department of Computer Science & Engineering,**

**GITAM SCHOOL OF TECHNOLOGY**

**GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(Deemed to be University)**

**Bengaluru Campus.**

**April 2022**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## GITAM SCHOOL OF TECHNOLOGY

## GITAM

## (Deemed to be University)

## CERTIFICATE

This is to certify that the project report titled "**FAKE NEWS CLASSIFICATION USING LSTM**" is a bonafide work carried out by **Sushma Swaraj K (321810307011), Harish Kumar M L (321810307016), Chandra Sekhar (321810307034), Karthik M P (321810307050)** students of B Tech (CSE) of GITAM Deemed to be University, Bengaluru campus during the academic year 2021-22, in partial fulfillment of the requirement for the award of degree of **Bachelors of Technology in Computer Science and Engineering.**

| | |
|---|---|
| **Project Guide** | **Head of the Department** |
| **Mrs. Kamala L** | **Prof. Vamsidhar. Y** |
| Assistant Professor | Head of the Department |
| Department of CSE | Department of CSE |

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## GITAM SCHOOL OF TECHNOLOGY

## GITAM

**(Deemed to be University)**

## DECLARATION

We, hereby declare that the project report entitled **“FAKE NEWS CLASSIFICATION USING LSTM”** is an original work done in the **Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University)** submitted in partial fulfillment of the requirements for the award of the degree of **B.Tech.** in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree.

**Date:**

| Registration No(s). | Name(s) | Signature(s) |
|---|---|---|
| **321810307011** | **SUSHMA SWARAJ K** | |
| **321810307016** | **HARISH KUMAR M L** | |
| **321810307034** | **CHANDRA SEKHAR** | |
| **321810307050** | **KARTHIK M P** | |

# ACKNOWLEDGEMENT

The Satisfaction and Euphoria that accompany the successful completion of any Project would be incomplete without mentioning the people who made it possible, whose constant guidance and encouragements crowned our efforts with success. We take this opportunity to express the deepest gratitude and appreciations to all those who held us directly or indirectly towards the successful completion of the Project.

I would like to thank **Dr. S DINESH**, **Ph.D., Director, GITAM School of Technology** for all the facilities provided.

I would like to thank HOD **Dr. VAMSHIDAR**, **Ph.D., Professor-HOD, Department of Computer Science and Engineering** for his support and encouragement that went a long way in successful completion of this Project.

I consider that as privilege to express our heartfelt gratitude and respect to **MRS.KAMALA LF, Assistant professor, department of Computer Science and Engineering** for being our internal guide for his integral and incessant support offered to us throughout the course of this project and for constant source of inspiration throughout the Seminar.

I would like to thank our parents and our friends for their support and encouragement in the completion of project in time.

Last but not least I would like to thank all the teaching and non-teaching staff members of the Computer Science and Engineering Department, for the support in completion of the project in time.

| **Student's Name** | **Registration No.** |
|---|---|
| **321810307011** | **SUSHMA SWARAJ K** |
| **321810307016** | **HARISH KUMAR M L** |
| **321810307034** | **CHANDRA SEKHAR** |
| **321810307050** | **KARTHIK M P** |

# ABSTRACT

In recent years, deceptive content such as fake news and fake reviews, also known as opinion spam, have increasingly become a dangerous prospect for online users. Fake reviews have affected consumers and stores alike. Furthermore, the problem of fake news has gained attention from 2018 onwards, especially in the aftermath of the last U.S. presidential elections. Intentionally misleading content presented under the guise of legitimate journalism most so-called '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. In this project, we will classify whether the news is fake or not using LSTM machine learning algorithm.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER-1

# INTRODUCTION

Fake News is news, stories, or hoaxes created to deliberately misinform or deceive readers. Usually, these stories are created to either influence people's views, push a political agenda, or cause confusion and can often be a profitable business for online publishers. The purpose of choosing this topic is because it is becoming a serious social challenge. It is leading to a poisonous atmosphere on the web and causing riots and lynching on the road. Examples: political fake news, news regarding sensitive topics such as religion, covid news like salt and garlic can cure corona and all such messages we get through social media. We all can see the damage that can be caused because of fake news which is why there is a dire need for a tool that can validate particular news weather it is fake or real and give people a sense of authenticity based on which they can decide whether or not to take action, amongst so much noise of fake news and fake data if people lose faith in information, they will no longer be able to access even the most vital information that can even sometimes be life- changing or lifesaving. Our approach is to develop a model where in it will detect whether the given news is false or true using LSTM (long short-term memory) and other machine learning concepts such as NLP, word embedding, one hot representation, etc. The model will give us the results for the dataset provided.

## 1.1AIM OF THE PROJECT:

The term “Fake News” was a lot less unheard of and not prevalent a couple of decades ago but in this digital era of social media, it has surfaced as a huge monster. Fake news, information bubbles, news manipulation, and the lack of trust in the media are growing problems within our society. However, in order to start addressing this problem, an in-depth understanding of fake news and its origins is required. Only then one can look into the different techniques and fields of machine learning (ML), natural language processing (NLP), and artificial intelligence (AI) that could help us fight this situation. “Fake news” has been used in a multitude of ways in the last half a year and multiple definitions have been given. Measuring fake news or even defining it properly could very quickly become a subjective matter rather than an objective metric. In its purest form, fake news is completely made up, manipulated to resemble credible journalism and attract maximum attention and, with it, advertising revenue.

The proposed project aimed to develop and test new efficient feature extraction methods and converting text to number and then splitting the dataset into training and testing sets for a more accurate classification of fake news based on LSTM model.

## 1.2 EXISTING SYSTEM

Detecting fake news is believed to be a complex task and much harder than detecting fake product reviews. The open nature of the web and social media, in addition to the recent advance in computer technologies, simplifies the process of creating and spreading fake news. While it's easier to understand and trace the intention and the impact of fake reviews, the intention and the impact of creating propaganda by spreading fake news cannot be measured or understood easily. For instance, it is clear that fake review affects the product owner, customers, and online stores; on the other hand, it is not easy to identify the entities affected by the fake news. This is because identifying these entities requires measuring the news propagation, which has shown to be complex and resource intensive.

**Working of Existing System**

Each is a representation of inaccurate or deceptive reporting. Furthermore, the authors weight the different kinds of fake news and the pros and cons of using different text analytics and predictive modelling methods in detecting them. In their paper, the y separated the fake news types into 3 groups: -

1. Serious fabrications are news not published in mainstream or participant media, yellow press, or tabloids, which, as such, will be harder to collect.

2. Large-Scale hoaxes are creative and unique and often appear on multiple platforms. The authors argued that it may require methods beyond text analytics to detect this type of fake news.

3. Humorous fake news is intended by their writers to be entertaining, mocking, and even absurd. According to the authors, the nature of the style of this type of fake news

could have an adverse effect on the effectiveness of text classification techniques.

It starts with preprocessing the dataset by removing unnecessary characters and words from the data. The n-gram features are extracted, and a matrix of features is formed representing the documents involved. The last step in the classification process is to train the classifier. We investigated different classifiers to predict their class of the documents. We specifically investigated 6 different machine learning algorithms, namely, stochastic gradient descent (SGD), SVM, linear support vector machines (LSVM), K-nearest neighbor (KNN), LR, and decision trees (DT).

Term Frequency is a method that uses word count from texts to find similarities between texts. Each document is represented by a vector of equal length that contains word counts. Next, each vector is made in such a way that the sum of its elements will be added to the other. Each number of words is converted into opportunities for such a word that is present in the documents. For example, if the word is something document, will be represented as 1,
and if any not in the document, it will be set to 0. So, each the document is represented by groups of names. The typical TF of the word w in terms of document d is defined as follows:
Standard Time = Value for Documentary / Total Number of Documentary Opposition (IDF) term w in reference to document corpus D, define as IDF(w) D, by algorithm of the total number of documents in the corpus divided by the number of letters in which the particular name appears, and is calculated as follows:
Inverted document TF = 1+log (total documents /no of documents with particular term)

# CHAPTER-2

# LITERATURE SURVEY

## 1. Shlok Gilda, ‘Evaluating Machine Learning Algorithms for Fake News Detection’ December 2017

The application of natural language processing techniques for the detection of 'fake news', that is, misleading news stories that come from non-reputable sources. Using a dataset obtained from Signal Media and a list of sources from OpenSources.co, Applying term frequency-inverse document frequency (TF-IDF) of bi-grams and probabilistic context free grammar (PCFG) detection to a corpus of about 11,000 articles and test the dataset on multiple classification algorithms-Support Vector Machines, Stochastic Gradient Descent, Gradient Boosting, Bounded Decision Trees, and Random Forests. Resulting that TF-IDF of bi-grams fed into a Stochastic Gradient Descent model identifies non-credible sources with an accuracy of 77.2%, with PCFGs having slight effects on recall.

## 2. Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang and Huan Liu ‘Fake News Detection on Social Media: A Data Mining Perspective’ August 2017

Social media for news consumption is a double-edged sword, it enables the wide spread of “fake news”, i.e., low quality news with intentionally false information. The extensive spread of fake news has the potential for extremely negative impacts on individuals and society. Therefore, fake news detection on social media has recently become emerging research that is attracting tremendous attention. Fake news detection social media presents unique characteristics and challenges that make existing detection algorithms from traditional news media ineffective or not applicable. First, fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content

### 3. Shivam B. Parikh and Pradeep K, Atrey, ‘Media Rich Fake News Detection: A Survey’, April 2018

Fake News has been around for decades and with the advent of social media and modern-day journalism at its peak, detection of media-rich fake news has been a popular topic in the research community. Given the challenges associated with detecting fake news research problem, researchers around the globe are trying to understand the basic characteristics of the problem statement. In this paper aims to present an insight on characterization of news story in the modern diaspora combined with the differential content types of news story and its impact on readers. Subsequently, dive into existing fake news detection approaches that are heavily based on text-based analysis, and also describe popular fake news data-sets.

### 4. Mykhailo Granik and Volodymyr Mesyura, ‘Fake News Detection using Naive Bayes classifier’, 2019

This paper shows a simple approach for fake news detection using naive Bayes classifier. This approach was implemented as a software system and tested against a data set of Facebook news post and achieved classification accuracy of approximately 74% on the test set which is a decent result considering the relative simplicity of the model.

### 5. Gaurav Bhatt, Aman Sharma, Shivam Sharma, Ankush Nagpal, Balasubramanian Raman, and Ankush Mittal, ‘On the Benefit of Combining Neural, Statistical and External Features for Fake News Identification’

In this article they explored a subtask to fake news identification, and that is stance detection. Given a news article, the task is to determine the relevance of the body and its claim. Here they compute the neural embedding from the deep recurrent model, statistical features from the weighted n-gram bag-of-words model and hand-crafted external features with the help of feature engineering heuristics.

### 6. Z Khanam, B N Alwasel and H Sirafi and M Rashid, ‘Fake News Detection Using Machine Learning approaches’

The easy access and exponential growth of the information available on social media networks has made it intricate to distinguish between false and true information. The easy dissemination of information by way of sharing has added to exponential growth of its falsification. The credibility of social media networks is also at stake where the spreading of fake information is prevalent. Thus, it has become a research challenge to automatically check the information viz a viz its source, content and publisher for categorizing it as false or true. Machine learning has played a vital role in classification of the information although with some limitations. This paper reviews various Machine learning approaches in detection of fake and fabricated news. The limitation of such and approaches and improvisation by way of implementing deep learning is also reviewed.

### 7. Tejaswini Yesugade1, Shrikant Kokate, Sarjana Patil, Ritik Varma, Sejal Pawar, ‘Fake News Detection using LSTM’ 2021

We are in the age of information, every time we read a piece of information or watch the news on TV, we look for a reliable source. There is so many fake news spread all over the internet and social media. Fake news is misinformation or manipulated news that is spread across the social media with an intention to damage a person, agency and organization. The spread of misinformation in critical situations can cause disasters. Due to the dissemination of fake news, there is need for computational methods to detect them. So, to prevent the harm that can be done using technology, we have implemented Machine Learning algorithms and techniques such as NLTK, LSTM. Our contribution is bifold. First, we must introduce the dataset which contain both fake and real news and conduct various experiments to organize fake news detector. We got better results compared to the existing systems.

# CHAPTER-3

# SOFTWARE AND HARDWARE SPECIFICATIONS

## 3.1 INTRODUCTION:

Here the aim is to create a machine learning model that could detect fake news. Detecting fake news is one of the most important to avoid spread of false or misleading information in today's world where news plays important role in making decision in everyone life. For this reason, a project where we could detect a fake news using ML models.

In this project, the libraries spacy, nltk, numpy, sklearn, tensorflow, pandas and WordCloud, Seaborn and keras are used.

Python-Python is easy to learn and work on with the language. It is an elevated level, broadly useful programming and profoundly intruded on language.

## 3.2 SOFTWARE REQUIREMENT:

- **Python idle 3.7 version (or)**
- **Anaconda 3.7 (or)**
- **Jupiter (or)**
- **Google Colab**

**Package Required**:

keras==2.8.0, ipython==7.4.0p, keras==2.3.1, matplotlib==3.2.2, numpy==1.21.5, pandas==1.3.5, plotly==5.5.0, scipy==1.4.1, seaborn==0.11.2, sklearn==0.1

Jupyter Notebook has the following features:
It is very flexible tool to create readable analyses, because one can keep code, images, comments, formula and plots together: Jupyter is quite extensible, supports many programming languages, easily hosted on almost any server you only need to have SSH or http access to a server. And it is completely free.

## Libraries

- **Tensor flow**

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications. TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well. TensorFlow provides stable Python and C++ APIs, as well as non-guaranteed backward compatible API for other languages.

- **Numpy**

NumPy is often used along with packages like Scipy (Scientific Python) and Matplotlib (plotting library). This combination is widely used as a replacement for MATLAB, a popular platform for technical computing. However, Python alternative to MATLAB is now seen as a more modern and complete programming language. It is open-source, which is an added advantage of NumPy. The most important object defined in NumPy is an N-dimensional array type called 2$^{nd}$ array. It describes the collection of items of the same type. Items in the collection can be accessed using a zero-based index.

- **Keras**

Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers, Francois Chollet. It is made user-friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination. It cannot handle low-level computations, so it makes use of the Backend library to resolve it. The backend library act as a high-level API wrapper for the low-level API, which lets it run on TensorFlow, CNTK, or Theano.

- **Gensim**

  Genism is an open-source library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. Genism is implemented in Python and Python for performance. Genism is designed to handle large text collections using data streaming and incremental online algorithms, which differentiates it from most other machine learning software packages that target only in-memory processing.

- **Sklean**

  Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

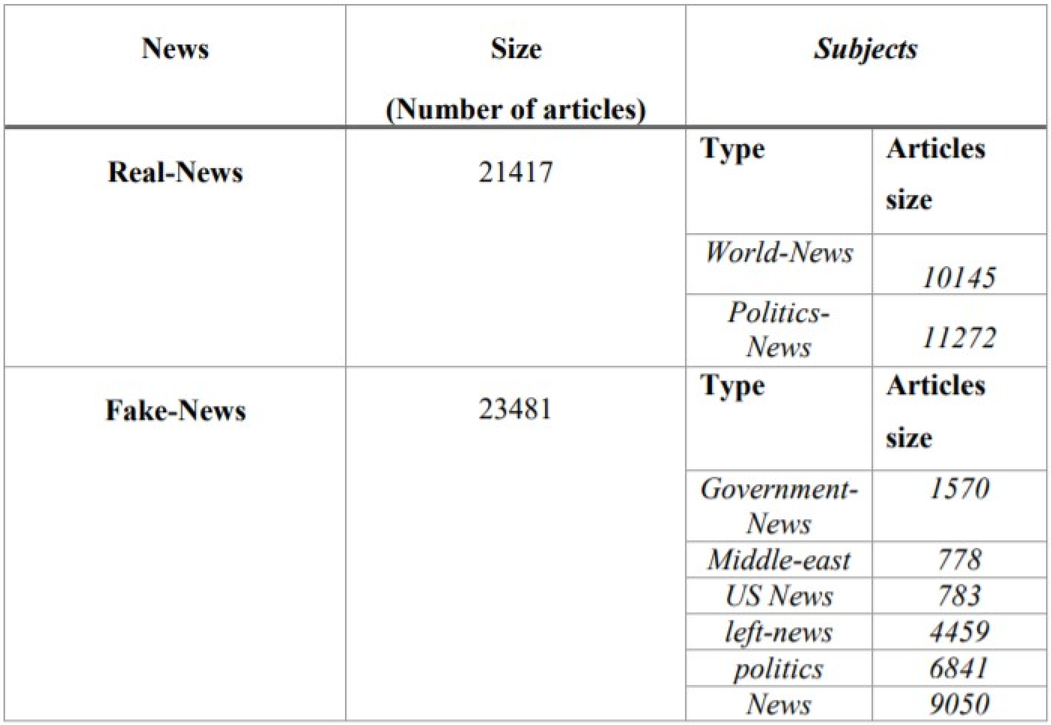
## 3.3 HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

| | | |
|---|---|---|
| • | **Operating system** | **: Windows, Linux** |
| • | **Processor** | **: Minimum intel i3** |
| • | **Ram** | **: Minimum 4 GB** |
| • | **Hard disk** | **: Minimum 250 GB** |

## 3.4 ABOUT DATASET

The dataset contains two types of articles fake and real News. This dataset was collected from real world sources; the truthful articles were obtained by crawling articles from Reuters.com (News website). As for the fake news articles, they were collected from different sources. The fake news articles were collected from unreliable websites that were flagged by PolitiFact (a fact-checking organization in the USA) and Wikipedia. The dataset contains different types of articles on different topics; however, the majority of articles focus on political and World news topics. The dataset consists of two CSV files. The first file named “True.csv” contains more than 21417 articles from reuter.com. The second file named “Fake.csv” contains more than 23481 articles from different fake news outlet resources. Each article contains the following information: article title, text, type and the date the article was published on. To match the fake news data collected for kaggle.com, we focused mostly on collecting articles from 2016 to 2017. The data collected were cleaned and processed, however, the punctuations and mistakes that existed in the fake news were kept in the text. The following table gives a breakdown of the categories and number of articles per category.



| **News** | **Size (Number of articles)** | ***Subjects*** | |
|---|---|---|---|
| **Real-News** | 21417 | **Type** | **Articles size** |
| | | *World-News* | *10145* |
| | | *Politics-News* | *11272* |
| **Fake-News** | 23481 | **Type** | **Articles size** |
| | | *Government-News* | *1570* |
| | | *Middle-east* | *778* |
| | | *US News* | *783* |
| | | *left-news* | *4459* |
| | | *politics* | *6841* |
| | | *News* | *9050* |

Table 1: Number of articles in dataset

# CHAPTER-4

# PROBLEM STATEMENT

Fake news is false or misleading information presented as news. Fake news often aims to damage the reputation of a person or entity or make money through advertising revenue. In a world becoming more and more connected, it is easier for lies to spread. It turns out that it is possible to detect fake news with a dataset consisting of news articles classified as either reliable or not. Artificial Intelligence or Machine learning-based counterfeit news detector is crucial for companies and media to predict whether circulating information is fake or not automatically. In this project, (LSTM) model will be trained to anticipate if the news is classified as authentic or fake.

## 4.1 OBJECTIVES

Fake news classification using LSTM

- To build a model to recognize fake news using the spacy, and NLTK for Natural Language Processing and Tensorflow, pandas with WordCloud, Seaborn, and Plotty for visualizations libraries and the dataset.
- We use the libraries spacy for advanced NLP and sklearn to build a model. That is LSTM, which will recognize text from the dataset by first loading the data, extracting features from it, and converting text to number and then splitting the dataset into training and testing sets. Then, initialize an LSTM and train the model. Finally, the accuracy of the model will be calculated.

## 4.2 APPLICATIONS

- **Journalism**

  The major spread of information and trusted source is through newspapers and news channels, so this detection can be used to verify the news before broadcasting it.

- **Social Media**

  In today's world of social media, it is easy to manipulate any information or news. Such manipulated news misguides the readers. It is import ant to identify that news is fake or real. This paper provides various techniques that can be used in detection and classification of information.

- Fake News Detection system will help control the spread of fake news over social media.
- This way, we can help the people to make more informed decisions, and they are not made to think about what others are trying to manipulate to believe.
- A Fake News Detection system will reduce the burden of checking the news's authenticity manually and save lots of time.

## 4.3 Limitations

While the results discussed herein suggest for model some external features like source of the news, author of the news, place of origin of the news, time stamp of news were not considered in our model which can be influence the outcome of the model. Availability of dataset and literature papers are limited for fake news detection. The length of the news that is heading or whole news is less which affects the result in terms of accuracy. In Fake News with increasing in layer of module training time increases.

# CHAPTER-5

# DESIGNING DESIGN

## 5.1 SYSTEM ARCHITECTURE

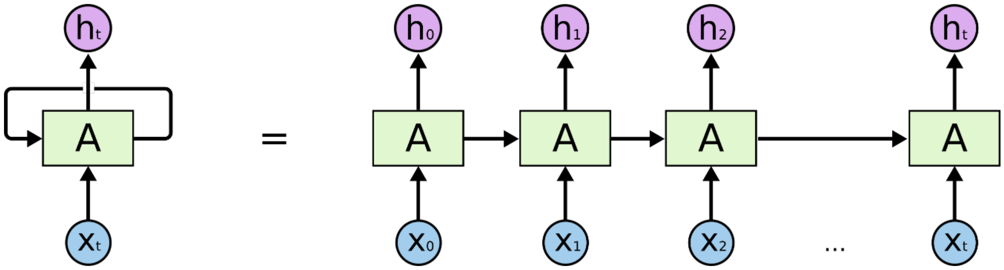


**Fig 1: LSTM Architecture**

## SYSTEM DESCRIPTION

Overview of Dataset: Dataset is taken from Kaggle platform. It has the following attributes: id: unique id for a news article, title: the title of a news article, author: author of the news article, text: the information of news article. Dataset consist of total 44898 news articles for training and testing of model. Dataset is formed with combination of real and fake news.

## Implementation details:

## PREPROCESSING:

To transform data into the relevant format the data set needs preprocess. Firstly, we removed all the NAN values from the dataset. Vocabulary size of 5000 words is decided. Then NLTK (Natural Language Processing) Tool Kit is used to remove all the stop words from the dataset. Stop words is list of punctuations + stop words from nltk toolkit i.e.. Words such as ‘and’ ‘the’ and ‘I’ that don’t convey much information converting them to lowercase and removing punctuation. For each word in documents if it is not a stop word then that words tag is taken from postag. Then, this collection of words is appended to document.

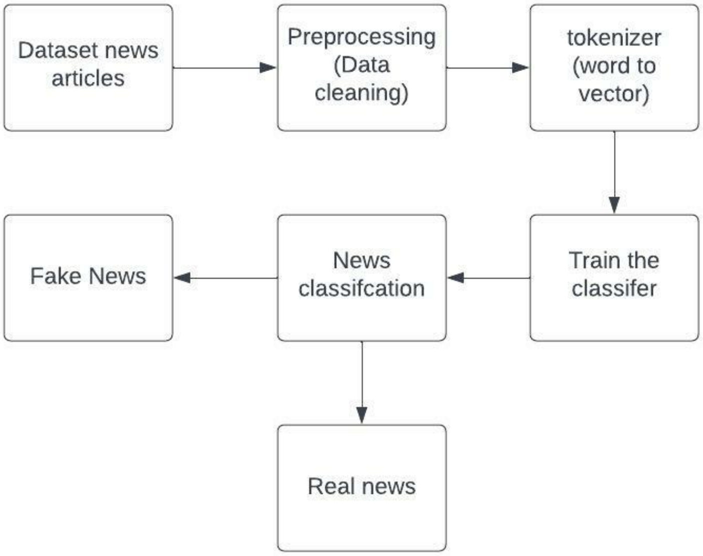


**Fig 2: System block diagram**

## NLP

Humans communicate with each other using words and text. The way that humans convey information to each other is called Natural Language. Every day humans share a large quality of information with each other in various languages as speech or text.

However, computers cannot interpret this data, which is in natural language, as they communicate in 1s and 0s. The data produced is precious and can offer valuable insights. Hence, you need computers to be able to understand, emulate and respond intelligently to human speech.

Natural Language Processing or NLP refers to the branch of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages.

NLP combines the field of linguistics and computer science to decipher language structure and guidelines and to make models which can comprehend, break down and separate significant details from text and speech.

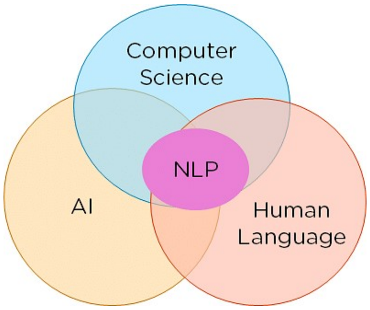


**Fig 3: Constituents of NLP**

## Neural Network

A Neural Network consists of different layers connected to each other, working on the structure and function of a human brain. It learns from huge volumes of data and uses complex algorithms to train a neural net.

Here is an example of how neural networks can identify a dog's breed based on their features.

- The image pixels of two different breeds of dogs are fed to the input layer of the neural network.
- The image pixels are then processed in the hidden layers for feature extraction.
- The output layer produces the result to identify if it's a German Shepherd or a Labrador.
- Such networks do not require memorizing the past output.

Several neural networks can help solve different business problems. Let's look at a few of them.

- Feed-Forward Neural Network: Used for general Regression and Classification problems.
- Convolutional Neural Network: Used for object detection and image classification.
- Deep Belief Network: Used in healthcare sectors for cancer detection.

- RNN: Used for speech recognition, voice recognition, time series prediction, and natural language processing.

## Recurrent Neural Network (RNN)

RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer.

Below is how you can convert a Feed-Forward Neural Network into a Recurrent Neural Network:

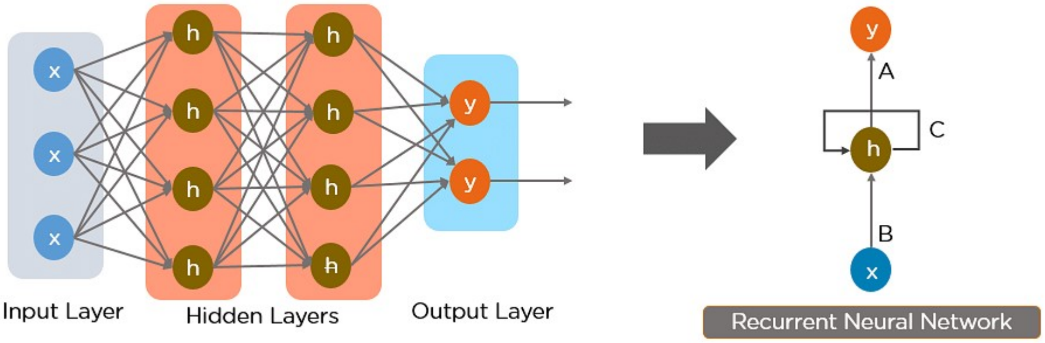


**Fig 4: Simple Recurrent Neural Network**

The nodes in different layers of the neural network are compressed to form a single layer of recurrent neural networks. A, B, and C are the parameters of the network.

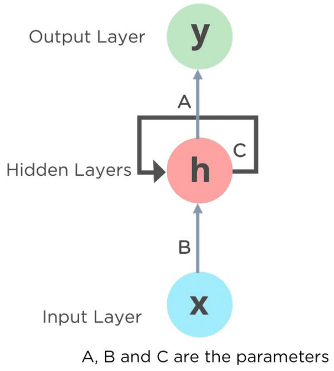


**Fig 5: Fully connected Recurrent Neural Network**

Here, “x” is the input layer, “h” is the hidden layer, and “y” is the output layer. A, B, and C are the network parameters used to improve the output of the model. At any given time, t, the current input is a combination of input at x(t) and x(t-1). The output at any given time is fetched back to the network to improve on the output.

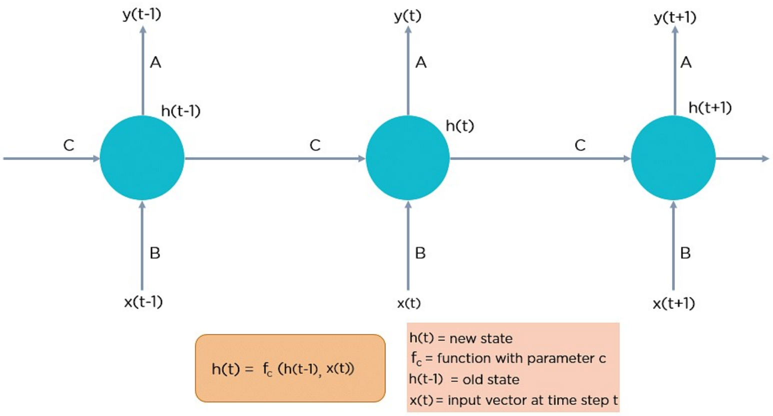


**Fig 6: Fully connected Recurrent Neural Network**

## Reasons to choose Recurrent Neural Networks

RNN were created because there were a few issues in the feed-forward neural network:

- Cannot handle sequential data
- Considers only the current input
- Cannot memorize previous inputs

The solution to these issues is the RNN. An RNN can handle sequential data, accepting the current input data, and previously receive inputs. RNNs can memorize previous inputs due to their internal memory.

## LSTM MODEL:

Long short-term memory (LSTM) units are a building block f or the layers of a recurrent neural network (RNN). A LSTM unit is composed of a cell, an input gate an output gate and a forget gate The cell is responsible for "remembering" values over a vast time interval so that the relation of the word in the starting of the text can influence the output of the word later in the sentence. Traditional neural networks cannot remember or keep the record of what all is passed before they are executed this stops the desired influence of words that comes in the sentence before to have any influence on the ending words, and it seems like a major shortcoming.

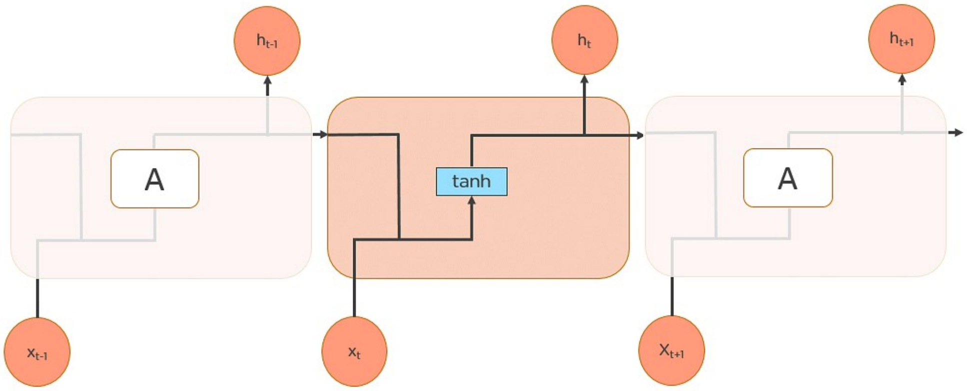


**Fig 7: Long Short-Term Memory Networks**

## Dense Layer

In any neural network, a dense layer is a layer that is deeply connected with its preceding layer which means the neurons of the layer are connected to every neuron of its preceding layer. This layer is the most commonly used layer in artificial neural network networks.

The dense layer's neuron in a model receives output from every neuron of its preceding layer, where neurons of the dense layer perform matrix-vector multiplication. Matrix vector multiplication is a procedure where the row vector of the output from the preceding layers is equal to the column vector of the dense layer. The general rule of matrix-vector multiplication is that the row vector must have as many columns like the column vector.

## 5.2 METHODOLOGY

### 5.2.1. PROPOSED SYSTEM

The analyses were carried out on FNC dataset. After pre-processing the dataset, then the text in the dataset is converted to vectors using tokenizer and genism models. Which then are fed to the machine learning model such as Long Short-Term Memory (LSTM).

## 5.2.2 DATA VISUALIZATION

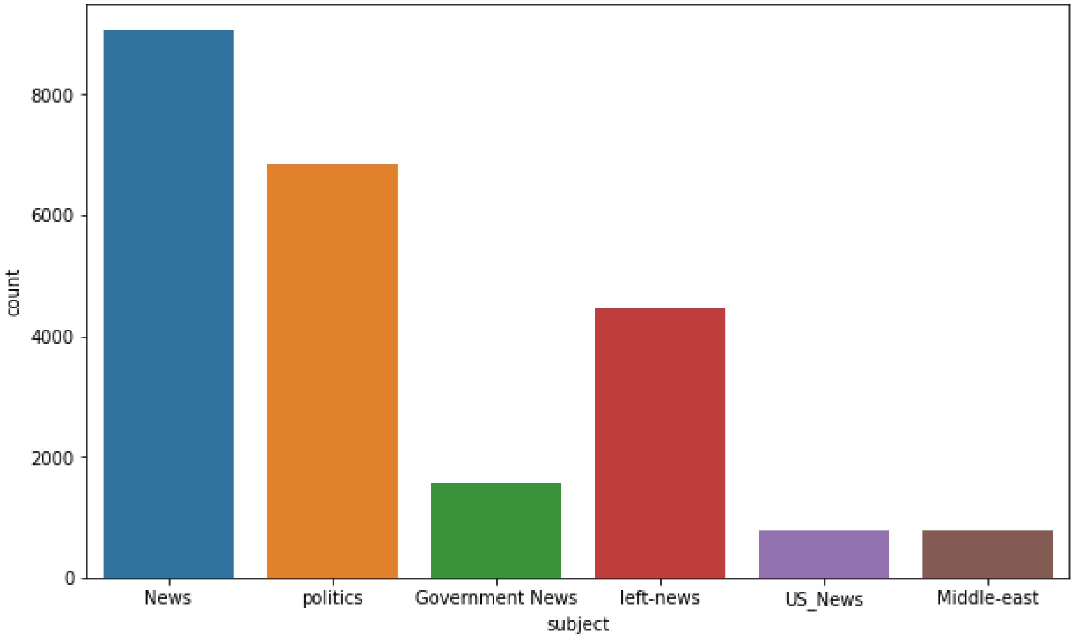


**Fig 8: Number of sample of subjects in dataset**

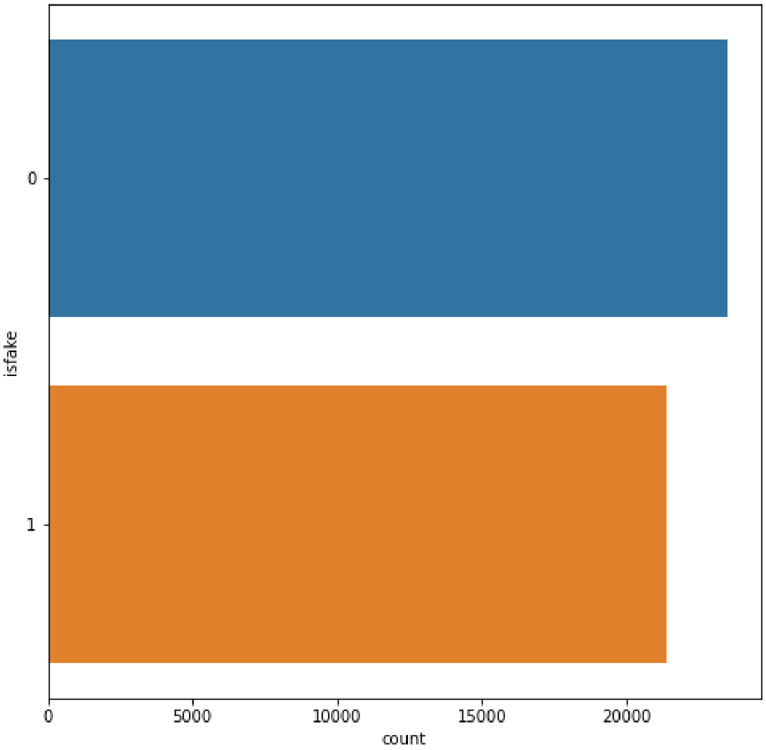


**Fig 9: Fake news vs Real news**

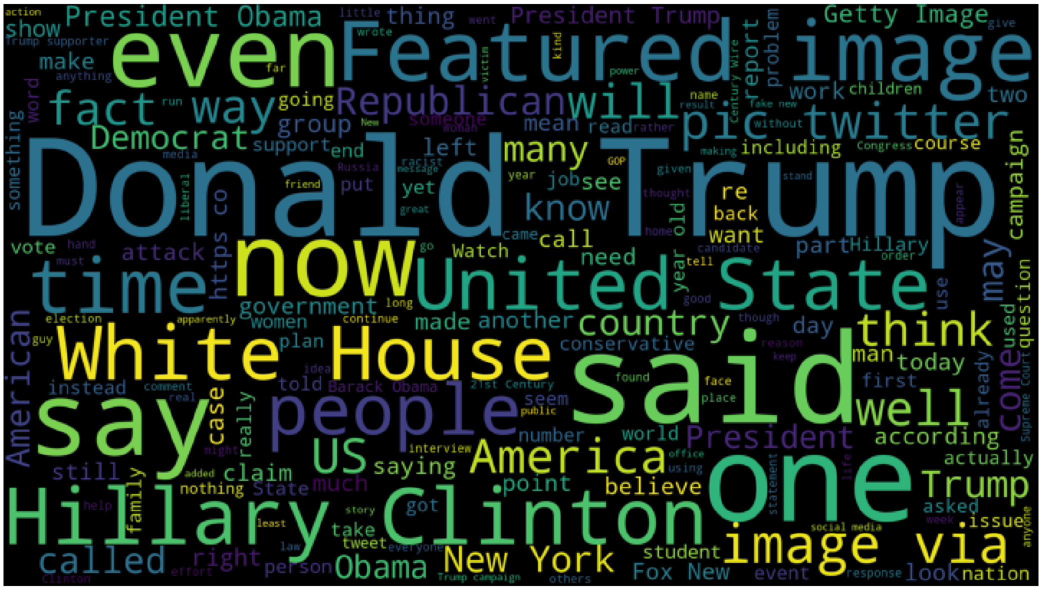


Fig 10: WordCloud text of fake

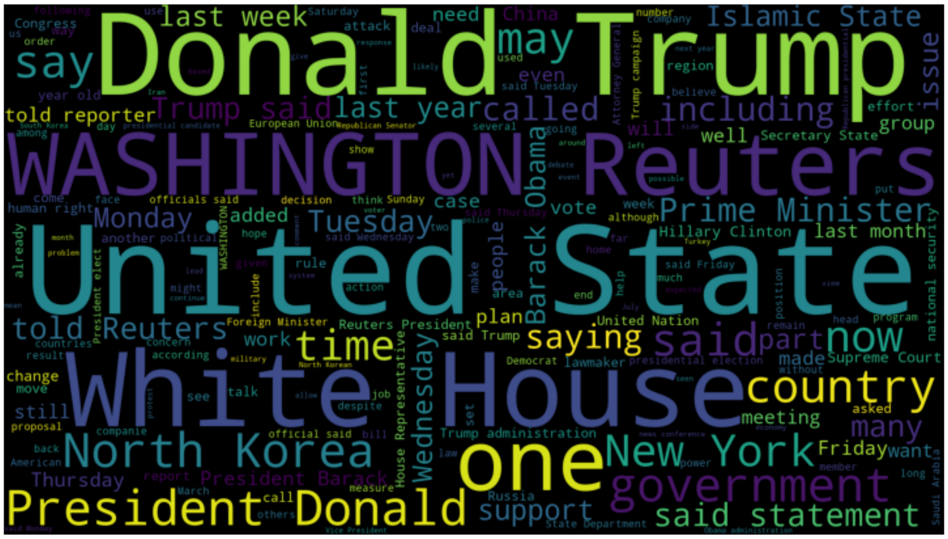


Fig 11: WordCloud text of real

## 5.2.3 DATA PRE-PROCESSING

Before representing the data using LSTM model and vector-based model, the data need to be subjected to certain refinements like stop-word removal, tokenization, a lower casing, sentence segmentation, and punctuation removal. This will help us reduce the size of actual data by removing the irrelevant information that exists in the data. We created a generic processing function to remove punctuation and non-letter characters for each document; then we lowered the letter case in the document. In addition, an LSTM word-based tokenizer was created to slice the text based on the length of n.

### WORD INDEX OF TOKENIZE DATASET

Word tokenizing, appends text to a list and the list be named as documents. The output for this stage is the list of all the words in the narration.

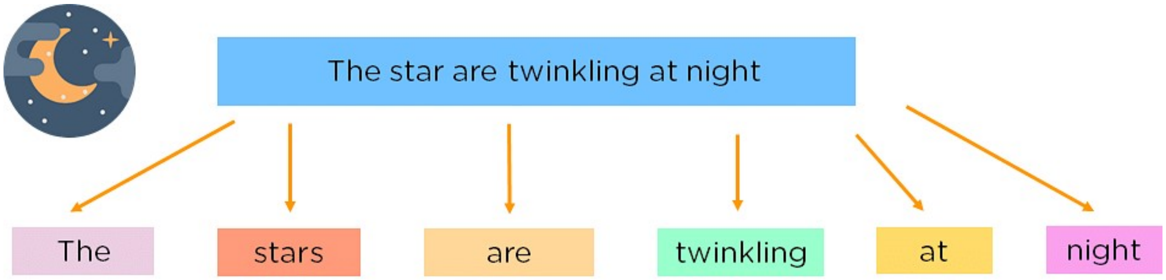


**Figure 13: Tokenization**

### Stop Word Removal

Stop words are insignificant words in a language that will create noise when used as features in text classification. These are words commonly used a lot in sentences to help connect thought or to assist in the sentence structure. Articles, prepositions and conjunctions and some pronouns are considered stop words. We removed common words such as, a, about, an, are, as, at, be, by, for, from, how, in, is, of, on, or, that, the, these, this, too, was, what, when, where, who, will, etc. Those words were removed from each document, and the processed documents were stored and passed on to the next step.

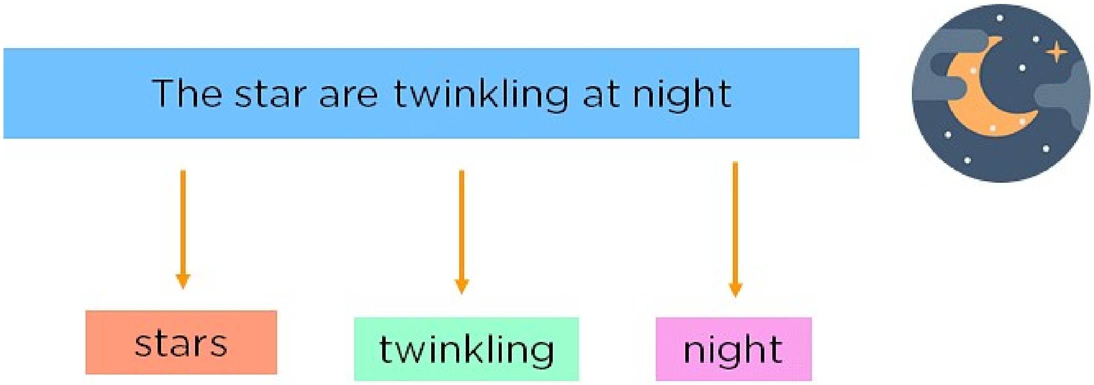


**Figure 12: Stop Words**

## WORD EMBEDDING

Gensim Representation: We cannot give input in the form of text format to the algorithm so we have to convert them into the numeric form, for which we are using one hot representation. In Gensim representation each word in the dataset will be provided its index from the define vocabulary size and these indexes are replaced in sentence. While giving input to the word embedding, we have to provide it with the fix length. To convert each sentence into the fix length padding sequences is used. We have considered max length of 20 words while padding title. Either we can provide padding before the sentence (pre) or after the sentence (post), and then these sentences pass as input to the word embedding. Word embedding apply feature extraction on the provided input vector. In total 40ivector features are considered.

## MODEL

Output from the word embedding is provided to the model. The machine learning model implemented here is a sequential model consisting of embedding as first layer which consist of values vocabulary size, number of features and length of sentence. The next is LSTM with 100 neurons for each layer, followed by Dense layer with sigmoid activation function as we need one final output. We have used binary cross entropy to calculate loss, Adam optimizer for adaptive estimation, finally adding drop out layer in between so that overfitting is avoided. Then training and testing of model id done.

## CLASSIFICATION

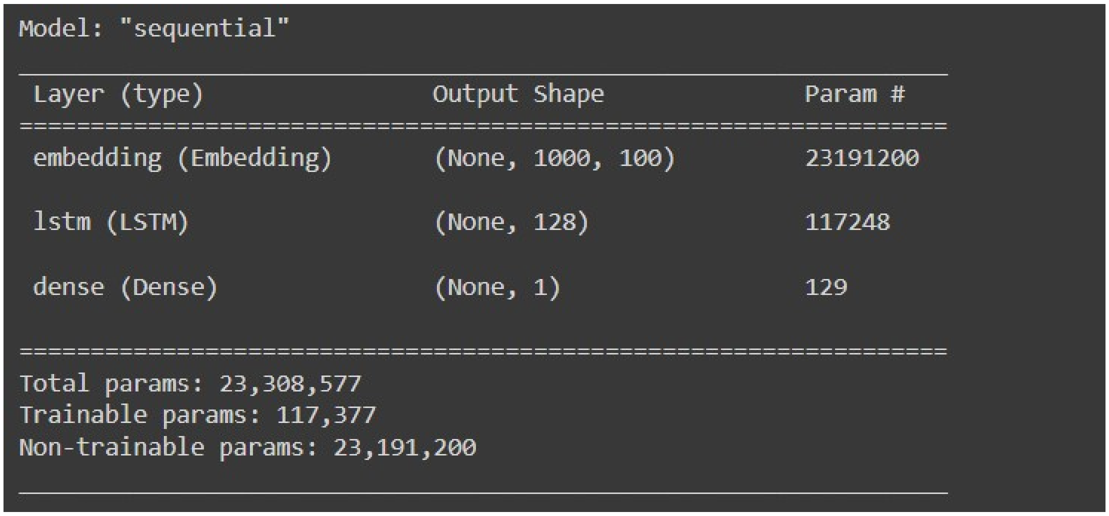
For both preprocessed testing data, the result is predicted. If the predicted value>0.5 Classified as 1 is real and 0 is fake. Accuracy = (TP + TN) / Total. The following terms were used:

**True Negative (TN)**: The prediction was negative and test cases, too, were actually negative.

**True Positive (TP)**: The prediction was positive and test cases, too, were actually positive.

**False Negative (FN)**: The prediction was negative, but the test cases were actually positive.

**False Positive (FP)**: The prediction was positive, but the test cases were actually negative.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1000, 100)         23191200

 lstm (LSTM)                 (None, 128)               117248

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 23,308,577
Trainable params: 117,377
Non-trainable params: 23,191,200
_________________________________________________________________
```

**Fig 14: Model**

**Model Training and Model Evaluation**

We train the model with the dataset which we downloaded from kaggle which consists of fake news and real news and following is the output of the trained model which gives us an accuracy rate of 99%.

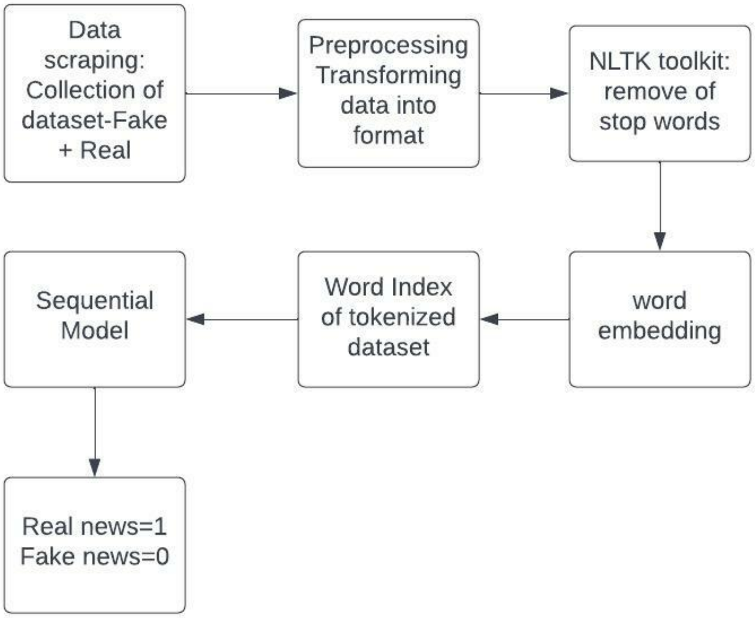

Data scraping: Collection of dataset-Fake + Real
Preprocessing Transforming data into format
NLTK toolkit: remove of stop words
word embedding
Word Index of tokenized dataset
Sequential Model
Real news=1
Fake news=0

**Fig 15: Architecture flow of proposed system**

# CHAPTER-6

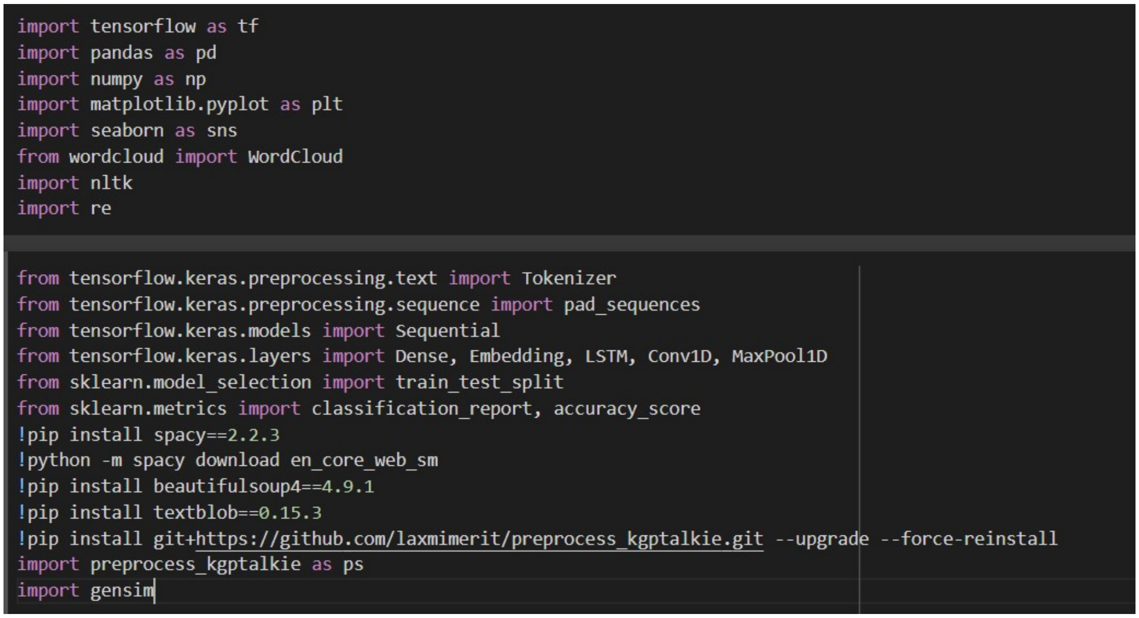
# IMPLEMENTATION

## 6.1 DATA COLLECTION

The first step in implementing the Fake news classification using LSTM system is to collect fake and real news data under different categories which can be used to train the model. The data samples are usually stored in csv format.

## 6.2 PYTHON LIBRARY

The next step is to import all the necessary libraries and modules.



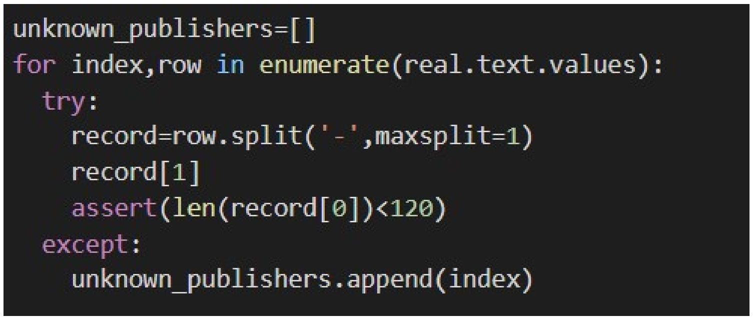
```
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
import nltk
import re
```

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM, Conv1D, MaxPool1D
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
!pip install spacy==2.2.3
!python -m spacy download en_core_web_sm
!pip install beautifulsoup4==4.9.1
!pip install textblob==0.15.3
!pip install git+https://github.com/laxmimerit/preprocess_kgptalkie.git --upgrade --force-reinstall
import preprocess_kgptalkie as ps
import gensim
```

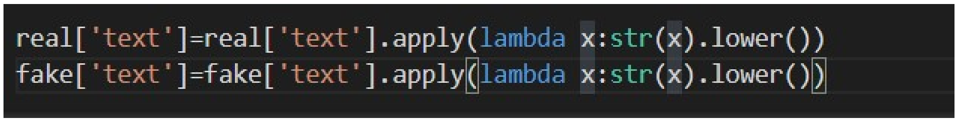
**Fig 16: Import required libraries**

## 6.3DATA PRE-PROCESSING

The next step after data collection is to clean the data and to represent this text numerically, in order to perform further analysis on them. This step is called data pre-processing and tokenizers.



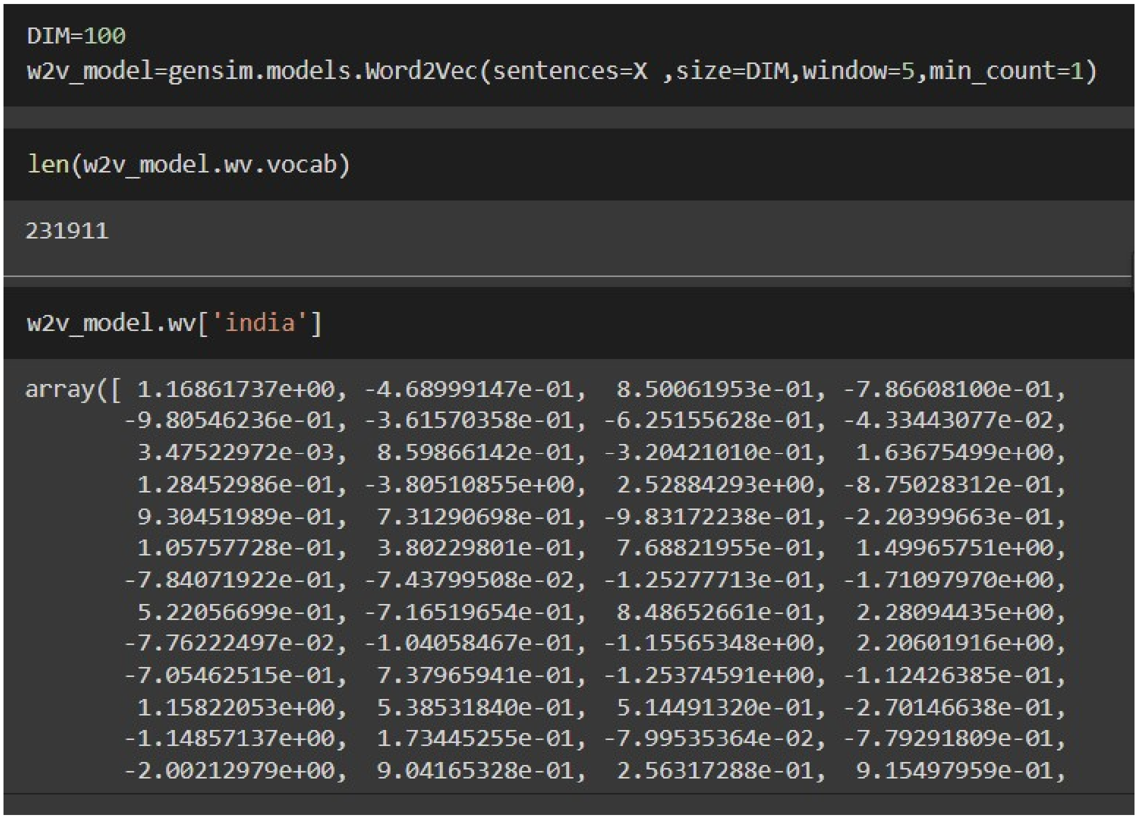
```
unknown_publishers=[]
for index,row in enumerate(real.text.values):
  try:
    record=row.split('-',maxsplit=1)
    record[1]
    assert(len(record[0])<120)
  except:
    unknown_publishers.append(index)
```



```
real['text']=real['text'].apply(lambda x:str(x).lower())
fake['text']=fake['text'].apply(lambda x:str(x).lower())
```

**Fig 17: Data Pre-processing**

After pre-processing the text, the text will be converted into vectors using tokenizer and genism modules.



```
DIM=100
w2v_model=gensim.models.Word2Vec(sentences=X ,size=DIM,window=5,min_count=1)
```

```
len(w2v_model.wv.vocab)
```

```
231911
```

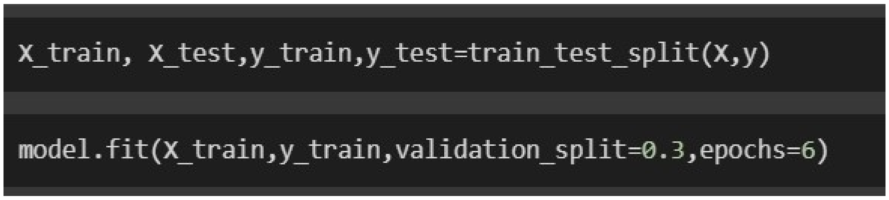
```
w2v_model.wv['india']
```

```
array([ 1.16861737e+00, -4.68999147e-01,  8.50061953e-01, -7.86608100e-01,
       -9.80546236e-01, -3.61570358e-01, -6.25155628e-01, -4.33443077e-02,
        3.47522972e-03,  8.59866142e-01, -3.20421010e-01,  1.63675499e+00,
        1.28452986e-01, -3.80510855e+00,  2.52884293e+00, -8.75028312e-01,
        9.30451989e-01,  7.31290698e-01, -9.83172238e-01, -2.20399663e-01,
        1.05757728e-01,  3.80229801e-01,  7.68821955e-01,  1.49965751e+00,
       -7.84071922e-01, -7.43799508e-02, -1.25277713e-01, -1.71097970e+00,
        5.22056699e-01, -7.16519654e-01,  8.48652661e-01,  2.28094435e+00,
       -7.76222497e-02, -1.04058467e-01, -1.15565348e+00,  2.20601916e+00,
       -7.05462515e-01,  7.37965941e-01, -1.25374591e+00, -1.12426385e-01,
        1.15822053e+00,  5.38531840e-01,  5.14491320e-01, -2.70146638e-01,
       -1.14857137e+00,  1.73445255e-01, -7.99535364e-02, -7.79291809e-01,
       -2.00212979e+00,  9.04165328e-01,  2.56317288e-01,  9.15497959e-01,
```

**Fig 18: Text to Vector conversions**

Here we can observe India keyword is represent as a vector format similar it’s done to every word in the dataset and then we will be training and testing the model.

## 6.4 LOADING DATA AND DATA SPLITTING

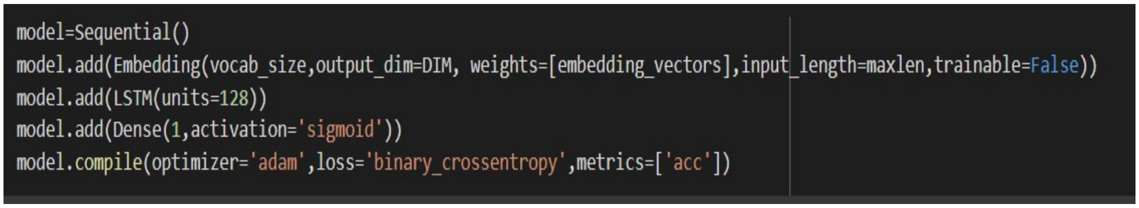


```
X_train, X_test,y_train,y_test=train_test_split(X,y)

model.fit(X_train,y_train,validation_split=0.3,epochs=6)
```

**Fig 19: Loading Data and Data Splitting**

where we are using 80% of data for training and 20 % for the testing by splitting

## 6.5 INTIALIZING THE SEQUENTIAL MODEL



```
model=Sequential()
model.add(Embedding(vocab_size,output_dim=DIM, weights=[embedding_vectors],input_length=maxlen,trainable=False))
model.add(LSTM(units=128))
model.add(Dense(1,activation='sigmoid'))
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['acc'])
```

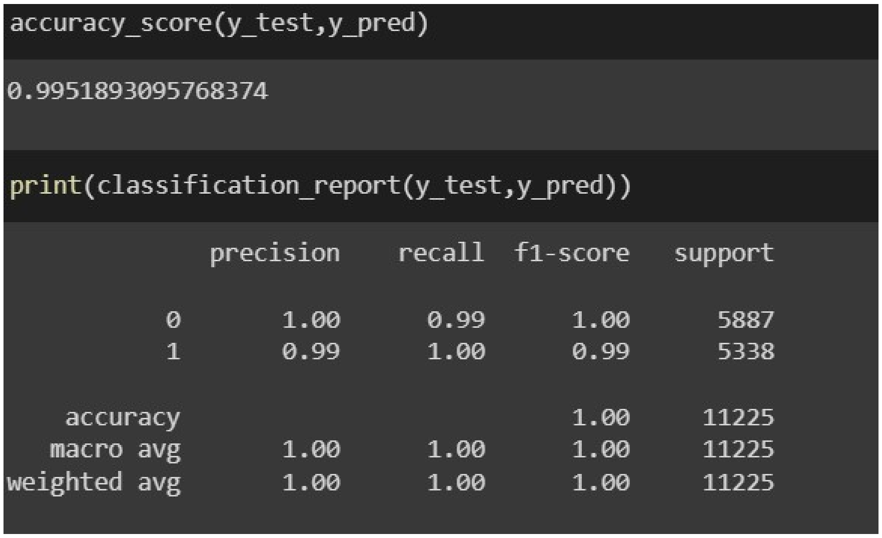
**Fig 20: Initializing Sequential model**

## 6.6ACCURACY

When performing classification predictions, there's two types of outcomes that could occur.

**Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

$$\text{Accuracy} = \frac{\text{Number of correct predic}}{\text{Total number of predicti}}$$



```
accuracy_score(y_test,y_pred)

0.9951893095768374

print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

           0       1.00      0.99      1.00      5887
           1       0.99      1.00      0.99      5338

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225
```

**Fig 21: Accuracy of trained model**

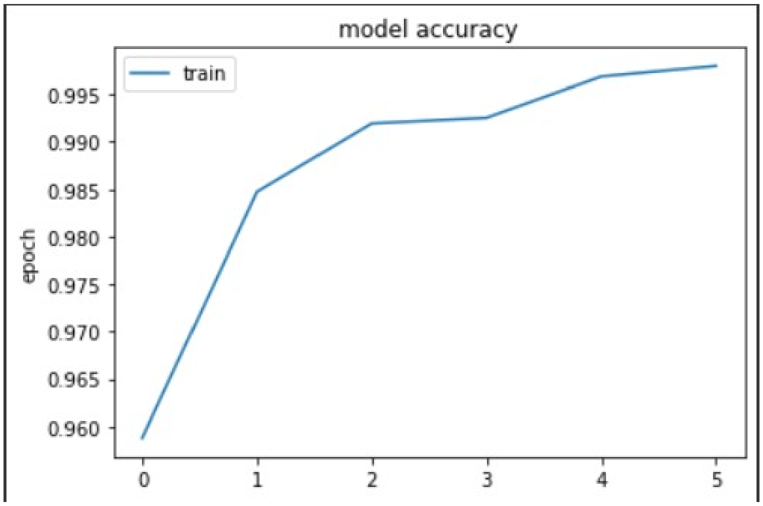


**Fig 22: Accuracy chart**

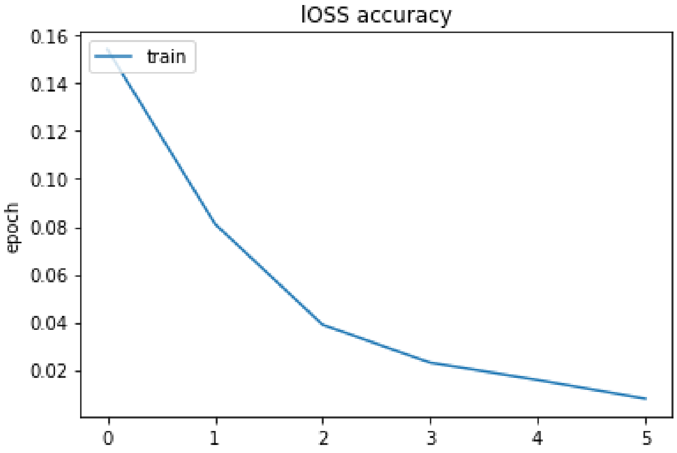


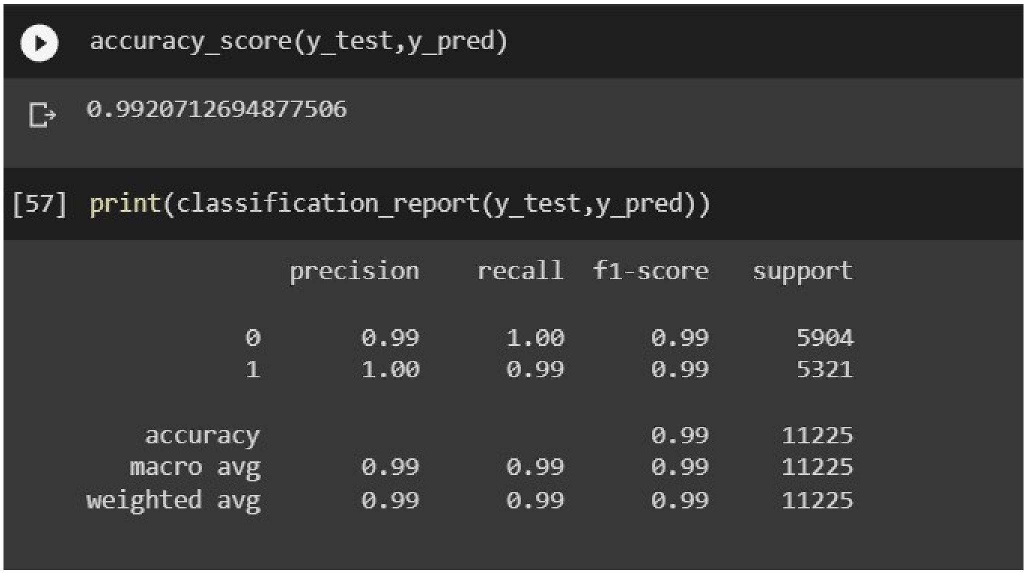
**Fig 23: Loss Chart**

Once the text is fed as input to the model, it predicts the news in the text file. In the above figure for the text the news recognized is ‘REAL’, which has an accuracy of 99%.

# CHAPTER-7

# EXPERIMENT RESULTS

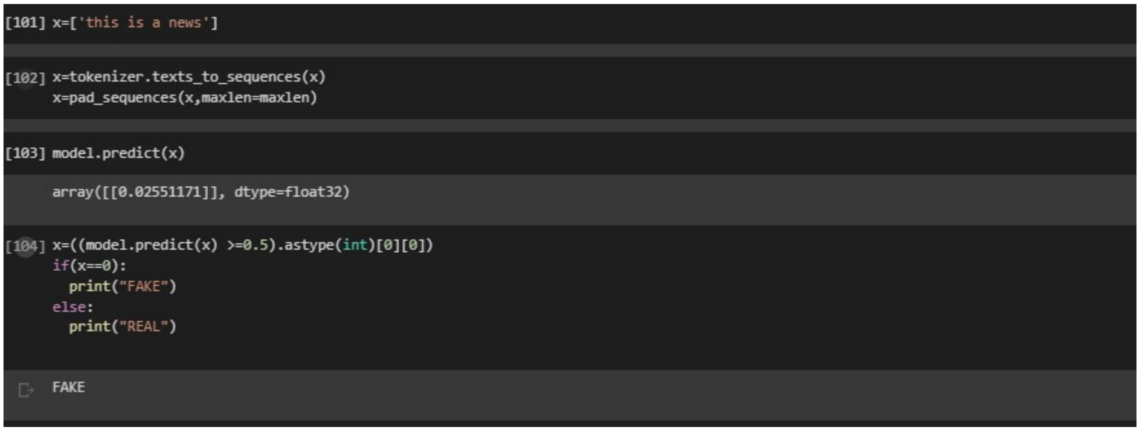
## 7.1 ACCURACY



```
accuracy_score(y_test,y_pred)

0.9920712694877506

[57] print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

           0       0.99      1.00      0.99      5904
           1       1.00      0.99      0.99      5321

    accuracy                           0.99     11225
   macro avg       0.99      0.99      0.99     11225
weighted avg       0.99      0.99      0.99     11225
```

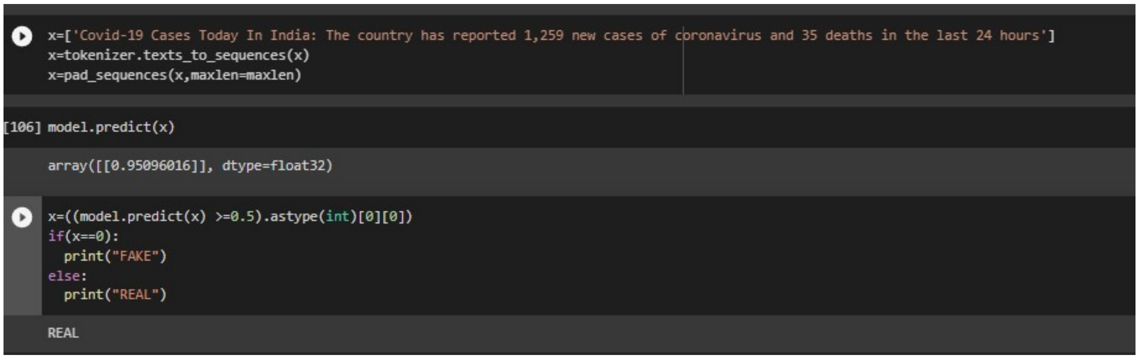
**Fig 24: Experiment Result and accuracy**

## 7.2PREDICTING THE NEWS



```
[101] x=['this is a news']

[102] x=tokenizer.texts_to_sequences(x)
      x=pad_sequences(x,maxlen=maxlen)

[103] model.predict(x)

      array([[0.02551171]], dtype=float32)

[104] x=((model.predict(x) >=0.5).astype(int)[0][0])
      if(x==0):
        print("FAKE")
      else:
        print("REAL")

      FAKE
```

**Fig 25: Example Predictions 1**



```
x=['Covid-19 Cases Today In India: The country has reported 1,259 new cases of coronavirus and 35 deaths in the last 24 hours']
x=tokenizer.texts_to_sequences(x)
x=pad_sequences(x,maxlen=maxlen)
```

```
[106] model.predict(x)
```

```
array([[0.95096016]], dtype=float32)
```

```
x=((model.predict(x) >=0.5).astype(int)[0][0])
if(x==0):
  print("FAKE")
else:
  print("REAL")
```

```
REAL
```

Fig 26: Example Predictions 2

# CONCLUSION

In recent years, deceptive content such as fake news and fake reviews, also known as opinion spams, have increasingly become a dangerous prospect for online users. Fake reviews have affected consumers and stores alike. Furthermore, the problem of fake news has gained attention in 2018 onwards, especially in the aftermath of the last U.S. presidential elections.

It turns out this architecture works well for fake news detection. The advantage of LSTM is that the performance time is much shorter, while the performance remains the same. This architecture is useful in other Natural Language Classification tasks as well. The model gives good results in Toxic comment classification as well as sentiment analysis.

# FUTURE WORK

In this chapter, we present some open issues in fake news detection and future research directions. Fake news detection on social media is a newly emerging research area, so we aim to point out promising research directions from a data mining perspective. we outline the research directions in two categories: Data-oriented, Application-oriented.

Data-oriented: Data-oriented fake news research is focusing on different kinds of data characteristics, such as: dataset, temporal and psychological. From a dataset perspective, we demonstrated that there is no existing benchmark dataset that includes resources to extract all relevant features. A promising direction is to create a comprehensive and large-scale fake news benchmark dataset, which can be used by researchers to facilitate further research in this area. From a temporal perspective, fake news dissemination on social media demonstrates unique temporal patterns different from true news.

Application-oriented: Application-oriented fake news re-search encompass research that goes into other areas beyond fake news detection. We propose two major directions along these lines: fake news diffusion and fake news intervention. Fake news diffusion characterizes the diffusion paths and patterns of fake news on social media sites. Some early re-search has shown that true information and mis information follow different patterns when propagating in online social networks and we will run model on the few other publicly available dataset.

# REFERENCES

1. Evaluating Machine Learning algorithms for Fake News Detection.(2017) Author: - Shloka Gilda
2. Fake News Detection on Social Media: A Data Mining Perspective.(2019) Author: - Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang and Huan Liu.
3. Media Rich Fake News Detection: A Survey.(2018) Author:- Shivam B. Parikh and Pradeep K. Atrey.
4. Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.
5. Ahmed H, Traore I, Saad S. (2017) “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham.
6. Tejaswini Yesugade1, Shrikant Kokate, Sarjana Patil, Ritik Varma, Sejal Pawar, ‘Fake News Detection using LSTM’ 2021
7. https://www.kaggle.com/dataset/clmentbisaillon/fake-and-real-news-dataset