# RUMOR DETECTION FROM SOCIAL MEDIA

# USING DEEP LEARNING

A PROJECT REPORT

Submitted by

**LISMY THOMAS [SCM21MCA-2012]**

to

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

*Master of Computer Application*



### **Department of Computer Science and Engineering**

**SCMS SCHOOL OF ENGINEERING AND TECHNOLOGY**

**(*Affiliated to APJ Abdul Kalam Technological University*)**

**VIDYA NAGAR, PALISSERY, KARUKUTTY**

**ERNAKULAM - 683582**

MAY 2023

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCMS SCHOOL OF ENGINEERING AND TECHNOLOGY**

**(*Affiliated to APJ Abdul Kalam Technological University*)**

**VIDYA NAGAR, PALISSERY, KARUKUTTY**

**ERNAKULAM - 683582**



**CERTIFICATE**

This is to certify that the report entitled **‘RUMOR DETECTION FROM SOCIAL MEDIA USING DEEP LEARNING’** submitted by **LISMY THOMAS** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Application is a bonafide record of the project work carried out by him/her under my guidance and supervision.

**PROJECT GUIDE HEAD OF THE DEPARTMENT**

Ms. Greshma P Sebastian Dr. Varun G Menon

**PROJECT COORDINATOR EXTERNAL EXAMINER**

Dr. Deepa K

# ACKNOWLEDGEMENT

I am greatly indebted to Dr.Anitha G.Pillai, Principal, SSET, Ernakulam and Dr. Varun G Menon, Head of department, Department of Computer Science and Engineering, SSET, who whole heartedly granted me the permission to do the project.

I would like to thank my guide, Ms. Greshma P Sebastian , and project coordinator, Dr. Deepa K, Department of Computer Science and Engineering, SSET who has given me valuable guidance and support all the way.

I would like to express my sincere gratitude to all the teachers of Computer Science Department who gave me moral and technical support. I would like to thank the supporting staff in the Computer lab whose dedicated work kept the lab working smoothly, thus enabling me to have access to various resources which helped me understand more about the project topic. I would also like to thank friends and family members for providing me with necessary resources and support. Last but not the least, I would like to thank God Almighty for helping me to do my project hassle Free.

# ABSTRACT

The spread of rumors and misinformation on social media can have serious consequences, from influencing public opinion to creating panic and confusion. This project addresses the problem of rumor detection from social media using deep learning techniques. Specifically, two models, a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network, are developed to differentiate between rumor and non-rumor news articles.

A dataset of labeled news articles is gathered and preprocessed into numerical vectors that can be fed into the models in order to accomplish this. On this dataset, CNN and LSTM models are trained and tested to see how accurate and effective they are at detecting rumors. The findings demonstrate that both models are able to distinguish between rumors and non-rumors with high accuracy.

# In general, this project demonstrates the potential of deep learning-based methods for identifying false information and rumors on social media. The proposed approach can possibly be utilized as a device to battle the spread of falsehood, upgrade public trust, and advance informed conversations and navigation.

# TABLE OF CONTENTS

# CONTENTS Page no.

[Acknowledgement i](#_Toc133495349)

[Abstract ii](#_Toc133495350)

List of Figures v

List of Tables vi

[Chapter 1.Introduction 1](#_Toc133495353)

[1.1. General Background](#_Toc133495353) 1

1.2. Objective 2

1.3. Scope 2

1.4. Organization of Report 2

Chapter 2.Literature Survey 3

Chapter 3.Proposed System 8

3.1. Architecture of the System 9

3.2. Dataset 10

3.3. Preprocessing 10

3.3.1. Data Cleaning 11

3.3.2. Removing Special Characters 11

3.3.3. Stop Word Removal 12

3.3.4. Lemmatization 12

3.4. Feature Extraction 12

3.5. Models 13

3.5.1. Long Short Term Memory 13

3.5.2. Convolutional neural network 15

3.6. Evaluation 17

3.6.1. Classification Report 17

3.6.2. Confusion Matrix 18

3.7. Libraries 18

Chapter 4. Experiments and Results 21

4.1. Classification Report 21

4.2. Confusion Matrix 22

4.3. User Interface 23

4.3.1. Flask 23

Chapter 5.Conclusion and Future Scope 25

[References vii](#_Toc133495358)

# 

# LIST OF FIGURES

# No. Title Page no.

# 3.1 General Diagram 8

# 3.2 Proposed System 9

# 3.3 Fake News Dataset 10

# 3.4 Preprocessing Steps 11

# 3.5 Single LSTM Cell Architecture 14

# 3.6 Simple CNN Architecture 16

# 4.1 Confusion Matrix of LSTM 22

# 4.2 Confusion Matrix of CNN 22

# 4.3 Home User Interface 23

# 4.4 Result User Interface 24

# LIST OF TABLES

# No. Title Page no.

# 4.1 Classification Report of LSTM 21

# 4.2 Classification Report of CNN 21

# CHAPTER-1

# INTRODUCTION

## **1.1 GENERAL BACKGROUND**

Machine learning, a sub-class of artificial intelligence. It is self-learning based on algorithms that mean the system learns from its experience. For instance, the type of data given input to the system learns the pattern and responds from its learning at the output. Deep learning is a subgroup of machine learning. Deep is the term that refers to several layers in between the input and output of a neural network [1]. The term "deep" refers to the use of multiple layers of artificial neurons to create a hierarchical representation of the data, which allows for more complex and abstract features to be learned.

Natural Language Processing (NLP) is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications [2]. NLP has become increasingly important in the era of big data, where vast amounts of textual data are generated daily. It assists developers in organizing knowledge for tasks like translation, automatic summarization, text classification, Named Entity Recognition (NER), speech recognition, relationship extraction, and topic segmentation [3].

Rumor detection from social media using deep learning involves developing models that can automatically identify and classify rumors from social media data. This field has gained increasing attention due to the negative impact of rumors on society and the need to combat the spread of disinformation. Deep learning techniques are used to extract meaningful features from large amounts of text data, aiding in the development of accurate rumor detection models.

## **1.2. OBJECTIVE**

Rumor detection on social media using deep learning aims to develop automated systems that can accurately and efficiently detect and classify rumors in social media posts. These systems aim to help individuals, organizations, and governments quickly identify and respond to rumors before they can cause harm. The goal is to mitigate the negative impact of rumors by detecting them quickly and accurately, promoting a safer and more informed social media environment.

## **1.3. SCOPE**

The scope of "rumors detection from social media using deep learning" project involves the exploration of advanced deep learning models and techniques, such as attention mechanisms and graph neural networks, to improve the accuracy and reliability of the system. Additionally, the integration of natural language generation techniques to generate fact-based responses to rumors can also be explored.

## **1.4. ORGANIZATION OF REPORT**

The report is divided into five chapters. The report's first chapter deals with the general background, objective and scope of the project. The second chapter contains various literature reviews related to the project. The third chapter discusses the proposed system's detailed architecture and operation. The fourth chapter contains the experiments and results . The conclusions are summarized in the final chapter. The future scope of the given project is also added in the last chapter. Finally the references are given in the last pages.

# CHAPTER-2

# LITERATURE SURVEY

The study by Zhenyu He, Ce Li, Fan Zhou, and Yi Yang [4] focused on detecting rumors on social media with event augmentations. The authors used the BERT model in combination with algorithms such as LSTM, CNN, and CNN-LSTM to train their model. The results showed that their approach achieved F1 scores ranging from 0.779 to 0.860 on various benchmark datasets, indicating its effectiveness in detecting rumors on social media. This study highlights the importance of incorporating event augmentations in rumor detection and demonstrates the potential of using deep learning models for this task. The findings could have implications for improving the accuracy of rumor detection systems on social media platforms, ultimately helping to reduce the spread of misinformation.

Akshi Kumar and Saurabh Raj Sangwan[5] conducted a study on detecting rumors on social media using machine learning techniques. Their methods included the use of bag of words, TF-IDF, and n-grams, while the algorithms they employed were Random Forest (RF), Naive Bayes (NB), and Support Vector Machine (SVM). The authors obtained an accuracy ranging from 88% to 90% in detecting rumors on social media, indicating the effectiveness of their approach. The study highlights the potential of using machine learning techniques to detect rumors and misinformation on social media platforms. The findings could have implications for improving the accuracy of existing rumor detection systems and could aid in reducing the spread of false information on social media.

According to a study conducted by Pathak, Ajeet Ram [6] titled "Analysis of Techniques for Rumor Detection in Social Media," the author analyzed various techniques for detecting rumors in social media. The study used three different methods for text representation: Bag of Words (BoW), TF-IDF, and Word2Vec, along with six different classification algorithms: SVM, LSTM, RNN, RF, DT, LR, and NB.

The results showed that the best performing algorithm for rumor detection was LSTM, with an accuracy of 92%. This is significant because rumor detection in social media is a challenging task that requires accurate and efficient algorithms. The study's findings suggest that LSTM, a type of neural network algorithm, is highly effective for detecting rumors in social media due to its ability to process sequential data and capture long-term dependencies between words.

The article "Rumor Detection On Social Media: Datasets, Methods And Opportunities" by Quanzhi Li [7] and colleagues explores the topic of detecting rumors on social media. The authors describe various datasets and methods that have been used for this task, including the use of linguistic analysis tools such as Linguistic Inquiry and Word Count (LIWC) and POS tagging. Additionally, the authors examine a range of algorithms that have been applied to this problem, such as SVM, RF, DT, LR, KNN, Neural Network, and Hidden Markov Model. The results of their study show that the best performing algorithm was SVM, which achieved an F1-score of 0.78. These findings highlight the potential of using machine learning and natural language processing techniques for detecting rumors on social media, which could help to mitigate the negative impact of false information on society.

In their paper, Alsaeedi and Al-Sarem [8] introduced a deep learning technique based on convolutional neural networks (CNNs) to detect rumors on social media. The authors used a dataset of Twitter posts related to a real-world rumor to train and test their model, and achieved an impressive accuracy of 92%. The results of their study highlight the effectiveness of using deep learning approaches, such as CNNs, for identifying and combating misinformation on social media platforms. These findings have important implications for various applications where detecting rumors is critical to prevent the spread of false information and minimize its negative impact on society.

The article titled "Rumor Detection System Using Machine Learning" by Anil Kr Dubey [9]and others presents a machine learning approach for detecting rumors. The authors used the Random Forest algorithm to train and test their model on a dataset of Twitter posts related to a real-world rumor. Their method achieved an accuracy of 85%, indicating its effectiveness in detecting rumors on social media platforms. The study highlights the potential of using machine learning techniques for identifying and combating misinformation, which is becoming increasingly prevalent on social media. The results suggest that the Random Forest algorithm is a promising approach for detecting rumors, and could be useful in a range of applications where the identification of false information is critical. Overall, this work contributes to the growing body of research aimed at mitigating the negative impact of rumors on society.

In the paper "Multimodal Arabic Rumors Detection" by Rasha M. and Albalawi [10], the authors explore a multimodal approach for detecting rumors in Arabic social media. They used a combination of text and image data and employed the Random Forest and CNN algorithms to train and test their model. Their method achieved an accuracy of 87%, indicating its effectiveness in detecting Arabic rumors using multiple modalities. The study demonstrates the potential of using multimodal approaches for identifying and combating false information on social media, and contributes to the growing body of research in this area.

In the paper titled "Detection of Fake News Using Machine Learning" by Smriti Agarwal [11] and colleagues, the authors explored the use of machine learning algorithms to detect fake news. They used TF-IDF, Word2Vec, and GloVe methods for feature extraction and trained various algorithms including SVM, Naive Bayes, RF, and Gradient Boosting to classify the data. The approach achieved an accuracy of 89%, indicating its effectiveness in detecting false information. The study highlights the potential of using feature extraction methods and machine learning algorithms for identifying and combating fake news, which is becoming a major concern in our society.

In their study, "Efficient Fake News Detection Mechanism Using Enhanced Deep Learning Model," Tahir Ahmad [12] and colleagues developed a deep learning model to detect fake news. The authors utilized the TF-IDF, word2vec, and GloVe methods for feature extraction and combined CNN and LSTM algorithms for classification. The approach achieved an accuracy of 94%, demonstrating the effectiveness of the model in identifying false information on social media platforms. The study highlights the potential of using deep learning techniques for detecting fake news and suggests that such models could be used as a powerful tool to combat the spread of false information.

The paper "Supervised Learning for Fake News Detection" by Julio CS Reis [13] and others provides a literature review on supervised learning techniques used for detecting fake news. The authors evaluated the performance of various algorithms such as Naive Bayes, Logistic Regression, and Support Vector Machines (SVM) in classifying news articles as true or false. The study shows that SVM performed the best among the algorithms and highlights the potential of using supervised learning techniques for detecting fake news. The authors also discussed the importance of feature engineering and the need for a large and diverse dataset for effective detection of fake news.

**CHAPTER-3**

**PROPOSED SYSTEM**

Rumor detection from social media using deep learning employs LSTM and CNN models that have been trained to differentiate between rumors and non-rumors. By providing the model with a large amount of labeled data and allowing it to learn patterns and features that distinguish rumors from non-rumors, this is accomplished. The labeled data consists of examples of both rumors and non-rumors, with the aim of training the model to accurately classify new, unseen data. By utilizing deep learning techniques, the model can make highly accurate predictions in real-time, which can help to prevent the spread of false information.

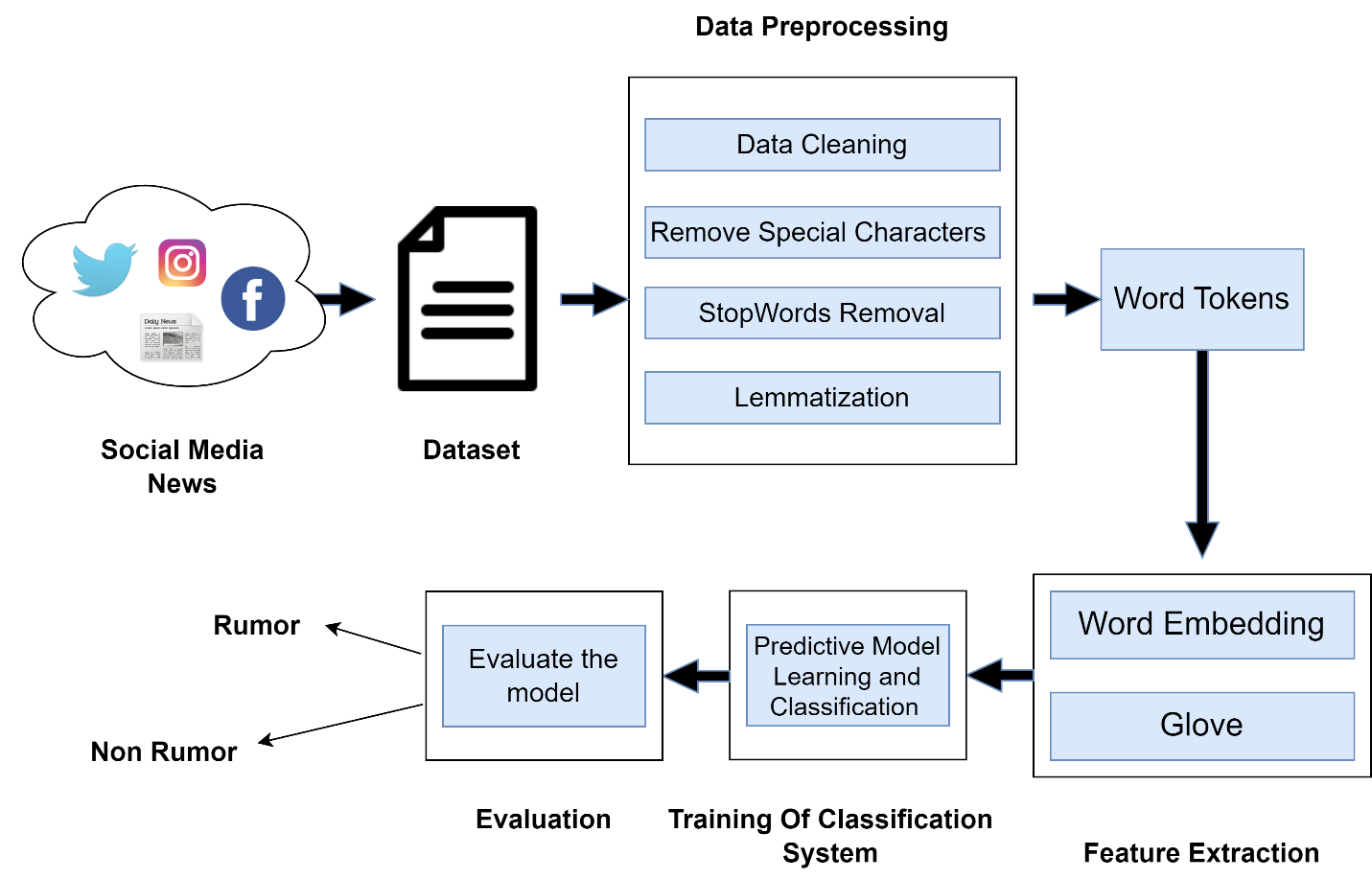


Fig 3.1: General diagram

## **3.1. ARCHITECTURE OF THE SYSTEM**

The proposed system, depicted in Figure 3.2, utilizes both LSTM and CNN models to classify text from the FAKENEWS dataset as rumor or non-rumor. To achieve this, the dataset undergoes preprocessing using NLP techniques such as text cleaning, lowercase conversion, stop word removal, and lemmatization, followed by feature extraction using a pre-trained GloVe model. This creates numerical representations of the text that can be input into both the LSTM and CNN models. The LSTM model uses the sequential nature of the text data to capture the temporal dependencies between words, while the CNN model leverages the convolution operation to capture local features within the text. Once the LSTM and CNN models are created, they are trained on the preprocessed data and can predict whether new text is a rumor or non-rumor. Overall, utilizing both LSTM and CNN models can improve the accuracy of the system's classification by leveraging the unique strengths of each model architecture.

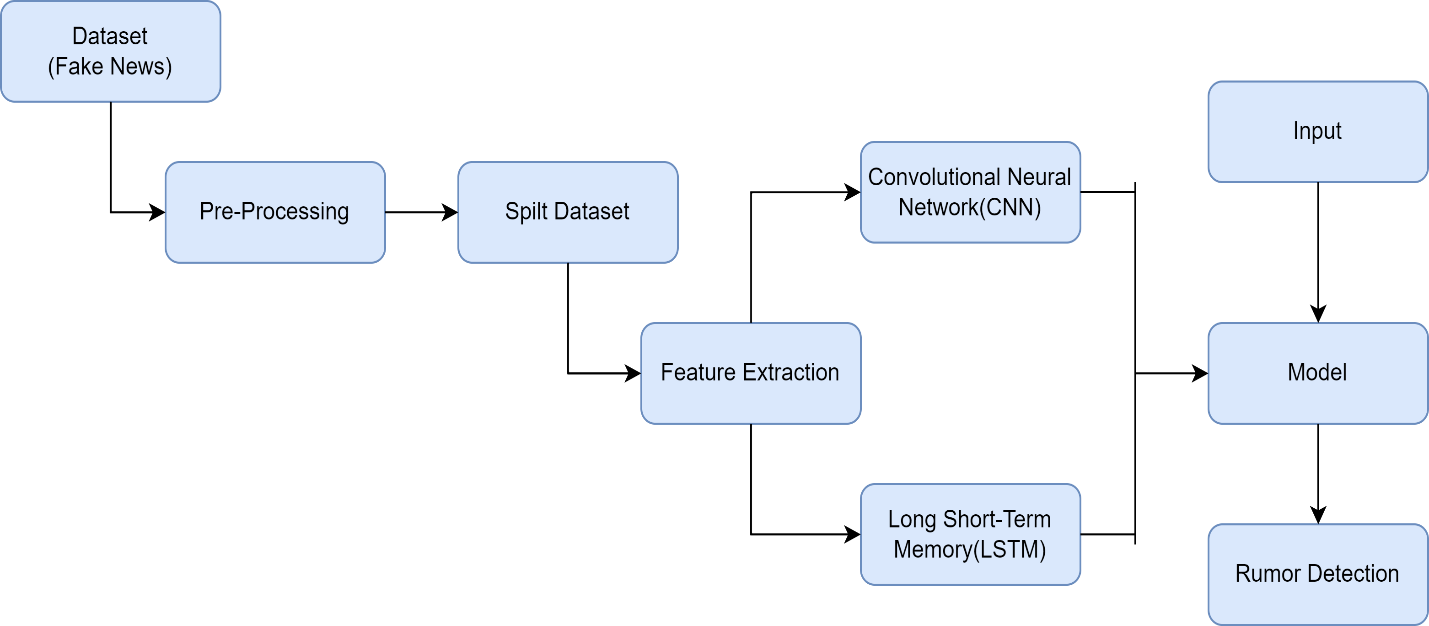
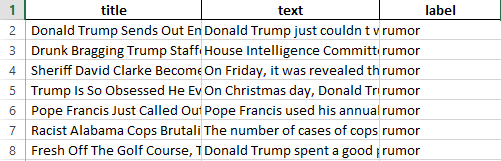
****

Fig 3.2.Proposed system

## **3.2. DATASET**

Two datasets, one from Kaggle and the other from a website with information on fake news articles [14], will be used in the project. Each dataset has features like "title," "text," and "label," with the labels "rumor" and "non-rumor" on them. The two datasets were joined to make a larger dataset, which was then split into training and testing sets. This dataset will be used to train and evaluate deep learning models for detecting and classifying rumors in social media.



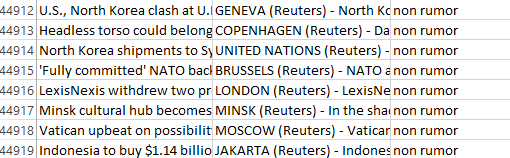


Fig 3.3: fake news dataset

## **3.3. PRE PROCESSING**

The text data preprocessing step performs several operations such as converting text to lowercase, removing special characters, removing stopwords, and lemmatization. These operations help clean up the data and make it suitable for feature extraction. After preprocessing is complete, the data is reconstructed and saved. Additionally, the data set is balanced so that each data class or category contains an equal number of samples. Both models are then preprocessed using the same method.

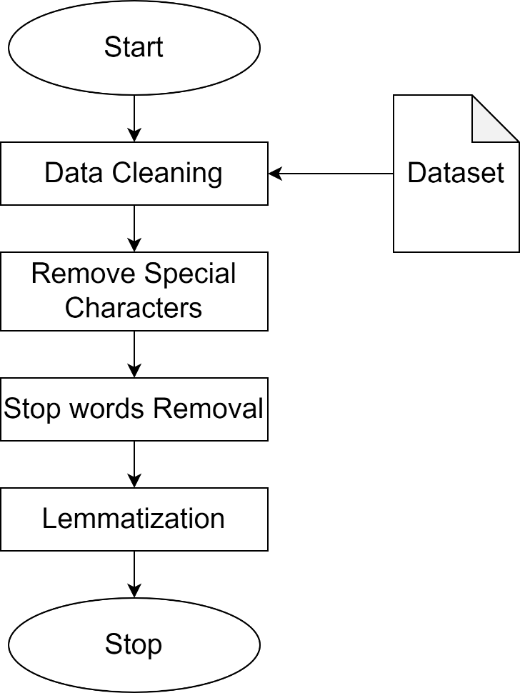


Fig 3.4.preprocessing steps

### **3.3.1. DATA CLEANING**

### Data cleaning is the process of identifying and correcting errors, inconsistencies, and inaccuracies in data to ensure accuracy and reliability. Tasks include removing duplicates, handling missing values, and equalizing the sample count. It's essential for accurate analysis and machine learning models.

### **3.3.2. REMOVING SPECIAL CHARACTERS**

Removing special characters involves removing non-alphanumeric characters like punctuation marks, symbols, and emoji’s from textual data. This is necessary to standardize data formats, improve data quality, and reduce noise in texts for getting better results. Special characters can cause problems in the process of data analysis or modeling. They can lead to errors or inconsistencies in the data.

### **3.3.3. STOP WORDS REMOVAL**

### Stop words are common words that do not carry much meaning [15], such as "the," "and," or "of," and can be safely removed without losing important information. Removing stop words can improve the performance and efficiency of NLP tasks, such as text classification, sentiment analysis etc. This can be achieved using pre-built libraries such as NLTK in Python.

### **3.3.4. LEMMATIZATION**

Lemmatization is the process of reducing words to their Root form, called a lemma [16].This helps in grouping together different forms of a word and improves the accuracy of text analysis. For example, the word "running" is pronounced "run," and the word "cats" is pronounced "cat." In natural language processing, lemmatization is frequently used to reduce the dimensionality of text data.

## **3.4. FEATURE EXTRACTION**

In the project, feature extraction is a critical step in processing raw text data from social media posts and transforming it into numerical vectors that can be used for deep learning models. The GloVe algorithm is a popular method for feature extraction in this context, as it can capture the relationships and meanings between words in the social media posts. In this project the model is based on trial and error experiments using LSTM and 300-dimensional word embedding features with Global Vector (GloVe) [17].

When using GloVe for NLP tasks, it is common to preprocess the text data by applying padding to make the sequences of equal length. Padding involves adding a certain number of tokens (usually zeros) to the beginning or end of shorter sequences so that all sequences have the same length. This is necessary for feeding the data into a neural network, which requires fixed-length inputs.

**GLOVE**

GloVe (Global Vectors for Word Representation) is an unsupervised machine learning algorithm used for generating word embedding’s. Glove technique that takes advantage of two different approaches: count-based and direct prediction [18]. Word embedding’s are numerical representations of words that capture their meanings and relationships with other words in a corpus of text. GloVe works by analyzing the co-occurrence statistics of words in a large corpus of text and learning vector representations of words that capture their semantic similarities. These word embedding’s can then be used for a wide range of natural language processing tasks such as language translation, sentiment analysis, and text classification. GloVe has become a popular method for generating word embedding’s and has been used in various research projects and commercial applications.

## **3.5. MODELS**

### **3.5.1. LONG SHORT-TERM MEMORY (LSTM)**

LSTM is an improved recurring neural network (RNN) architecture that uses a gating mechanism consisting of an input gate, forget gate, and output gate. These gates helps to determine whether the data in the previous state should be retained or forgotten in the current state. Hence, the gating mechanism helps the LSTM to address the issue of long-term information preservation and the vanishing gradient problem encountered by traditional RNNs [19]. The ability of LSTM to selectively store information and discard information, makes it a powerful tool for processing sequential data, such as text and speech.

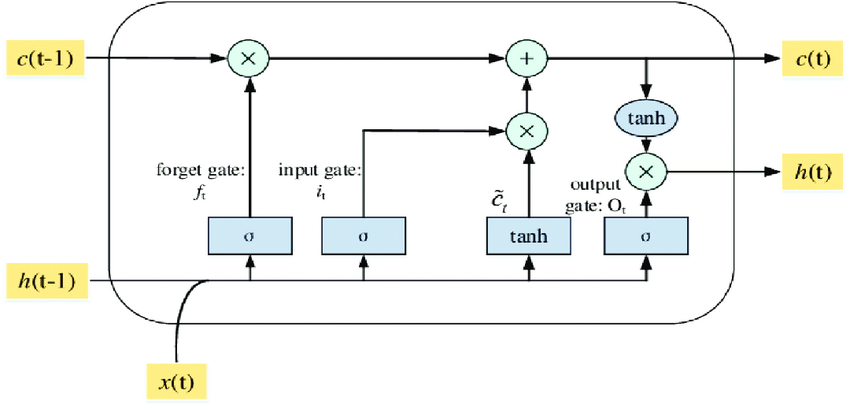


Fig.3.5.single LSTM cell architecture

In above figure 3.5 shows the architecture of single LSTM cell. A single LSTM (Long Short-Term Memory) cell architecture consists of three gates (input, forget, and output) and a memory cell that can selectively store or discard information [20].

The input gate controls which information from the input should be used. The forget gate controls what information should be discarded from the previous cell state. The output gate controls what information should be output to the next hidden state.

The gates of an LSTM network are designed to allow the network to selectively remember or forget information from previous timestamps, which helps to prevent the vanishing or exploding gradient problems commonly encountered in traditional RNNs.

### The model is a deep learning architecture consisting of two input layers, one for text data and another layer for title data. Each layer followed by an embedding layer that maps each word to a vector space. Then the layers are fed into a separate LSTM layers. The LSTM layers will convert the inputs to encoded sequence. The outputs of each LSTM layer is concatenated and create a combined feature representation. This combined representation is then fed into a fully connected layer with a sigmoid activation function. Which gives a prediction value between 0 and 1 indicating that the input text is a rumor. The model is optimized using binary cross-entropy loss and trained on a labeled dataset with the intention of maximizing prediction accuracy. Overall, the model is a powerful tool for rumor detection in text data, leveraging the strengths of both LSTM and embedding techniques to create an accurate and efficient classifier.

### **3.5.2. CONVOLUTIONAL NEURAL NETWORK (CNN)**

Convolutional Neural Networks (CNNs) are commonly used in text data for tasks such as sentiment analysis, text classification, and natural language processing. The CNN model works by using convolutional filters on text data to find local features that are indicative of rumors, like n-grams. The most significant features are then extracted from the feature maps by passing them through a max-pooling layer [21]. These highlights are leveled and taken care of into a completely associated layer for order. Due to their capacity to capture local patterns in the text data, CNNs have been shown to be effective at rumor detection.



Fig.3.6.Simple CNN Architecture

The model is a sequential deep learning architecture with an embedding layer that maps each word to a vector space and a 1D convolutional layer that filters the text data to find local features that suggest rumors. After that, a global max-pooling layer is fed through the feature maps to get the most important features. These features are then fed into a fully connected layer with a ReLU activation function, followed by a dropout layer to prevent over fitting. The output of the dropout layer is then fed into a final fully connected layer with a sigmoid activation function, outputting a prediction value between 0 and 1 indicating the likelihood that the input text is a rumor. The model is optimized using binary cross-entropy loss and trained on a labeled dataset in order to achieve maximum prediction accuracy. The model is tested on the test data to see how the model works with unseen data. Overall, the model is an effective tool for rumor detection in text data, leveraging the strengths of both convolutional and embedding techniques to create an accurate and efficient classifier.

**3.6. EVALUATION** After all of the models have been trained on the training dataset and the model has been evaluated. Several ways are used to assess the model's performance. The evaluation phase of a rumor detection system entails testing the model's performance against a set of known outcomes to determine its accuracy and reliability. Accuracy is used to calculate the percentage of correct predictions made by the model. To quantify the system's performance, classification reports and confusion matrices are often utilized evaluation metrics.

### **3.6.1. CLASSIFICATION REPORT**

A classification report is a tool used to evaluate the performance of a machine learning model by displaying precision, recall, F1 score, support and other metrics for each class in a classification problem [22].

ACCURACY: Accuracy is defined as the number of correct predictions divided by the total number of forecasts.

RECALL: The metric that is obtained by dividing the number of correctly identified positive instances (TP) by the sum of true positives and false negatives (FN) is known as the "recall" or "sensitivity."

PRECISION: The ratio of correctly classified (True Positive) positive samples to the total number of correctly or incorrectly classified positive samples is known as precision.

F1-SCORE: It is the weighted average of precision and recall.

SUPPORT: The number of instances for each class in the data.

### **3.6.2. CONFUSION MATRIX**

A confusion matrix is a table that summarizes the performance of a classification model and predicts and shows the correlation between the class label and the model's classification. Four different forms of data—True Negative (TN), True Positive (TP), False Positive (FP), and False Positive (FP)—can be derived from the matrix.

TP: Data points that have been identified as correct and positive.

FP: The data points that the model incorrectly categorized as positive.

FN: The model labelled FN as fake data items, but they are true.

TN: the negative data points that the model deemed to be false.

A confusion matrix is a visual depiction of how well a classification model works and may be used to analyze its accuracy, detect potential faults, and improve the model.

## **3.7. LIBRARIES**

The Jupyter Notebook platform is being utilized for the purpose of conducting a project on detecting rumors from social media. Jupyter Notebook is an open-source web-based platform used for data analysis, visualization, and collaboration. It allows users to create and share documents containing live code, equations, visualizations, and narrative text. The platform supports multiple programming languages, including Python.

The libraries used are,

**Numpy:** NumPy is a popular Python library for multi-dimensional array and matrix processing because it can be used to perform a great variety of mathematical operations [23].

**Pandas:** Pandas, which is based on NumPy, is in charge of creating high-quality data sets for machine learning and training. It utilizes both one-dimensional (series) and two-dimensional (Data Frame) data structures.

**Keras:** Keras is a Python library that has been specifically designed for the purpose of developing neural networks in machine learning models. It has the ability to run on top of TensorFlow, and can be utilized to train the neural networks. Keras is a highly versatile library that can be easily integrated with a variety of functions.

**Matplotlib:** Matplotlib is a Python library focused on data visualization and primarily used for creating beautiful graphs, plots, histograms, and bar charts. It is compatible for plotting data from SciPy, NumPy, and Pandas

**Tensorflow:** It is free and open-source software library for machine learning and artificial intelligence [24]. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

**NLTK (Natural Language Toolkit):** Library is a suite that contains libraries and programs for statistical language processing [25]. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response.

**Sklearn:** It is a Python library for machine learning, offering various data analysis and modeling tools such as classification, regression, and clustering [26]. It is built on top of other scientific computing libraries.

**Pickle:** The pickle library is a Python module for serializing and deserializing objects, converting them into byte streams, and reconstructing them, useful for saving complex data structures.

# CHAPTER-4

# EXPERIMENTS AND RESULTS

The proposed system “rumor detection from social media using deep learning “successfully able to predict rumor or not .In our study, we trained two deep learning models, a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) model. Both models achieved an impressive 98% and 99% accuracy on the test dataset and performance is evaluated.

## **4.1. CLASSIFICATION REPORT**

The classification report of LSTM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.98 | 1.00 | 0.99 | 4285 |
| 1 | 1.00 | 0.98 | 0.99 | 4168 |

Table 4.1. Classification report of LSTM

The classification report of CNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.99 | 1.00 | 0.99 | 4285 |
| 1 | 1.00 | 0.99 | 0.99 | 4168 |

Table 4.2. Classification report of CNN

## **4.2. CONFUSION MATRIX**

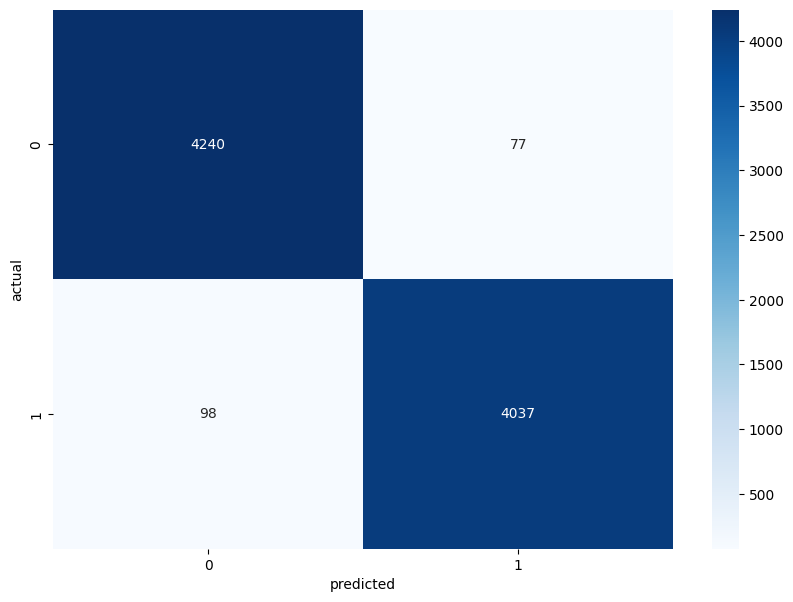
****

Fig 4.1.confusion matrix of LSTM

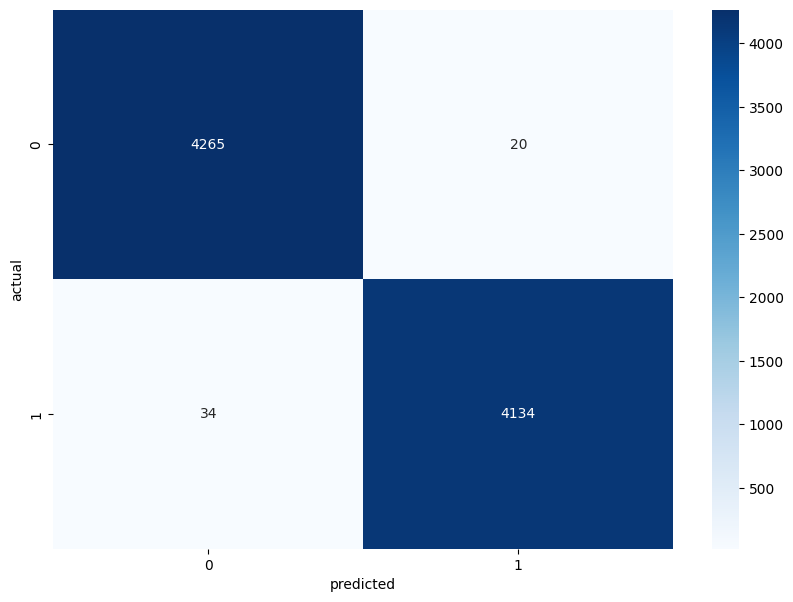


Fig 4.2.confusion matrix of CNN

## **4.3. USER INTERFACE**

Flask framework is used to create a user interface with two pages: a home page and a prediction page. On the home page, users can input data, while the prediction page displays the outcome as either a rumor or non-rumor.

### **4.3.1. FLASK**

Flask is a popular and widely-used Python-based micro web framework that allows developers to quickly build web applications. With its lightweight and flexible design, Flask simplifies web development by providing built-in tools and libraries for routing, request handling, and more. Its modular design also makes it highly customizable, allowing developers to add or remove components with ease.

Home.html



Fig 4.3.Home user interface

Result.html

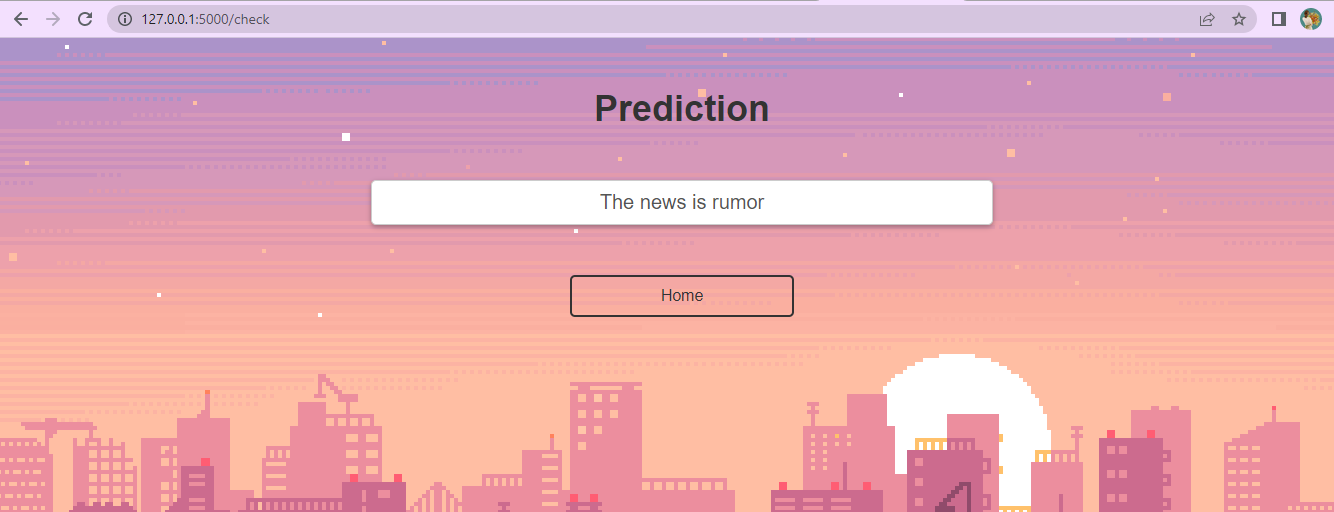


Fig 4.4.Result user interface

# CHAPTER-5

# CONCLUSION & FUTURE SCOPE

Deep learning algorithms for rumor identification in social media are an interesting field of research. With the growing prevalence of fake news and rumors on social media, it is critical to develop reliable and efficient automated detection systems. Deep learning techniques, such as convolutional neural networks and recurrent neural networks, have demonstrated significant promise in spotting rumors and fake news by analyzing textual and visual data from social media. However, there are still obstacles to overcome, such as the necessity for big annotated datasets and deep learning model interpretability. Despite these difficulties, ongoing research in this area shows considerable promise for enhancing the general quality and authenticity of content on social media platforms.

The future scope of rumor detection from social media using deep learning is vast and promising. Some potential areas of development and research include:

* Multiple language Rumor Detection: Currently, most rumor detection models are trained on data from English-speaking countries. However, social media is widely used across the globe in different languages. Future research could focus on training models to detect rumors in different languages using LSTM models.
* Identifying the source of rumors: LSTM-based models can be trained not only to detect rumors but also to identify the sources of those rumors. This can help in identifying the origin of fake news and stopping its spread.
* User-Level Rumor Detection: LSTM-based models can be used to identify users who are likely to spread rumors on social media. This can help in developing targeted interventions and strategies to prevent the spread of fake news and misinformation.
* Multimodal analysis: The integration of multiple data sources, such as text, images, and videos, can enhance the performance of deep learning models for rumor detection. Future research can explore the use of multimodal analysis techniques to improve the accuracy and robustness of these models.

Overall, the future scope of rumor detection from social media using deep learning is vast and holds great promise for improving the quality and credibility of information on social media platforms.

# REFERENCES

[1] Sharma, Neha, Reecha Sharma, and Neeru Jindal. "Machine learning and deep learning applications-a vision." Global Transitions Proceedings 2.1 (2021): 24-28.

[2] Liddy, Elizabeth D. "Natural language processing." (2001).

[3] https://www.javatpoint.com/nlp

[4] He, Zhenyu, et al. "Rumor detection on social media with event augmentations." Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021.

[5] Kumar, Akshi, and Saurabh Raj Sangwan. "Rumor detection using machine learning techniques on social media." International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2. Springer Singapore, 2019.

[6] Pathak, Ajeet Ram, et al. "Analysis of techniques for rumor detection in social media." ProcediaComputer Science 167 (2020): 2286-2296.

[7] Li, Quanzhi, et al. "Rumor detection on social media: datasets, methods and opportunities. “ arXiv preprint arXiv:1911.07199 (2019).

[8] Alsaeedi, Abdullah, and Mohammed Al-Sarem. "Detecting rumors on social media based on a CNN deep learning technique." Arabian Journal for Science and Engineering 45 (2020): 10813-10844.

[9] Dubey, Anil Kr, Apoorv Singhal, and Sarthak Gupta. "Rumor detection system using machine learning." Int Res J Eng Technol (IRJET) 7.05 (2020): 2395-0056.

[10] Albalawi, Rasha M., et al. "Multimodal Arabic Rumors Detection." IEEE Access (2023).

[11] Agarwal, Smriti, et al. "Detection of Fake News Using Machine Learning." Intelligent Systems and Smart Infrastructure: Proceedings of ICISSI 2022 (2023): 375.

[12] Ahmad, Tahir, et al. "Efficient Fake News Detection Mechanism Using Enhanced Deep Learning Model." Applied Sciences 12.3 (2022): 1743.

[13] Reis, Julio CS, et al. "Supervised learning for fake news detection." IEEE Intelligent Systems 34.2 (2019): 76-81.

[14]https://www.kaggle.com/competitions/fake-news/data https://github.com/KaiDMML/FakeNewsNet

[15] Uysal, Alper Kursat, and Serkan Gunal. "The impact of preprocessing on text classification." Information processing & management 50.1 (2014): 104-112.

[16] https://www.educative.io/answers/preprocessing-steps-in-natural-language-processing-nlp

[17] Sari, Winda Kurnia, Dian Palupi Rini, and Reza Firsandaya Malik. "Text Classification Using Long Short-Term Memory with GloVe." Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI) 5.1 (2019): 85-100.

[18] Dharma, EDDY MUNTINA, et al. "The accuracy comparison among Word2vec, Glove, and Fasttext towards convolution neural network (CNN) text classification." Journal of Theoretical and Applied Information Technology 100.2 (2022): 31.

[19] Jang, Beakcheol, et al. "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism." Applied Sciences 10.17 (2020): 5841.

[20] Pulver, Andrew, and Siwei Lyu. "LSTM with working memory." 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017.

[21] Jurtz, Vanessa Isabell, et al. "An introduction to deep learning on biological sequence data: examples and solutions." Bioinformatics 33.22 (2017): 3685-3690.

[22] Nadikattu, Ashok Kumar Reddy. "Machine Learning vs Deep Learning Models for Detecting Fake News: A Comparative Analysis on Fake-NewsNet Dataset." Available at SSRN 4382241 (2023).

[23] https://levelup.gitconnected.com/top-10-python-libraries-and-5-best-books-for-data-science-fa0d0cf171a6

[24] https://www.tensorflow.org/resources/libraries-extensions

[25] https://realpython.com/nltk-nlp-python/

[26] https://docs.python.org/3/library/pickle.html