A
Mini Project
on

DETECTING FAKE NEWS USING MACHINE LEARNING ALGORITHMS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

by

A. MAHALAKSHMI (227R5A6706)

S. SHASHANK (217R1A6752)

P.SAI CHANDU REDDY (217R1A6744)

Under the Guidance of

Dr. M. KISHORE KUMAR

Professor CSE(DATA SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

Accredited by NBA & NAAC with 'A' Grade

Approved by AICTE, New Delhi and JNTU, Hyderabad.

2024-2025

DEPARTMENT OF COMPUTER SCIENC&ENGINEERING

(DATA SCIENCE)

CERTIFICATE

This is to certify that the project entitled **"DETECTING FAKE NEWS USING MACHINE LEARNING ALGORITHMS"** being submitted by **A. MAHALAKSHMI (227R5A6706), S. SHASHANK (217R1A6752) & P.SAI CHANDU REDDY (217R1A6744)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (Data Science) to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-2025.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr.M.KISHORE KUMAR
Professor CSE(DS)
INTERNAL GUIDE

Dr. K. MURALI
HOD CSE(DS)

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ___

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide

Dr.M. KISHORE KUMAR, Professor CSE(DS) for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessings, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Mr. Sanjib Kumar Nayak, Mrs. V. Prema Tulasi, Mr. B. Ramesh, Mr. A. Veerender, Mrs. V. Sandya, Ms. Sandhyarani** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. MURALI,** Head, Department of Computer Science and Engineering (DS) for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,**Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take the opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

A.MAHALAKSHMI (227R5A6706)

S. SHASHANK (217R1A6752)

P.SAI CHANDU (217R1A6744)

ABSTRACT

Internet is one of the most important interventions and large numbers of its persons are its users.These persons use this for different purposes. There are different social media platforms that are accessible to these users. Any user can make a post or spread the news through these online platforms. These platforms do not verify the users or their posts. So some of the users try to spread fake news through these platforms. These fake news can be a propaganda against an individual, society, organization or political party. A human being is unable to detect all these fake news. So there is a need for machine learning classifiers that can detect these fake news automatically.

World is changing rapidly. No doubt we have a number of advantages of this digital world but it also has its disadvantages as well. There are different issues in this digital world. One of them is fake news. Someone can easily spread a fake news. Fake news is spread to harm the reputation of a person or an organization. It can be a propaganda against someone that can be a political party or an organization. There are different online platforms where the person can spread the fake news. This includes the Facebook, Twitter etc.

Machine Learning is a part of artificial intelligence that helps that can help in making the

Systems that can learn and perform different actions. A variety of machine learning algorithms are available that include the supervised, unsupervised, reinforcement machine learning algorithms. The algorithms first have to be trained with a data set called train data set. After the training, these algorithms can be used to perform different tasks. Machine learning is using in different sectors to perform different tasks. Most of the time machine learning algorithms are used for prediction purpose or to detect something that is hidden.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SCREENSHOTS

Chapter 1

INTRODUCTION

1.1 PROJECT SCOPE

The project aims to develop a system for detecting fake news using machine learning algorithms. With the rapid increase in the spread of misinformation on digital platforms, the need for reliable tools to identify and filter out fake news has become crucial. This project will explore various machine learning techniques such as natural language processing (NLP) for text analysis, sentiment analysis, and classification algorithms like Support Vector Machines (SVM), Decision Trees, Random Forest, and Neural Networks to distinguish between fake and authentic news. The system will be trained on large datasets containing both verified news and fabricated articles, allowing the model to learn patterns and linguistic cues typical of misinformation. The goal is to create an accurate, efficient, and scalable solution that can assist in combating the proliferation of false information across media platforms, contributing to informed decision-making and enhancing the credibility of online content.

1.2 PROJECT PURPOSE

The purpose of the "Detecting Fake News Using Machine Learning Algorithms" project is to combat the growing issue of misinformation and disinformation in digital media. As false news spreads rapidly, it can have significant negative impacts on public opinion, political stability, and social trust. This project seeks to develop a reliable and automated solution that leverages machine learning to analyze and classify news content, identifying misleading or fabricated information. By implementing algorithms that can learn from patterns in authentic and fake news, the system aims to enhance the credibility of onlinenews sources, help users make informed decisions, and reduce the social and economic consequences of fake news. Ultimately, the project strives to create a tool that aids in maintaining the integrity of information shared across various digital platforms.

1.3 PROJECT FEATURES

The project will incorporate several key features to effectively detect fake news using machine learning algorithms. First, it will utilize data preprocessing techniques, including text cleaning, tokenization, and normalization, to prepare news articles for analysis. A natural language processing (NLP) pipeline will be integrated to extract linguistic features such as word frequency, syntax patterns, and semantic relationships. Additionally, the system will include a sentiment analysis component, enabling the identification of emotional bias or manipulative language often found in fake news.

Moreover, the system will provide real-time detection capabilities, allowing users to input new articles and receive instant feedback on their authenticity. A user-friendly interface will display the classification results, along with confidence scores, to help users understand the reliability of the news. Lastly, the project will offer continuous model updates through regular retraining using new datasets, ensuring the system adapts to evolving misinformation tactics.

Chapter 2

SYSTEM ANALYSIS

The system analysis for detecting fake news using machine learning algorithms involves understanding the problem, requirements, and technical aspects to ensure effective implementation. First, the problem domain focuses on identifying fake news from legitimate sources in an era where online misinformation spreads rapidly, often causing public confusion or harm.

The system will require data collection, where large datasets of both fake and genuine news articles are gathered. This data will serve as the foundation for training machine learning models. Data preprocessing will involve cleaning, tokenizing, and normalizing the text to prepare it for feature extraction.

The machine learning component will involve the selection of algorithms such as Naive Bayes, Support Vector Machines (SVM), Random Forest, or deep learning models like Long Short-Term Memory (LSTM) networks to perform classification. These models will be trained on the preprocessed data to identify patterns, such as writing style, sentiment, and word choice, which can distinguish fake news from legitimate sources.

2.1 PROBLEM DEFINITION

The problem of detecting fake news has become increasingly significant due to the widespread use of digital media and social platforms, where misinformation can quickly go viral and influence public opinion. Traditional methods of fact-checking are often manual, slow, and insufficient to keep up with the volume and speed at which false information spreads. This creates a need for automated systems capable of identifying fake news in real-time. The challenge lies in accurately distinguishing between truthful and misleading content, as fake news articles are often crafted to appear credible. Furthermore, fake news detection is complicated by the diversity of topics, writing styles, and intentions behind misinformation. This project seeks to address these issues by leveraging machine learning algorithms that can analyze textual features, detect patterns in language, and classify news articles as either fake or real. The objective is to build a scalable solution that improves the efficiency and accuracy of fake news detection, helping to curb the spread of misinformation and its negative impact on society

2.2 EXISTING SYSTEM

Sentiment analysis is a classification problem where the main focus is to predict the polarity of words and then classify them into positive or negative sentiment. Classifiers used are of mainly two types, namely lexicon-based and machine learning based. The former include Senti Word Net and Word Sense Disambiguation while the latter include Multinomial Naive Bayes (MNB), Logistic Regression (LR), Support Vector Machine (SVM) and RNN Classifier

2.2.1 LIMITATIONS OF EXISTING SYSTEM

DISADVANTAGES:

- ❖ Sentiment Analysis which means to analyze the underlying emotions of a given text using Natural Language Processing
- ❖ ML algorithm using some labelled data and then use that model to predict a class for a new text.

Algorithm: Naive Bayes Classifiers, linear regression.

2.3 PROPOSED SYSTEM

We propose a scalable community-based probabilistic framework to model the spreading of news about events in online media. Our approach exploits the latent community structure in the global news media and uses the affiliation of the early adopters with a variety of communities to identify the events widely reported in the news at the early stage of their spread. The time complexity of our approach is linear in the number of news reports. It is also amenable to efficient parallelization. To demonstrate these features, the inference algorithm is parallelized for message passing paradigm and tested on the Rensselaer Polytechnic Institute Advanced Multiprocessing Optimized System, one of the fastest Blue Gene/Q supercomputers in the world. Thanks to the community- level features of the early adopters, the model gains an improvement of 20% in the early detection of the most massively reported events compared with the feature-based machine learning algorithm. Its parallelization scheme achieves orders of magnitude speedup.

2.3.1 PROPOSED APPROACH

The proposed approach for developing a machine learning-based Twitter cyberbullying detection system involves several key steps: pre-processing, feature extraction, and algorithm selection. Pre-processing involves using the Natural Language Toolkit (NLTK) to prepare the text data. Initially, the text is tokenized into sentences using the Punkt Sentence Tokenizer, followed by breaking these sentences into words with four different tokenizers: Whitespace Tokenizer, WordPunct Tokenizer, TreebankWord Tokenizer, and PunctWord Tokenizer. Next, the text is converted to lowercase to standardize the input, transforming "Hey There" into "hey there." An essential part of this stage is the removal of stop words—common but unhelpful words such as \t, https, and \u—which are filtered out using NLTK. Finally, the WordNet lemmatizer is employed to find and link synonyms, enhancing the model's understanding of the text. Feature extraction transforms the processed data into a suitable format for machine learning algorithms. The Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer is used to extract features from the data, creating a feature list that includes the polarity classification of each text (bullying or non-bullying).

In the algorithm selection step, supervised binary classification algorithms, specifically Support Vector Machine (SVM) with a linear kernel and Naive Bayes, are utilized to automatically detect bullying in social media texts. Both algorithms calculate probabilitiesfor the two classes (bullying and non-bullying) but SVM demonstrates superior performance over Naive Bayes when evaluated on the same dataset. The effectiveness of both models is assessed through a classification report that includes metrics such as accuracy, recall, F-score, and precision.

2.3.2 ADVANTAGES OF THE PROPOSED SYSTEM

ADVANTAGES :

- It can join both substance based and synergistic separating approaches.
- The proposed framework utilizes changed cosine likeness technique which is more valuable because of the taking away the relating client normal from every co-appraised pair.

Algorithm: Fake news, SVM, Naive Bayes, Machine learning, Social media, Twitter APJ, Sentimation analysis

2.4 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.4.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements:

- System: Intel Core i3
- Ram: 8GB(min)
- Hard disk:1TB
- Monitor:15LED

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specify the logical characteristics of each interface and software components of the system. The following are some software requirements:

- OPERATING SYSTEM : Windows 10
- CODE LANGUAGE : Python
- TOOLS : PyCharm, Visual Studio Code
- Database : SQLite

2.6 PYTHON

Python is a high-level, interpreted programming language known for its readability and simplicity. Developed by Guido van Rossum and first released in 1991, Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Key features of Python include:

1. **Readable Syntax**: Python’s syntax is clear and concise, making it easy for beginners to learn and for developers to maintain code.
2. **Dynamic Typing**: Variables in Python do not require an explicit declaration to reserve memory space, allowing for flexible code.
3. **Extensive Libraries**: Python boasts a rich standard library and a vibrant ecosystem of third-party packages, making it suitable for various applications, from webdevelopment to data analysis and machine learning.
4. **Cross-Platform Compatibility**: Python runs on various operating systems, including Windows, macOS, and Linux.
5. **Community and Support**: A large and active community provides extensive resources, tutorials, and frameworks, enhancing the language's versatility.

Libraries and Frameworks

- **Standard Library:** Comes with a robust standard library that includes modules for file I/O, system calls, data manipulation, and more.
- **Popular Libraries:**
  - NumPy: For numerical computing.

    - Pandas: For data manipulation and analysis.
    - Matplotlib: For data visualization.
    - Django and Flask: Frameworks for web development.
    - TensorFlow and PyTorch: For machine learning and AI.

Advantages

- **Ease of Learning:** Suitable for beginners and experienced programmers alike.
- **Versatility:** Applicable in various domains, from web apps to scientific computing.
- **Strong Community Support:** Resources are readily available for troubleshooting and learning.

Chapter 3

ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

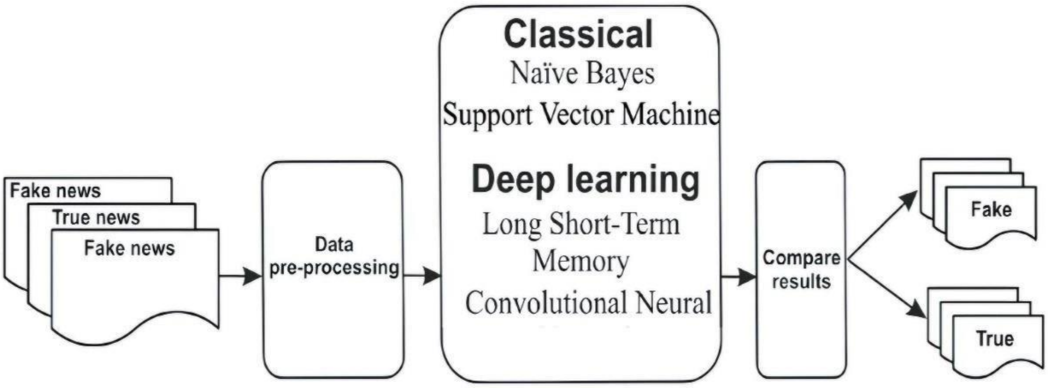


Fig 3.1 Project Detecting Fake News Using Machine Learning Algorithms

A machine learning framework for classifying news as either fake or true. The process begins with an input dataset that contains examples of both fake and true news. This dataset is subjected to data pre-processing, a crucial step where text data is cleaned and prepared for model input. This might involve removing irrelevant words, tokenizing the text, and converting it into a numerical format suitable for machine learning algorithms.

After pre-processing, the data is passed to two types of models: classical machine learning models and deep learning models. The classical models include Naïve Bayes, a probabilistic classifier that works well for text classification tasks by assuming independence between features, and Support Vector Machines (SVM), which attempt to find the optimal hyperplane to separate fake news from true news in high-dimensional space. On the other hand, deep learning models, such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN), are more sophisticated and capture deeper, complex patterns within the data. LSTMs excel at analyzing sequential data like news articles, maintaining long-term dependencies, while CNNs, though traditionally used for image data, are adapted to capture local patterns in text, such as word combinations.

Once the models process the data, their results are compared. This comparison step may involve ensemble learning, where the predictions from multiple models are combined for greater accuracy, or model validation, where metrics like precision, recall, and F1-score are used to assess performance. Finally, based on this comparison, the system makes a decision and classifies each article as either fake or true news. By blending classical and deep learning techniques, this architecture offers a robust and efficient way of identifying misinformation in news articles.

The architecture uses a hybrid approach, employing both classical and deep learning models to detect fake news. Data pre-processing is essential for preparing the text, after which the models learn from the input data to make predictions. The results from both classical and deep learning models are compared and used to classify news articles into "Fake" or "True." This system benefits from the strengths of different models, resulting in improved accuracy in detecting misinformation.

3.2 USECASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model. A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

The provided use case diagram illustrates the interactions between the system and its users, specifically the Service Provider and the Remote User, within the context of a cyberbullying detection system. The Service Provider is responsible for managing datasets, training and testing predictive models, and providing results, while theRemote User interacts with the system to register, log in, view predictions, and accessvarious analytical tools.

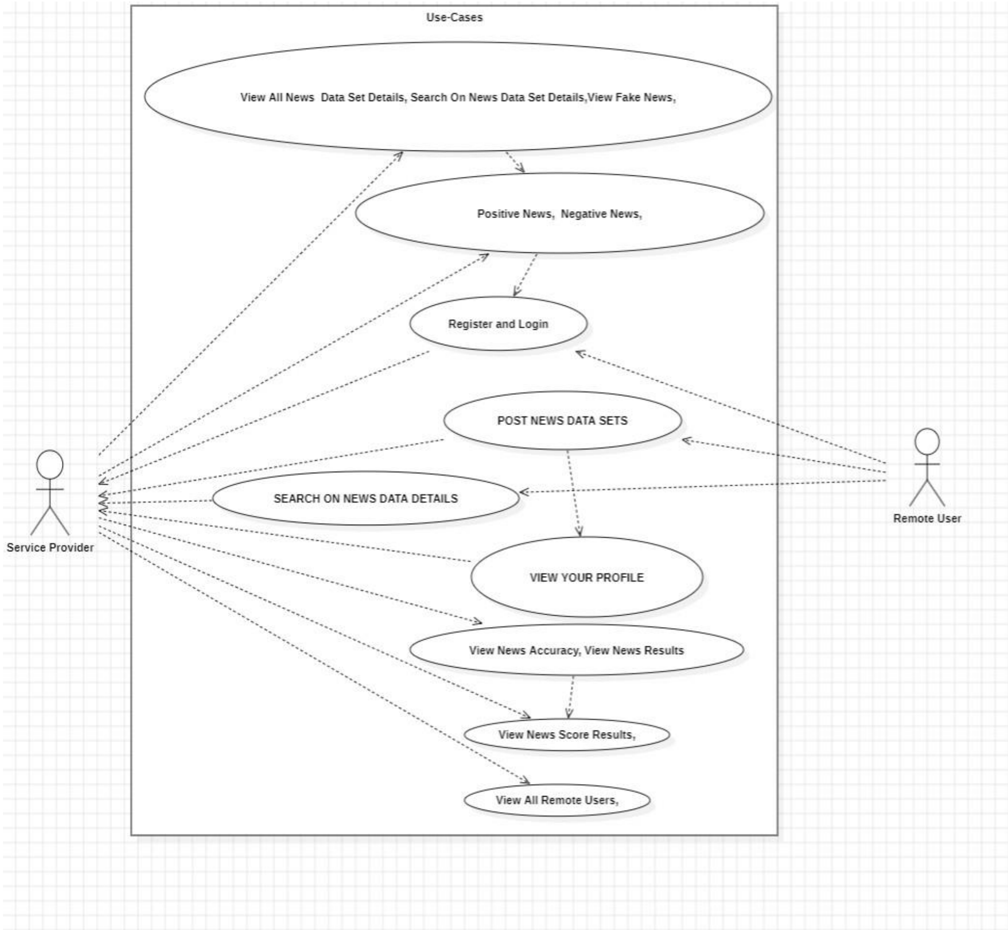


Fig 3.2 Use Case Diagram for Detecting Fake News using Machine Learning Algorithm

The use case diagram represents a user's interaction with a fake news detection system, showcasing the main functionalities available to the user. It begins with the User Login Screen, where the user must authenticate to access the system. This step ensures that only registered or authorized users can interact with the system, providing a layer of security and control over access.

Once logged in, the user is presented with the option to Upload News Articles. This function allows the user to input one or multiple news articles into the system for analysis. The uploaded articles may come from various sources, and the system is designed to assess whether they are legitimate or fabricated. This is a crucial step as the system relies on this data to perform its key function—evaluating the credibility of the news.

3.3 CLASS DIAGRAM

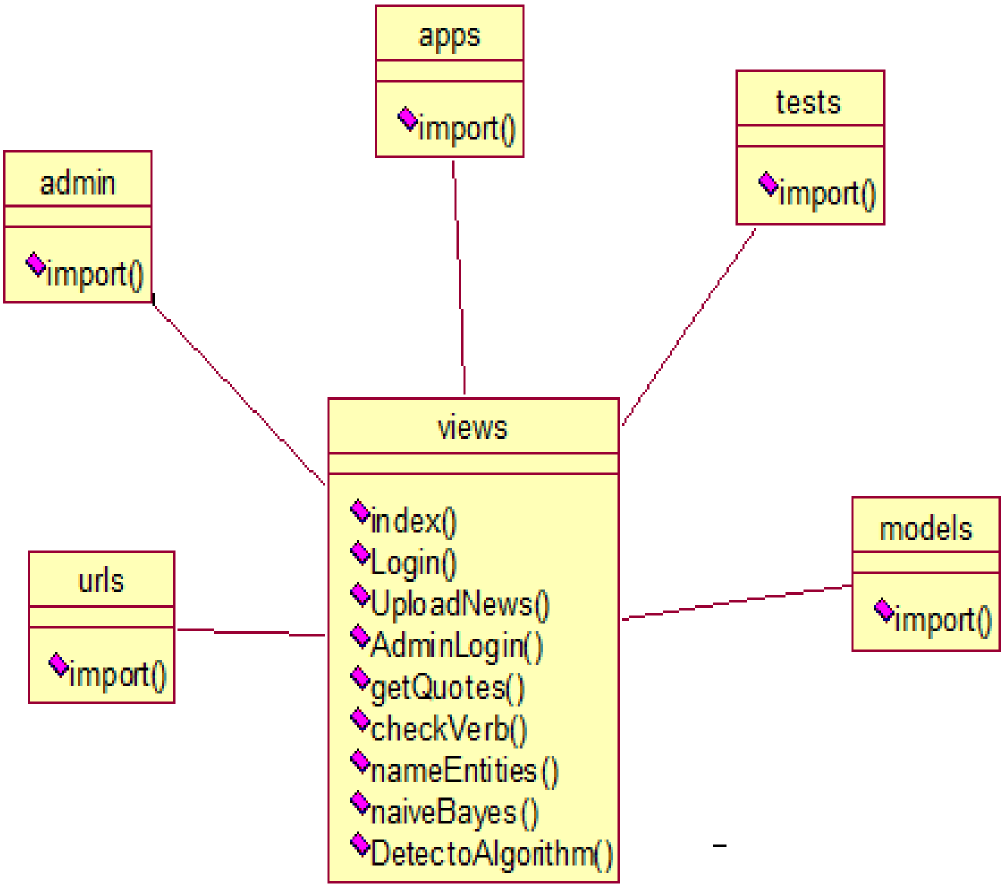


Fig 3.3:Class Diagram for Detecting Fake News

A UML (Unified Modelling Language) class diagram for detecting fake news using a machine learning algorithm typically represents the key components and their relationships within the system. The diagram would include classes such as News Dataset, Preprocessing, Feature Extraction, Machine LearnigModel, and Evaluation. The News Dataset class is responsible for managing the dataset, including loading and dividing it into training and testing sets. The Preprocessing class handles tasks like cleaning the text, removing stop words, and tokenizing the news articles. The Feature Extraction class, often using techniques like TF-IDF or word embeddings, converts the preprocessed text into numerical features. The Machine

Learning Model class encapsulates the machine learning algorithm (e.g., SVM, Random Forest, or Neural Networks) that is trained on the features. Finally, the Evaluation class computes metrics such as accuracy, precision, and recall to assess the model's performance. Relationships between these classes are typically shown through associations and dependencies, illustrating how each component interacts in the fake news detection process.

3.4 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. . It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

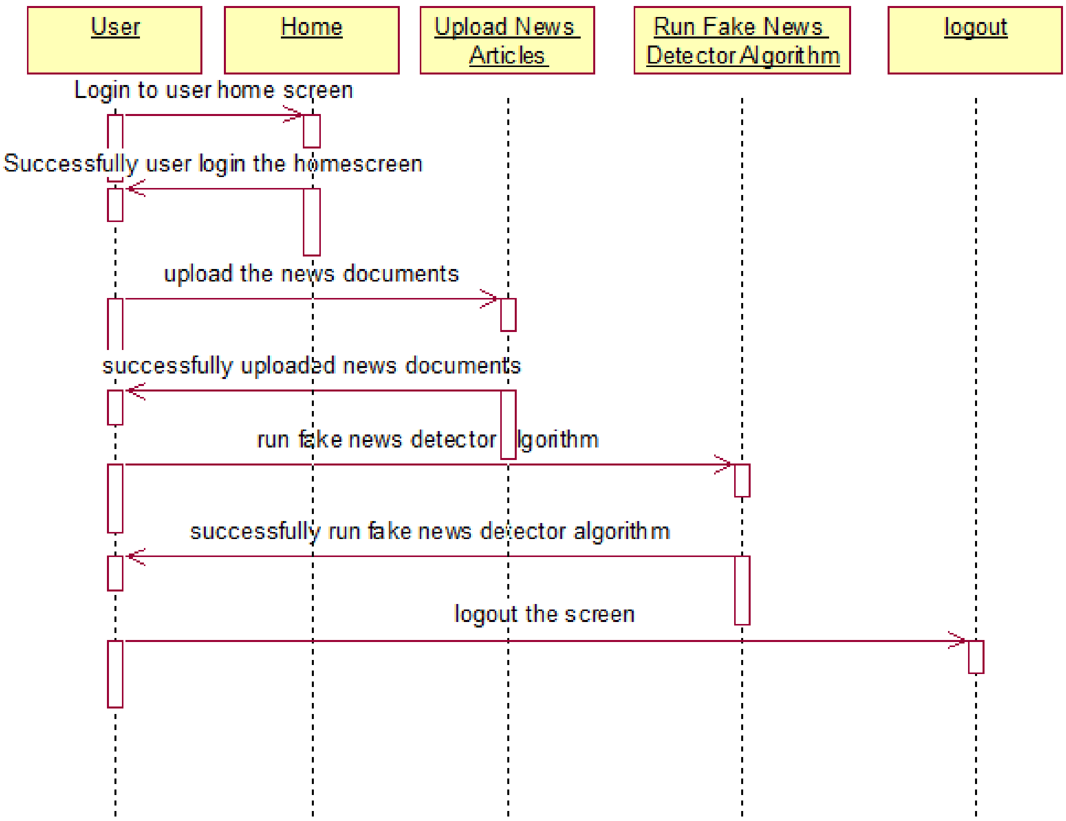


Fig 3.4:Diagram of sequential steps for detecting fake new

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

3.5 COMPONENT DIAGRAM

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the

structure of arbitrarily complex systems. Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer service provider relationship between the two components.

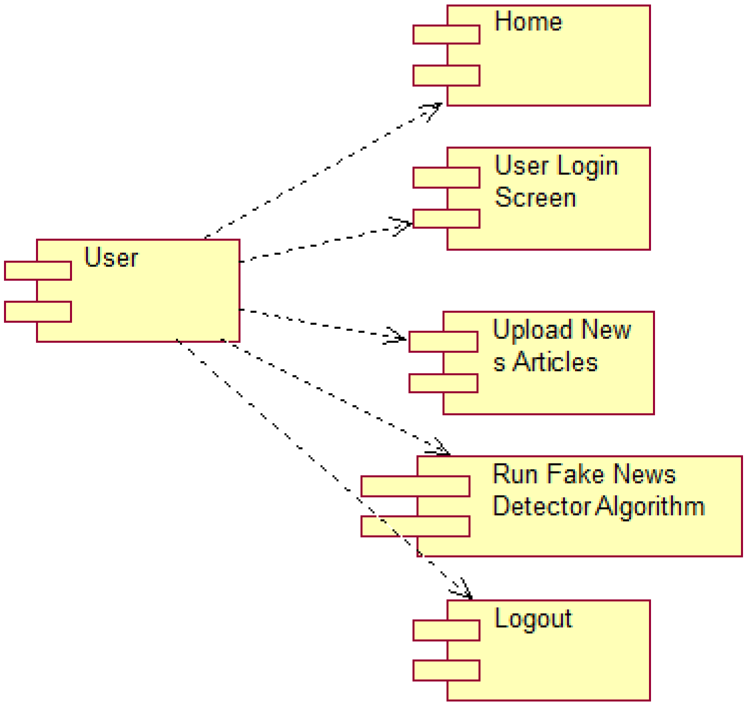


Fig 3.5 : Component Diagram for detecting Fake news

3.6 COLLABORATION DIAGRAM

A collaboration diagram describes interactions among objects in terms of sequenced messages . Collaboration diagrams represent a combination of information taken from class sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.

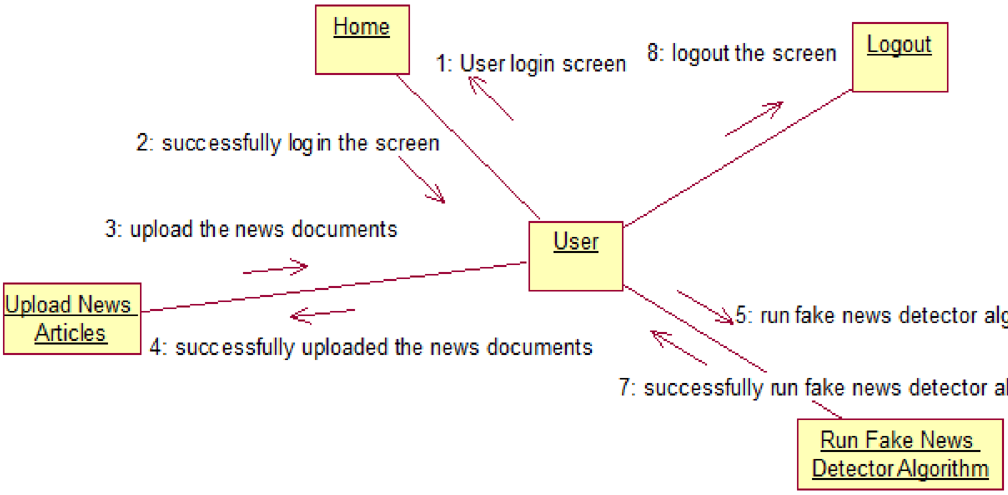


Fig 3.6: Visualization of fake news detection using a machine learning model.

3.7 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. An activity diagram for a fake news detection system can visually represent the flow of tasks in identifying and processing news articles to determine their authenticity.

Activity diagrams are an essential part of the UML that help visualize workflows, processes, or activities within a system. They depict how different actions are connected and how a system moves from one state to another. By offering a clear picture of both simple and complex workflows, activity diagrams make it easier for developers and stakeholders to understand how various elements interact in a system.

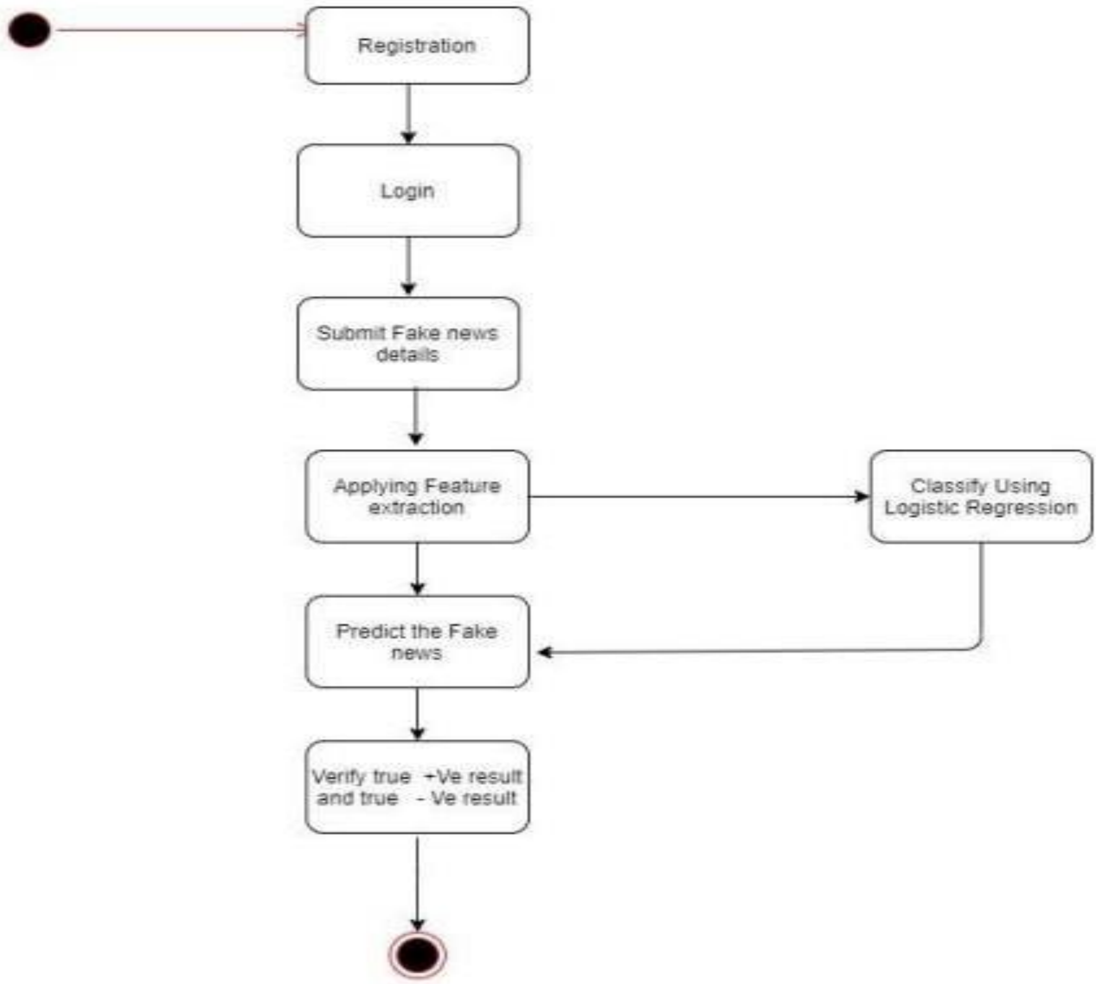


Fig 3.7 Activity Diagram for Fake news model

Chapter 4

IMPLEMENTATION

4.1 MACHINE LEARNING ALGORITHMS

4.1.1 Support Vector Machine

Support Vector Machine (SVM) plays a crucial role as a classification model. SVM is a supervised learning algorithm that aims to find the optimal hyperplane that separates data points from different classes—in this case, "fake" and "real" news articles. It works by transforming the input news articles, which are first processed and converted into numerical features (e.g., TF-IDF vectors or word embeddings), into a high-dimensional space. SVM then identifies the hyperplane that maximizes the margin between the two classes, effectively distinguishing between fake and real news. SVM is particularly effective in handling high-dimensional data and can be fine-tuned with kernel functions to manage non-linear relationships, making it a powerful tool in detecting subtle patterns within textual content. Its ability to generalize well, even with relatively small training datasets, makes it a popular choice in fake news detection systems.

4.1.2 LONG SHORT TERM MEMORY

The Long Short-Term Memory (LSTM) algorithm plays a significant role in detecting fake news by leveraging its ability to capture and understand sequential patterns in text data. LSTMs, a type of recurrent neural network (RNN), are particularly well-suited for processing natural language since they can retain information from earlier parts of a sequence to make better predictions. In fake news detection, LSTM models can analyze the sequence of words in news articles and learn the contextual relationships between them. This helps the model identify subtle patterns, linguistic cues, and inconsistencies that are often present in fake news. By maintaining long-term dependencies, LSTM can recognize misleading or deceptive language structures that may not be evident through simpler algorithms. After training on labeled datasets, the LSTM model can then classify new articles as real or fake based on its learned representations of truthful and misleading news patterns, contributing to more accurate fake news detection.

LSTM (Long Short-Term Memory) models enhance fake news detection by addressing the limitations of traditional machine learning algorithms in handling sequential data. Fake news often involves complex narratives, with subtle patterns of deception that unfold over multiple sentences or paragraphs. LSTMs excel in capturing these long-range dependencies and understanding the context across time steps due to their unique architecture, which includes gates for controlling the flow of information.

In fake news detection, LSTM models process a sequence of words or phrases in a news article while retaining important context from earlier parts of the text. This capability allows the model to recognize not only the specific meaning of individual words but also how those words interact within a larger narrative. LSTMs can detect misleading cues, like exaggerated claims or conflicting statements, and can identify stylistic inconsistencies (e.g., sensationalist language) that are commonly found in fake news.

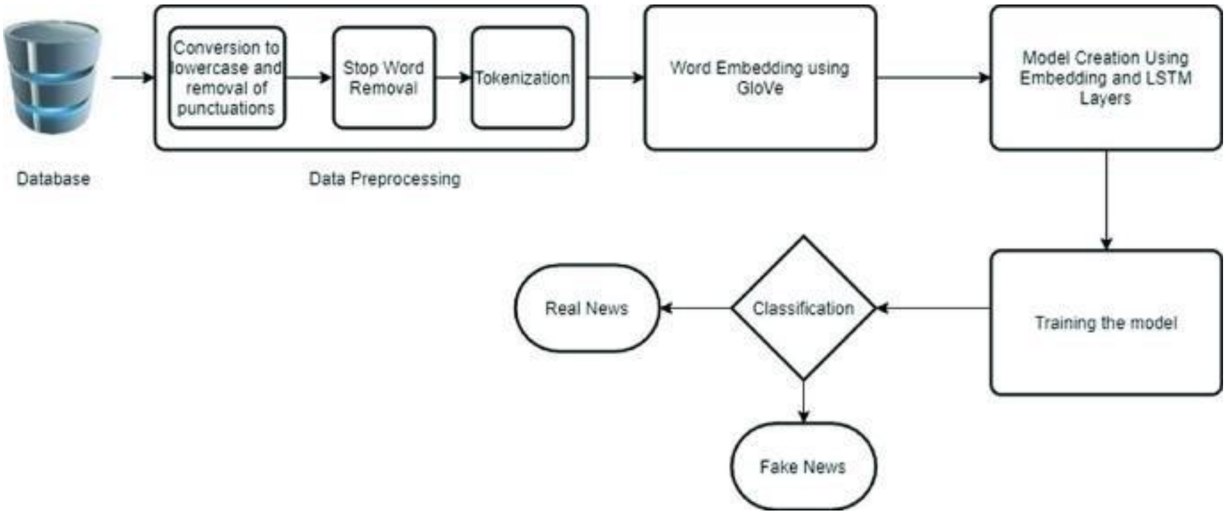


Fig 4.1.2 LSTM model

4.1.3 NAÏVE BIAS

Naive Bayes plays a fundamental role in detecting fake news by applying probabilistic reasoning to classify news articles based on the likelihood of certain words or features being present in fake or real news. As a supervised learning algorithm, Naive Bayes operates on the assumption of conditional independence between features, meaning that it assumes the presence of a particular word in a news article is independent of the presence of any other word, given the class (fake or real). Despite this simplifying assumption, Naive Bayes is highly effective in text classification tasks like fake news detection due to its efficiency and ability to handle high-dimensional data.

In fake news detection, Naive Bayes models are trained on a labeled dataset where news articles are categorized as real or fake. The algorithm calculates the probability of an article being fake or real based on the frequency of certain keywords, phrases, or linguistic patterns. For example, words like "breaking" or "shocking" may have a higher likelihood of appearing in fake news, while more neutral or factual language may indicate real news. Once trained, the model can classify new articles by determining which class (fake or real) the article's word distribution is more likely to belong to.

Naive Bayes is particularly useful in fake news detection because it is simple, fast, and works well even with relatively small datasets. Additionally, it requires less computational power compared to more complex algorithms like neural networks. Although Naive Bayes may not capture intricate relationships between words as effectively as more advanced models, it provides a strong baseline for fake news detection due to its interpretability and ease of implementation.

4.1.4 CONVOLUTIONAL NEURAL NETWORK (CNNs)

Convolutional Neural Networks (CNNs), commonly used in image processing, have also been effectively adapted for detecting fake news in text data by capturing local patterns within the text. In fake news detection, CNNs help extract important features from textual data by applying filters over the text, similar to how they detect edges or textures in images. These filters move across the sequence of words or word embeddings, identifying n-grams or combinations of words that may be indicative of deception, sensationalism, or bias in news articles.

The CNN architecture typically includes layers where convolution operations are performed to create feature maps, followed by pooling layers that reduce the dimensionality while preserving essential information. In the context of fake news detection, CNNs can capture specific phrases, patterns of words, or linguistic structures that are more likely to be associated with misleading content. This is particularly useful for detecting stylistic features such as sensationalist language or unusual patterns of word usage, which are often characteristic of fake news.

4.1.5 LOGISTIC REGRESSION

The logistic regression model applies a linear transformation to the input features, followed by a sigmoid function to map the result to a probability between 0 and 1. If the probability is above a certain threshold (typically 0.5), the news is classified as fake; otherwise, it is classified as real. One advantage of logistic regression is its simplicity and interpretability, making it easy to understand how specific features (e.g., certain words or phrases) influence the prediction of whether news is fake or real.

While logistic regression is not as powerful as more complex models like neural networks or LSTM, it performs well when the features are well-chosen, and the problem is linearly separable. It serves as a strong baseline model for fake news detection and can be computationally efficient, making it suitable for large datasets. Additionally, it provides insights into the significance of different features, which can help in understanding the characteristics of fake news.

4.1.6 PERFORMANCE METRICS

ACCURACY

The accuracy of detecting fake news using machine learning is measured using key metrics such as accuracy, precision, recall, F1 score, and support vector. Achieving high accuracy requires quality data, relevant feature selection, appropriate model choice, and hyperparameter tuning. This measures the overall correctness of the model by calculating the percentage of correctly classified instances (both real and fake news) out of the total number of instances. While accuracy is a useful metric, it can be misleading when dealing with imbalanced datasets where one class (e.g., real news) is much larger than the other (fake news).

Accuracy = (True Positives + True Negatives)/ Total Predictions

CLASSIFICATION REPORT

The classification report in the context of detecting fake news system provides a comprehensive summary of the model's performance by presenting various evaluation metrics for each class. It typically includes metrics such as precision, recall, F1 score, and support.

Precision:

The proportion of true positive predictions (correctly predicted approvals) out of all positive predictions (both true and false positives). Calculation: Precision= True Positives / {True Positives + False Positives}

text{Precision} =\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}Precision= True Positives + False Positives True Positives.

Recall:

Recall, also known as sensitivity, measures the proportion of correctly identified fake profiles among all actual fake profiles in the dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives (fake profiles misclassified as genuine). High recall indicates that the model effectively captures most of the fake profiles in the dataset. Recall= True Positives / (True Positives + False Negatives)

F1 Score:

The F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics. It takes into account both false positives and false negatives and is calculated as 2 * (precision * recall) / (precision + recall). F1 score ranges from 0 to 1, where higher values indicate better performance in terms of both precision and recall. F1 = 2 × (Precision × Recall / Precision + Recall)

Support:

Support represents the number of occurrences of each class in the dataset. It provides insights into the distribution of classes and helps interpret the significance of precision, recall, and F1 score. For instance, if one class has significantly higher support than the other, it may influence the interpretation of the model's performance metrics.

4.1.7 CONFUSION MATRIX

A confusion matrix is a performance evaluation tool commonly used in machine learning to assess the accuracy of classification models, including fake news detection. It provides a summary of prediction results on a classification problem, showing the number of true and false predictions across different classes (in this case, real news vs. fake news). The confusion matrix consists of four key elements:

1. True Positives (TP): The number of fake news articles that were correctly identified as fake.
2. True Negatives (TN): The number of real news articles that were correctly identified as real.
3. False Positives (FP): The number of real news articles that were incorrectly classified as fake (also known as a Type I error).
4. False Negatives (FN): The number of fake news articles that were incorrectly classified as real (also known as a Type II error).

By analyzing the values in the confusion matrix, one can gain insights into the strengths and weakness of the classification model. For instance, a high number of true positives and true negatives relative to false positives and false negatives indicates that the model performs well in accurately identifying both cyberbullying and not cyberbullying tweets. Conversely, a higher number of false positives or false negatives may indicate areas where the model's performance can be improved. The confusion matrix provides a clear and concise visualization of the classification results, enabling to understand the model's performance and make informed decisions about its effectiveness.

In the context of fake news detection, a confusion matrix allows us to analyze the model's performance, showing how often it confuses real news with fake news or vice versa. This helps identify if the model is biased towards over-predicting one class, and adjustments can be made to improve its balance between precision and recall, depending on the specific goal (e.g., minimizing false positives or false negatives).

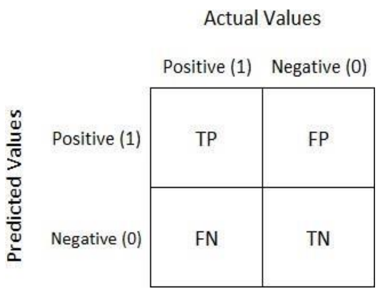


Fig 4.1.7 Confusion Matrix

4.2 DATASETS

The dataset for fake news detection is sourced from the Kaggle link it would typically contain essential features that facilitate training machine learning models to classify news articles as real or fake. The dataset would likely consist of the text, which represents the actual news content (either full articles, headlines, or social media posts), and the target label, which indicates whether the news is fake (e.g., labeled as "1") or real (e.g., labeled as "0"). The text feature is the core component that machine learning models analyze to detect patterns in language, style, or topics that can differentiate between authentic and deceptive information. Additional features, such as metadata like keywords, source, and location, might provide further context to help the model understand where the news originated from and its content focus.

<table>
<tr><td>ablaze</td><td></td><td>Revel in yours wmv videos by means of mac farewell ablaze wmv en route to dvd: GtxRWm</td><td>0</td></tr>
<tr><td>ablaze</td><td>Inang Pamantasan</td><td>Progressive greetings!</td><td>0</td></tr>
<tr><td>ablaze</td><td>Twitter Lockout in progress</td><td>Rene Ablaze & Jacinta - Secret 2k13 (Fallen Skies Edit) - Mar 30 2013 https://t.co/7MLMsUzV1Z</td><td>0</td></tr>
<tr><td>ablaze</td><td>Concord, CA</td><td>@Navista7 Steve these fires out here are something else! California is a tinderbox - and this clown was setting my 'hood al</td><td>1</td></tr>
<tr><td>ablaze</td><td>Calgary, AB</td><td>#NowPlaying: Rene Ablaze & Ian Buff - Magnitude http://t.co/Av2JSjfFtc #EDM</td><td>0</td></tr>
<tr><td>ablaze</td><td>Birmingham</td><td>@nxwestmidlands huge fire at Wholesale markets ablaze http://t.co/rwzbFVNXER</td><td>1</td></tr>
<tr><td>ablaze</td><td>San Francisco</td><td>@ablaze what time does your talk go until? I don't know if I can make it due to work.</td><td>0</td></tr>
<tr><td>accident</td><td>CLVLND</td><td>'I can't have kids cuz I got in a bicycle accident & split my testicles. it's impossible for me to have kids' MICHAEL YOU A</td><td>0</td></tr>
<tr><td>accident</td><td>Nashville, TN</td><td>Accident on I-24 W #NashvilleTraffic. Traffic moving 8m slower than usual. https://t.co/0GHk693EgJ</td><td>1</td></tr>
<tr><td>accident</td><td>Santa Clara, CA</td><td>Accident center lane blocked in #SantaClara on US-101 NB before Great America Pkwy #BayArea #Traffic http://t.co/pmlO</td><td>1</td></tr>
<tr><td>accident</td><td>UK</td><td>http://t.co/GKYe6gjTk5 Had a #personalinjury accident this summer? Read our advice & see how a #solicitor can help</td><td>0</td></tr>
<tr><td>accident</td><td>St. Louis, MO</td><td>#stlouis #caraccidentlawyer Speeding Among Top Causes of Teen Accidents https://t.co/k4zoMOF319 https://t.co/S2kXVM</td><td>0</td></tr>
<tr><td>airplane%20acc</td><td>Salt Lake City, Utah</td><td>@crobscarla your lifetime odds of dying from an airplane accident are 1 in 8015.</td><td>0</td></tr>
<tr><td>airplane%20acc</td><td>Palo Alto, CA</td><td>Experts in France begin examining airplane debris found on Reunion Island: French air accident experts on Wedn... http:/</td><td>1</td></tr>
<tr><td>airplane%20accident</td><td></td><td>@AlexAllTimeLow awwww they're on an airplane accident and they're gonna die what a cuties ???? good job!</td><td>1</td></tr>
<tr><td>airplane%20acc</td><td>Spain but Opa-Locka, FL</td><td>family members of osama bin laden have died in an airplane accident how ironic ?????? mhmmm gov shit i suspect</td><td>1</td></tr>
<tr><td>airplane%20acc</td><td>Jaipur, India</td><td>Man Goes into Airplane Engine Accident: http://t.co/TYJxrFd3St via @YouTube</td><td>1</td></tr>
<tr><td>airplane%20acc</td><td>Hyderabad Telangana INDIA</td><td>Horrible Accident Man Died In Wings of Airplane (29-07-2015) http://t.co/i7kZtevb2v</td><td>1</td></tr>
<tr><td>airplane%20acc</td><td>Eagle Pass, Texas</td><td>A Cessna airplane accident in Ocampo Coahuila Mexico on July 29 2015 killed four men including a State of Coahuila gover</td><td>1</td></tr>
<tr><td>airplane%20acc</td><td>bangalore</td><td>#Horrible #Accident Man Died In Wings Airplane (29-07-2015) #WatchTheVideo http://t.co/p64xRVgJIk</td><td>1</td></tr>
<tr><td>airplane%20acc</td><td>Financial News and Views</td><td>Experts in France begin examining airplane debris found on Reunion Island http://t.co/LsMx2vwr3J French air accident ex</td><td>1</td></tr>
<tr><td>armageddon</td><td>probably the strip club</td><td>//im gonna beat armageddon as Hsu Hao ????</td><td>0</td></tr>
<tr><td>armageddon</td><td>Canada</td><td>@ENews Ben Affleck......I know there's a wife/kids and other girls but I can't help it. I've loved him since Armageddon #eo</td><td>0</td></tr>
<tr><td>armageddon</td><td>England</td><td>'If I'd have had a long coat to hand I'd have worn it. The certainty of armageddon bears a sense of occasion.'</td><td>0</td></tr>
<tr><td>armageddon</td><td>USA</td><td>YOUR PHONE IS SPYING ON YOU! Hidden Back Door NSA Data Mining Software | THE FINANCIAL ARMAGEDDON BLOG htt</td><td>0</td></tr>
<tr><td>armageddon</td><td>California, United States</td><td>#PBBan (Temporary:300) hyider_ghost2 @'aRmageddon | DO NOT KILL | FLAGS ONLY | Fast XP' for Reason</td><td>1</td></tr>
<tr><td>armageddon</td><td></td><td>RT @RTRRTcoach: #Love #TrueLove #romance lith #Voodoo #seduction #Astrology #RTRRT #LOTZ 9-11 #apocalypse #Arm</td><td>0</td></tr>
<tr><td>armageddon</td><td>California, United States</td><td>#PBBan (Temporary:300) fighterdena @'aRmageddon | DO NOT KILL | FLAGS ONLY | Fast XP' for Reason</td><td>0</td></tr>
<tr><td>armageddon</td><td>New York City</td><td>Photo: Sketch I did based on the A Taste of Armageddon episode of #startrek #tos http://t.co/a2e6dcsk88</td><td>0</td></tr>
<tr><td>armageddon</td><td>Here And There</td><td>Armageddon https://t.co/uCSUDk3q1d</td><td>1</td></tr>
<tr><td>armageddon</td><td>Rotterdam, Zuid-Holland</td><td>@AberdeenFC @AberdeenFanPage</td><td>0</td></tr>
<tr><td>arsonist</td><td>toronto</td><td>Bloor/Ossington arsonist also burned a mattress on Northumberland St #cbcto http://t.co/wpDvT31sne</td><td>0</td></tr>
<tr><td>arsonist</td><td>[Blonde Bi Fry.]</td><td>'wHeRE's mY aRsOnISt aT???'</td><td>0</td></tr>
<tr><td>arsonist</td><td>America</td><td>If you don't have anything nice to say you can come sit with me.</td><td>0</td></tr>
<tr><td>arsonist</td><td>NYC :) Ex- #Islamophobe</td><td>#Vegetarian #Vegan Video shows arsonist torching popular BK restaurant Strictly Vegetarian... http://t.co/kxpLYoM9RR #G</td><td>0</td></tr>
<tr><td>arsonist</td><td>SF Bay Area</td><td>#Arsonist arrested for setting many fires. WATCH tonightÂ‰Ã›Âªs other #headlines: http://t.co/sqgogJ3S5r. #Nightbeat @</td><td>1</td></tr>
<tr><td>arsonist</td><td>Orange County, California</td><td>Video Captures Man Removing American Flag From Long Beach CA Home Burning It; Arsonist Sought http://t.co/JP2Qlrunj</td><td>0</td></tr>
<tr><td>arsonist</td><td>ss</td><td>@58hif my trick is to think about nasty things</td><td>0</td></tr>
<tr><td>attack</td><td></td><td>Heart disease prevention: What about secondhand smoke? http://t.co/YdgMGBrYL2</td><td>0</td></tr>
<tr><td>attack</td><td>Dayton, Ohio</td><td>A Dayton-area org tells me it was hit by a cyber attack: http://t.co/7LhKJz0IVO</td><td>1</td></tr>
<tr><td>attack</td><td></td><td>Attack on Titan game on PS Vita yay! Can't wait for 2016</td><td>0</td></tr>
<tr><td>attack</td><td>Global</td><td>[infowars] Nashville Theater Attack: Will Gun Grabbers Now Demand Â‰Ã›ÃHatchet Control?Â‰Ã›Â http://t.co/n3yJb8</td><td>1</td></tr>
<tr><td>attack</td><td>ph</td><td>anxiety attack ??</td><td>0</td></tr>
<tr><td>attacked</td><td>Port Jervis, NY</td><td>My dog attacked me for my food #pugprobs</td><td>0</td></tr>
<tr><td>attacked</td><td>City Of Joy</td><td>Cop injured in gunfight as militants attack Udhampur police post: Suspected militants attacked a police post i... http://t.co/</td><td>1</td></tr>
<tr><td>attacked</td><td>Los Angeles, CA</td><td>@envw98 @NickCoCoFree @JulieDiCaro @jdabe80 I asked how did he feel attacked by julie. I asked if he was frail. That</td><td>0</td></tr>
<tr><td>attacked</td><td>Texas, USA</td><td>@messeymetoo I feel attacked</td><td>0</td></tr>
</table>

4.3 SAMPLE CODE

FakeNews.py

```
from tkinter import messagebox
from tkinter import *

from tkinter import simpledialog

import tkinter

import matplotlib.pyplot as plt

import numpy as np

from tkinter import ttk

from tkinter import filedialog

import pandas as pd

from sklearn.model_selection import train_test_split

from string import punctuation

from nltk.corpus import stopwords

import nltk

from nltk.stem import WordNetLemmatizer

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder

from keras.models import Sequential

from keras.layers.core import Dense,Activation,Dropout

from sklearn.preprocessing import OneHotEncoder

import keras.layers
```

```
from keras.models import model_from_json

import pickle

import os

from sklearn.preprocessing import normalize

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten, LSTM

main = Tk()

main.title("DETECTION OF FAKE NEWS THROUGH IMPLEMENTATION

OF DATA SCIENCE APPLICATION")

main.geometry("1300x1200")

global filename

global X, Y

global tfidf_X_train, tfidf_X_test, tfidf_y_train, tfidf_y_test

global tfidf_vectorizer

global accuracy,error

stop_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

textdata = []

labels = []

global classifier

def cleanPost(doc):
```

```
    tokens = doc.split()
    table = str.maketrans('', '', punctuation)
    tokens = [w.translate(table) for w in tokens]
    tokens = [word for word in tokens if word.isalpha()]
    tokens = [w for w in tokens if not w in stop_words]
    tokens = [word for word in tokens if len(word) > 1]
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    tokens = ' '.join(tokens)
    return tokens
def uploadDataset():
    global filename
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="TwitterNewsData")
    textdata.clear()
    labels.clear()
    dataset = pd.read_csv(filename)
    dataset = dataset.fillna(' ')
    for i in range(len(dataset)):
        msg = dataset.get_value(i, 'text')
        label = dataset.get_value(i, 'target')
        msg = str(msg)
```

```
        msg = msg.strip().lower()

        labels.append(int(label))

        clean = cleanPost(msg)

        textdata.append(clean)

        text.insert(END,clean+" ==== "+str(label)+"\n")

    def preprocess():

    text.delete('1.0', END)

    global X, Y

    global tfidf_vectorizer

    global tfidf_X_train, tfidf_X_test, tfidf_y_train, tfidf_y_test

    stopwords=stopwords = nltk.corpus.stopwords.words("english")

    tfidf_vectorizer = TfidfVectorizer(stop_words=stopwords, use_idf=True,
ngram_range=(1,2),smooth_idf=False, norm=None, decode_error='replace',
max_features=200)

    tfidf = tfidf_vectorizer.fit_transform(textdata).toarray()

    df = pd.DataFrame(tfidf, columns=tfidf_vectorizer.get_feature_names())

    text.insert(END,str(df))

    print(df.shape)

    df = df.values

    X = df[:, 0:df.shape[1]]

    X = normalize(X)

    Y = np.asarray(labels)
```

```
    le = LabelEncoder()
    Y = le.fit_transform(Y)
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    Y = Y.reshape(-1, 1)
    print(X.shape)
    encoder = OneHotEncoder(sparse=False)
    #Y = encoder.fit_transform(Y)
    X = X.reshape((X.shape[0], X.shape[1], 1))
    print(Y)
    print(Y.shape)
    print(X.shape)
    tfidf_X_train, tfidf_X_test, tfidf_y_train, tfidf_y_test = train_test_split(X, Y, test_size=0.2)
    text.insert(END,"\n\nTotal News found in dataset : "+str(len(X))+"\n")
    text.insert(END,"Total records used to train machine learning algorithms : "+str(len(tfidf_X_train))+"\n")
    text.insert(END,"Total records used to test machine learning algorithms : "+str(len(tfidf_X_test))+"\n")
def runLSTM():
```

```
    text.delete('1.0', END)

    global classifier

    if os.path.exists('model/model.json'):

        with open('model/model.json', "r") as json_file:

            loaded_model_json = json_file.read()

            classifier = model_from_json(loaded_model_json)

        classifier.load_weights("model/model_weights.h5")

        classifier._make_predict_function()

        print(classifier.summary())

        f = open('model/history.pckl', 'rb')

        data = pickle.load(f)

        f.close()

        acc = data['accuracy']

        acc = acc[9] * 100

        text.insert(END,"LSTM Fake News Detection Accuracy : "+str(acc)+"\n\n")

        text.insert(END,'LSTM Model Summary can be seen in black console for layer details\n')

        with open('model/model.txt', 'rb') as file:

            classifier = pickle.load(file)

        file.close()

    else:
```

```
lstm_model = Sequential()
lstm_model.add(LSTM(128,input_shape=(X.shape[1:]),activation='relu',
return_sequences=True))
lstm_model.add(Dropout(0.2))
lstm_model.add(LSTM(128, activation='relu'))
lstm_model.add(Dropout(0.2))
lstm_model.add(Dense(32, activation='relu'))
lstm_model.add(Dropout(0.2))
lstm_model.add(Dense(2, activation='softmax'))
lstm_model.compile(loss='sparse_categorical_crossentropy',
optimizer='adam', metrics=['accuracy'])
hist = lstm_model.fit(X,  Y, epochs=10, validation_data=(tfidf_X_test,
tfidf_y_test))
classifier = lstm_model
classifier.save_weights('model/model_weights.h5')
model_json = classifier.to_json()
with open("model/model.json", "w") as json_file:
    json_file.write(model_json)
accuracy = hist.history
f = open('model/history.pckl', 'wb')
pickle.dump(accuracy, f)
```

```
        f.close()

        acc = accuracy['accuracy']

        acc = acc[9] * 100

        text.insert(END,"LSTM Accuracy : "+str(acc)+"\n\n")

        text.insert(END,'LSTM Model Summary can be seen in black console for
layer details\n')

        print(lstm_model.summary())

def graph():

    f = open('model/history.pckl', 'rb')

    data = pickle.load(f)

    f.close()

    acc = data['accuracy']

    loss = data['loss']

    plt.figure(figsize=(10,6))

    plt.grid(True)

    plt.xlabel('Epcchs')

    plt.ylabel('Accuracy/Loss')

    plt.plot(acc, 'ro-', color = 'green')

    plt.plot(loss, 'ro-', color = 'blue')

    plt.legend(['Accuracy','Loss'], loc='upper left')

    #plt.xticks(wordloss.index)

    plt.title('LSTM Model Accuracy & Loss Graph')
```

```
    plt.show()
def predict():
    testfile = filedialog.askopenfilename(initialdir="TwitterNewsData")
    testData = pd.read_csv(testfile)
    text.delete('1.0', END)
    testData = testData.values
    testData = testData[:,0]
    print(testData)
    for i in range(len(testData)):
        msg = testData[i]
        msg1 = testData[i]
        print(msg)
        review = msg.lower()
        review = review.strip().lower()
        review = cleanPost(review)
        testReview = tfidf_vectorizer.transform([review]).toarray()
        predict = classifier.predict(testReview)
        print(predict)
        if predict == 0:
            text.insert(END,msg1+" === Given news predicted as GENUINE\n\n")
        else:
```

```
        text.insert(END,msg1+" == Given news predicted as FAKE\n\n")

font = ('times', 15, 'bold')

title = Label(main, text='DETECTION OF FAKE NEWS THROUGH IMPLEMENTATION OF DATA SCIENCE APPLICATION')

title.config(bg='gold2', fg='thistle1')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)



font1 = ('times', 13, 'bold')

ff = ('times', 12, 'bold')

uploadButton = Button(main, text="Upload Fake News Dataset", command=uploadDataset)

uploadButton.place(x=20,y=100)

uploadButton.config(font=ff)

processButton = Button(main, text="Preprocess Dataset", command=preprocess)

processButton.place(x=20,y=150)

processButton.config(font=ff)

dtButton = Button(main, text="Run LSTM Algorithm", command=runLSTM)

dtButton.place(x=20,y=200)

dtButton.config(font=ff)
```

```
graphButton = Button(main, text="Accuracy & Loss Graph", command=graph)

graphButton.place(x=20,y=250)

graphButton.config(font=ff)

predictButton = Button(main, text="Test News Detection", command=predict)

predictButton.place(x=20,y=300)

predictButton.config(font=ff)

font1 = ('times', 12, 'bold')

text=Text(main,height=30,width=100)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=330,y=100)

text.config(font=font1)

main.config(bg='DarkSlateGray1')

main.mainloop()
```

4.4 RESULT ANALYSIS

The result analysis of a detecting twitter fake news using machine learning evaluates key performance metrics such as accuracy, precision, recall, F1 score, and support vector to understand the model's effectiveness. With an accuracy of 69.4999%, the model demonstrates good overall performance, but there's room for improvement.

Further considerations include addressing class imbalances, analyzing feature importance, comparing different models, and using cross-validation to ensure robust performance metrics. Overall, this analysis highlights the model's strengths and areas for refinement to ensure reliable predictions.

This interface allows users to load a dataset, preprocess it, run the LSTM algorithm, view performance metrics (accuracy and loss graphs), and finally test news detection using the trained model. The predictions are displayed directly in the interface, showing which news items are considered fake or real based on the algorithm's learned patterns.

The application demonstrates the implementation of a fake news detection system using data science and machine learning techniques, providing a simple yet functional interface for users to interact with the model and view its predictions.

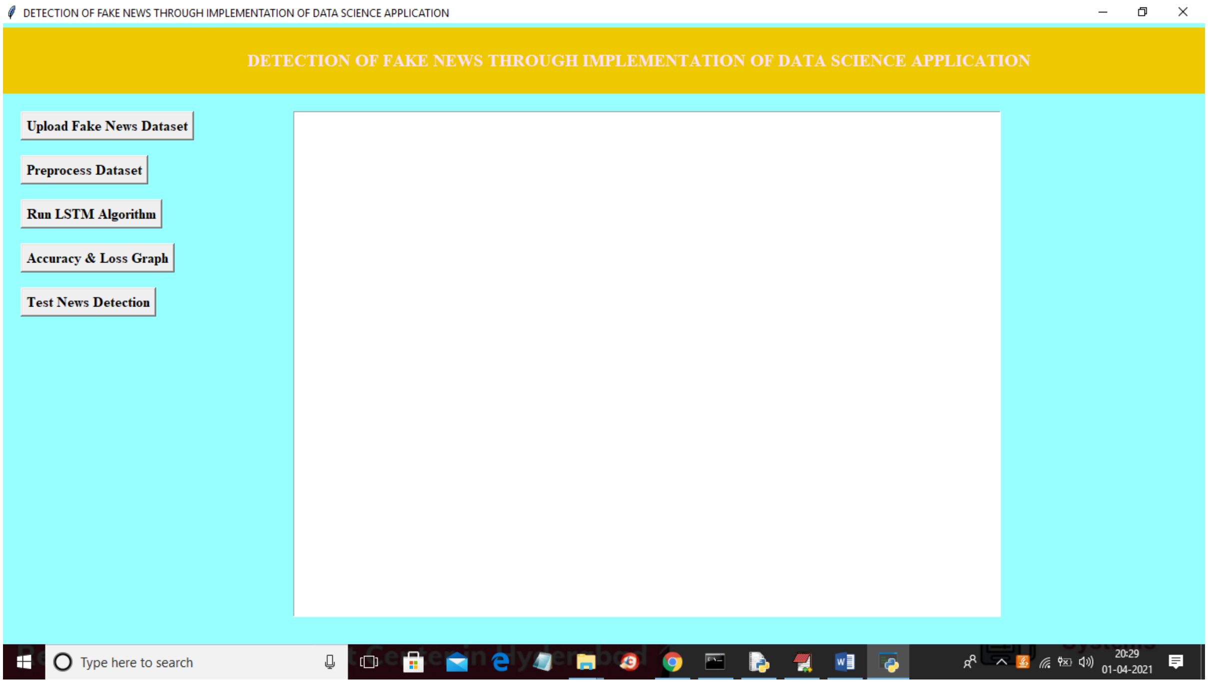
The large text area in the center of the interface displays a list of news sentences or phrases, each followed by a label (either "1" or "0"). These labels indicate whether the system has classified the news as **fake** (denoted by "1") or **real** (denoted by "0

Chapter 5

SCREENSHOTS

5.1 Home page

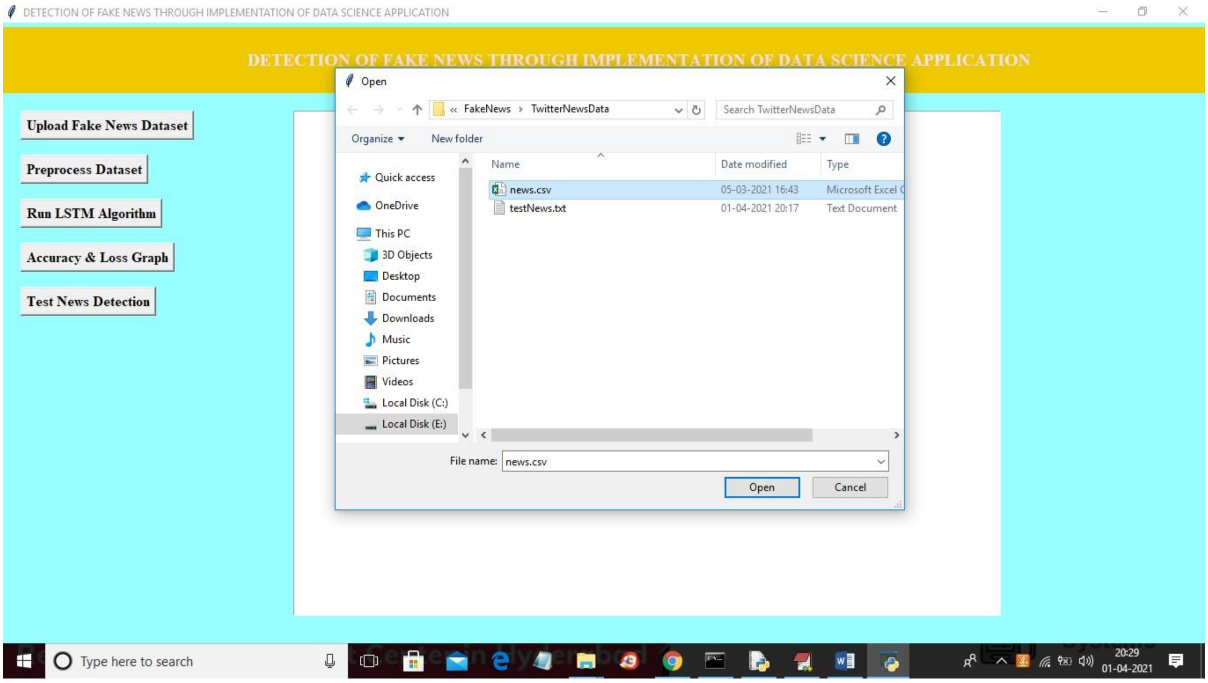
The project's homepage interface serves as the gateway for users, offering a seamless login experience. Users input their credentials in designated fields, ensuring secure access to the platform. With a focus on user-friendly design and robust security measures, the interface sets the stage for positive user interaction.



Screenshot 5.1 home page

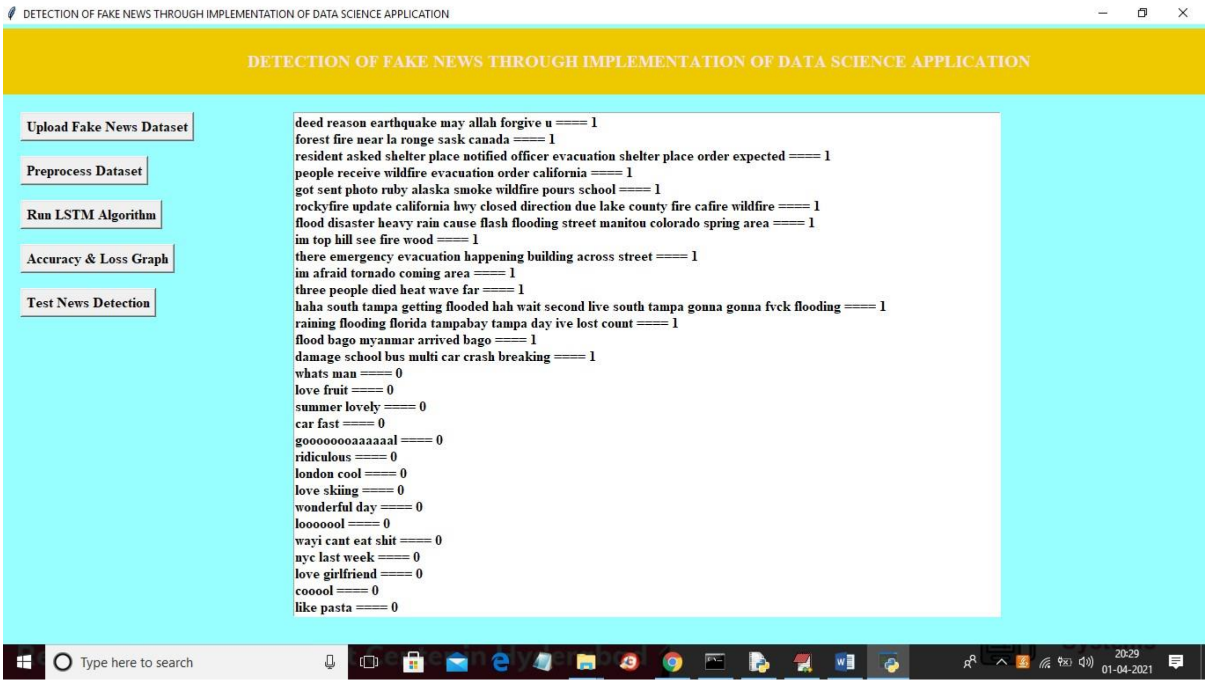
In above fig 5.1click on 'Upload Fake News Dataset' button to upload dataset

Upload Fake News Dataset: This button likely allows the user to upload a dataset containing news articles, labeled as either real or fake. This dataset is essential for training or testing the machine learning model.



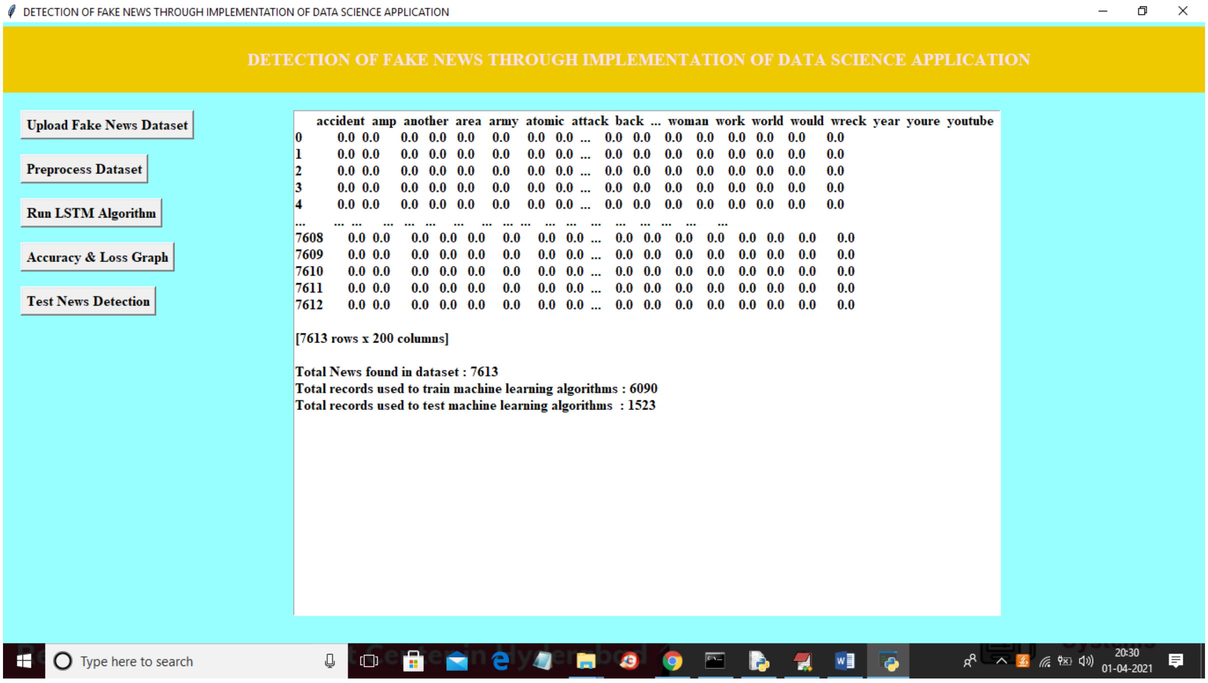
Screenshot 5.2 Uploading Train Dataset

In above fig 5.2 screen selecting and uploading 'news.csv' file and then click on 'Open' button to load dataset and to get below screen.



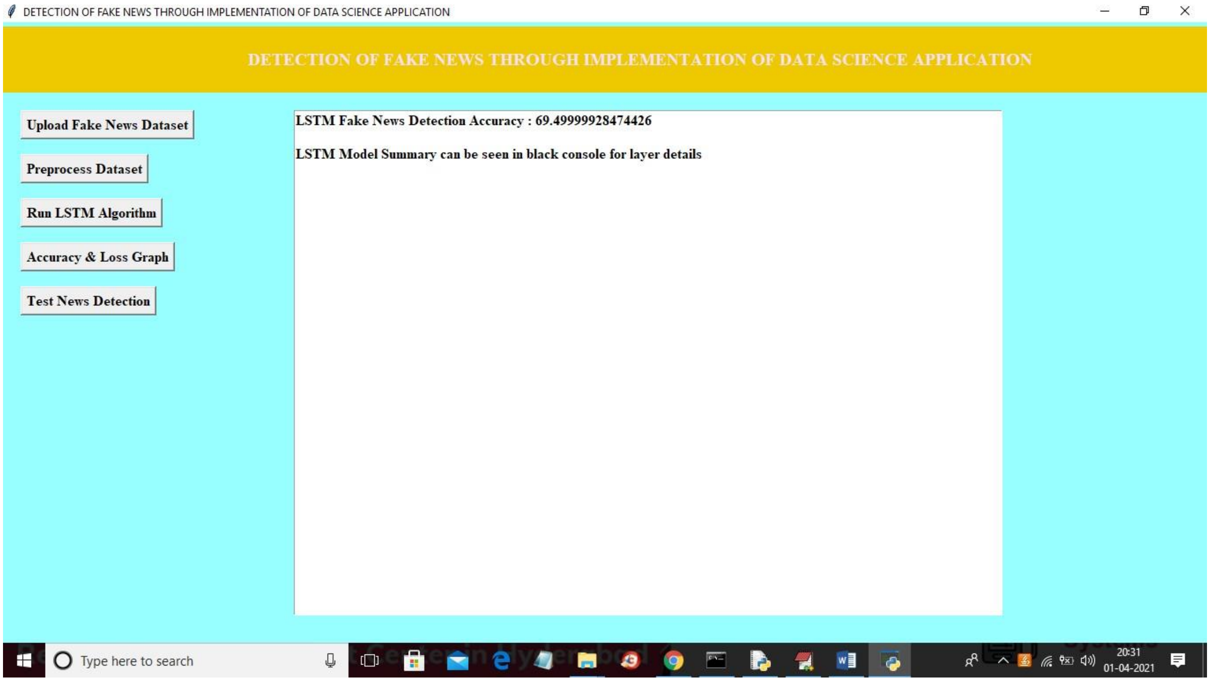
Screenshot 5.3 Reading data as fake and real

In above fig 5.3 dataset loaded and then in text area we can see all news text with the class label as 0 or 1 and now click on ‘Preprocess Dataset & Apply NGram’ button to convert above string data to numeric vector and to get below screen.



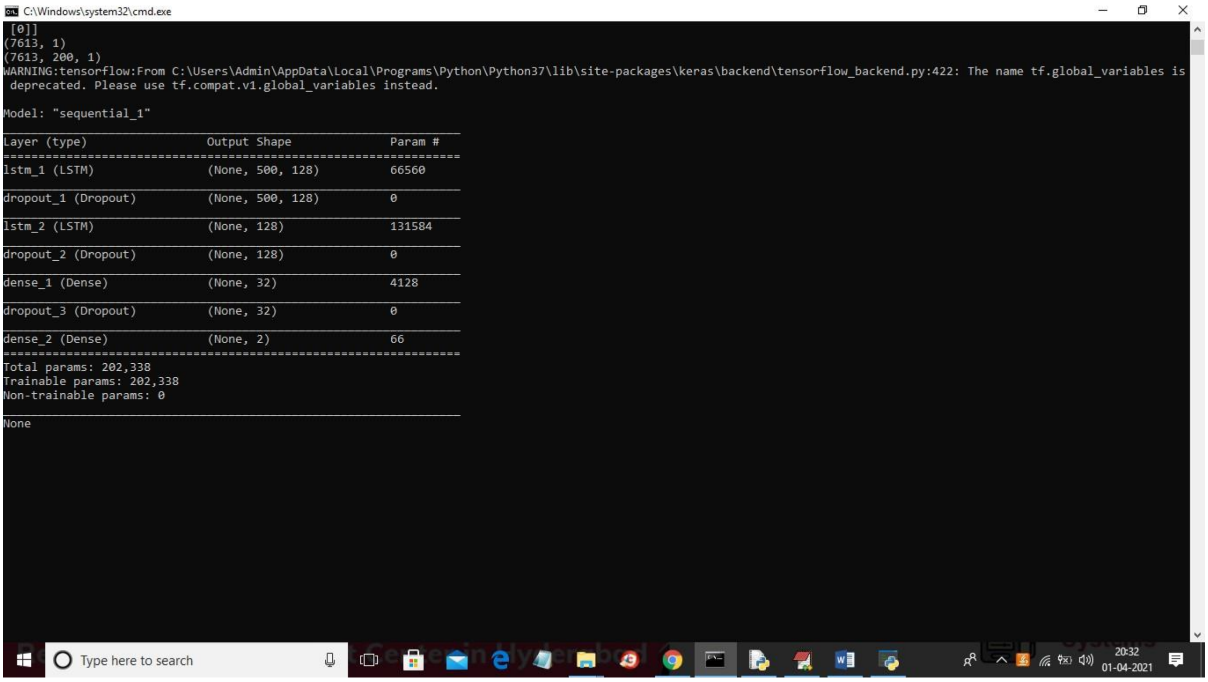
Screenshot 5.4 : Loaded data

In above fig 5.4 all news words put in column header and if that word appear in any row then that rows column will be change with word count and if not appear then 0 will be put in column. In above screen showing some records from total 7612 news records and in bottom lines we can see dataset contains total 7613 records and then application using 80% (6090 news records) for training and then using 20%(1523 news records) for testing and now dataset is ready with numeric record and now click on ‘Run LSTM Algorithm’ button to train above dataset with LSTM and then build LSTM model and then calculate accuracy and error rate.



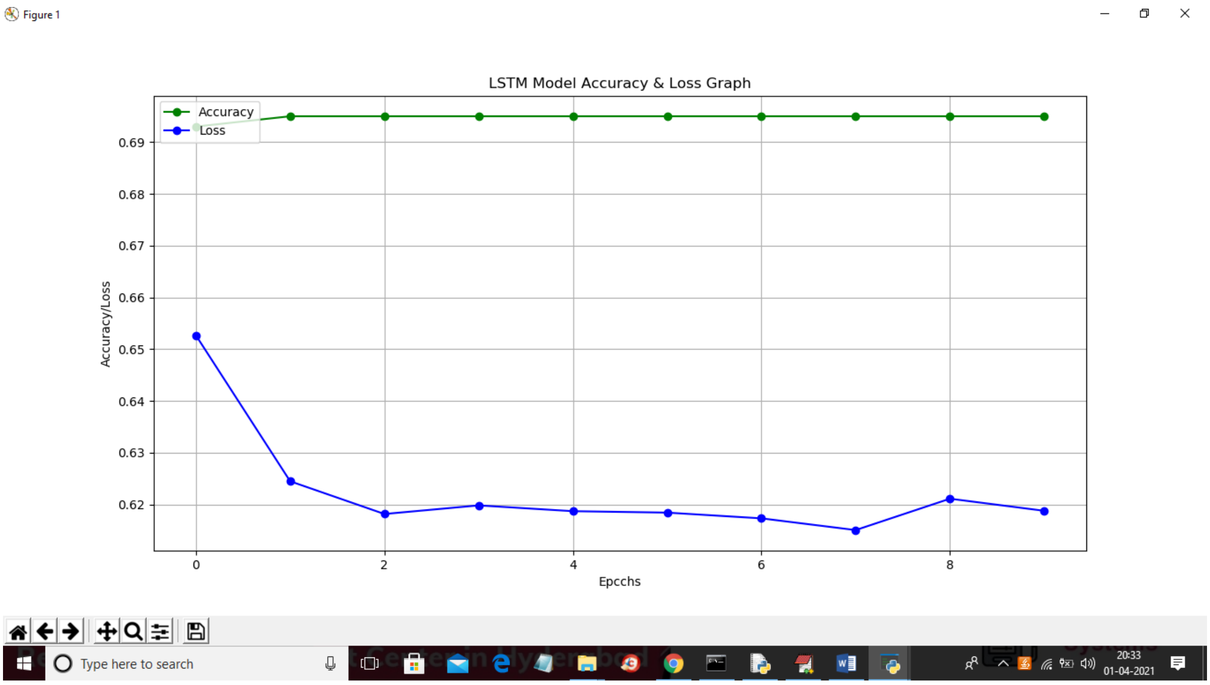
Screenshot 5.5: LSTM Prediction

In above fig 5.5 LSTM model is generated and we got its prediction accuracy as 69.49% and we can see below console to see LSTM layer details.



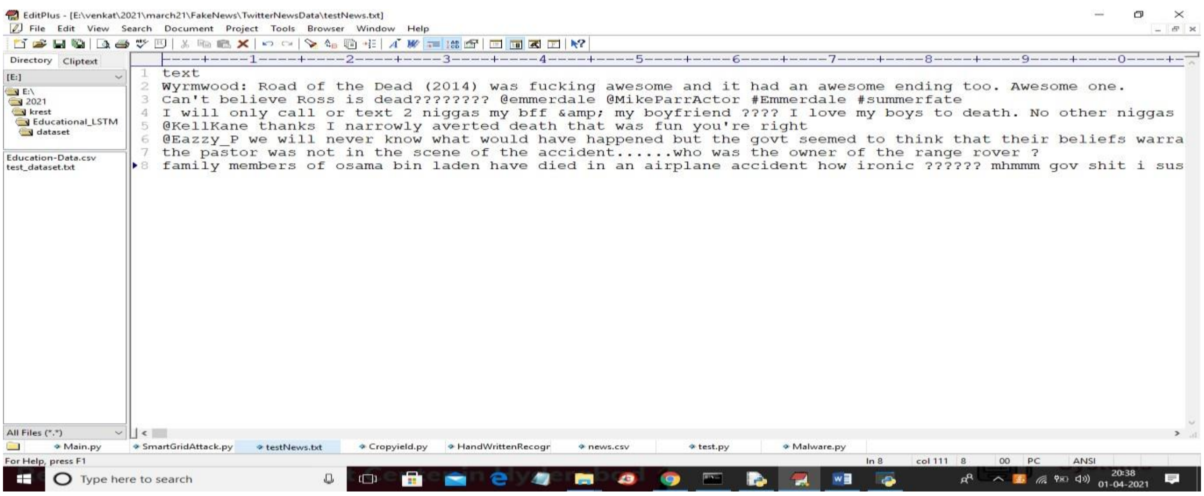
Screenshot 5.6: processing details

In above fig 5.6 different LSTM layers are created to filter input data to get efficient features for prediction. Now click on ‘Accuracy & Loss Graph’ button to get LSTM graph.



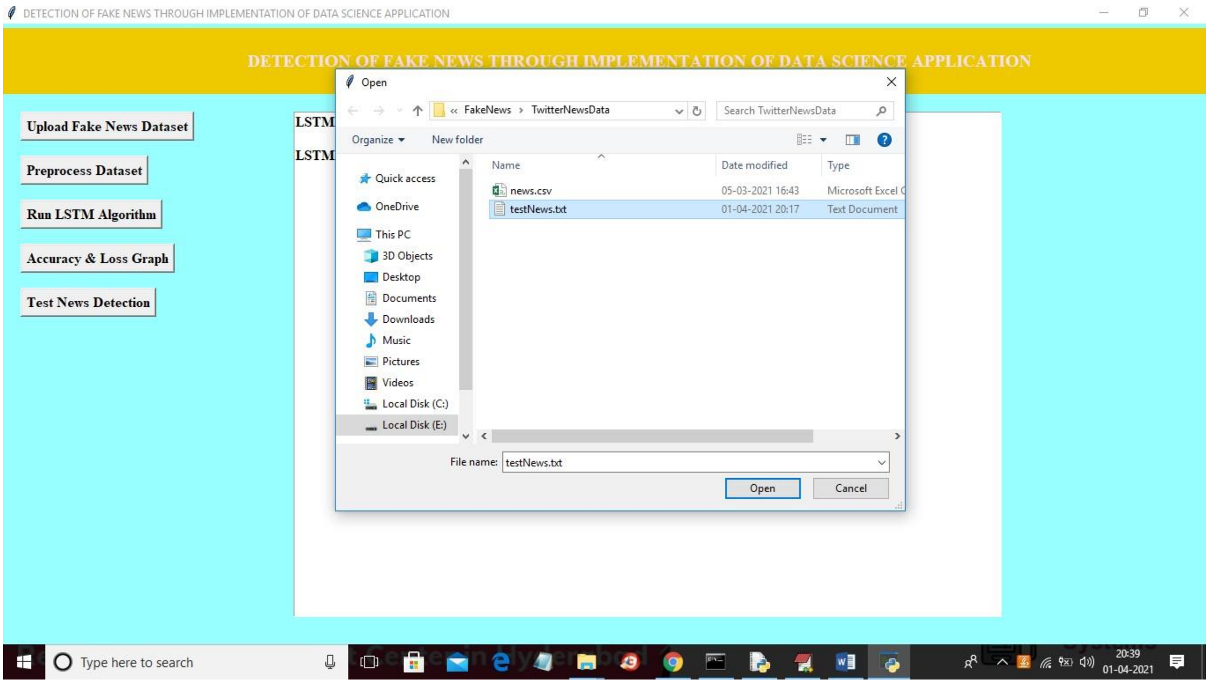
Screenshot 5.7:Graph

In above graph fig 5.7 x-axis represents epoch/iterations and y-axis represents accuracy and loss value and green line represents accuracy and blue line represents loss value and at each increasing epoch loss values get decrease and accuracy reached to 70%. Now click on ‘Test News Detection’ button to upload some test news sentences and then application predict whether that news is genuine or fake. In below test news dataset we can see only TEXT data no class label and LSTM will predict class label for that test news.



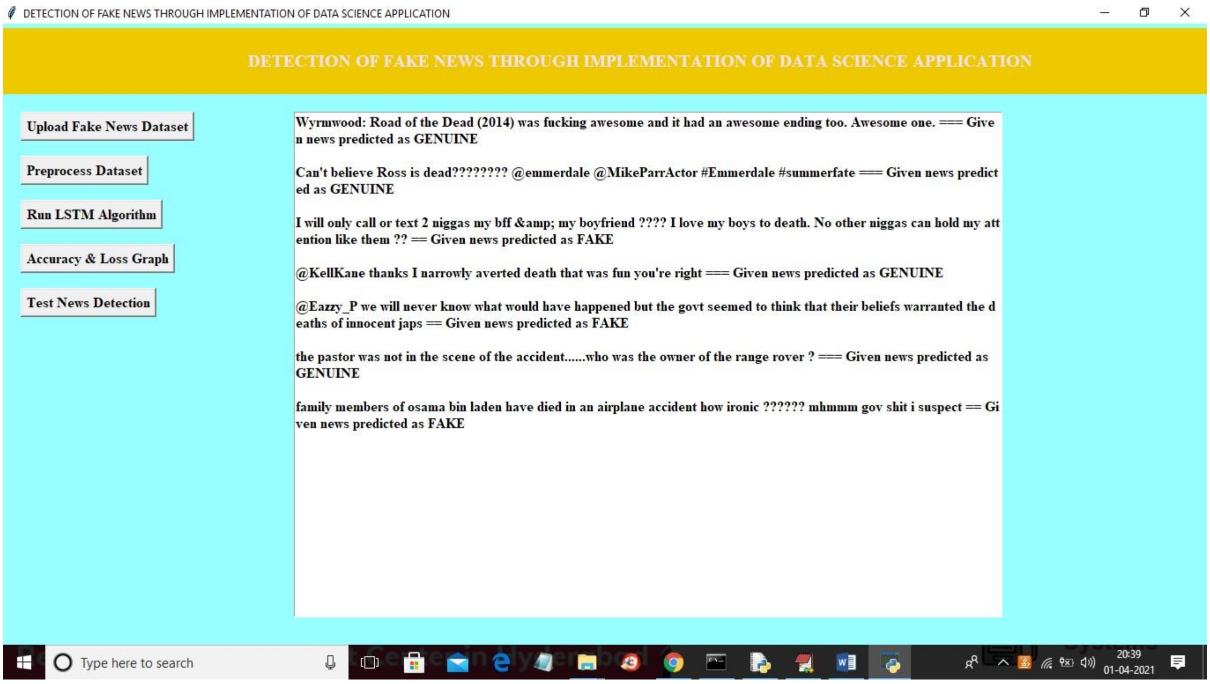
Screenshot 5.8: Text data

In above fig5.8 in test news we have only one column which contains only news 'TEXT' and after applying above test news we will get prediction result.



Screenshot 5.9 Test data

In above fig 5.9 screen selecting and uploading 'testNews.txt' file and then click on 'Open' button to load data and to get below prediction result.



Screenshot 5.10: Result

In above fig 5.10 before dashed symbols we have news text and after dashed symbol application predict news as ‘FAKE or GENUINE’. After building model when we gave any news text then LSTM will check whether more words belongs to genuine or fake category and whatever category get more matching percentage then applicable.

Chapter 6

TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid input : identified classes of valid input must be accepted.

Invalid input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.3 TEST CASES

Test Case ID	Test Case Name	Input	Expected output	Actual Output	Test Case Pass/Fail
1	Upload f	Username: abcde Password : abcde@345	It should move to user home page	It moves to the user home page	Pass
2	Check Username	Username: XYZ (Which is invalid)	It shows the error The username is not available	It shows the error The username is not available	Pass
3	Creating an account	Username: hello (if username is already taken)	Gives the error Username already exists	Gives the error that username already exists	Pass
4	Upload news articles	Without loading dataset	We cannot do further operations.	We can do further operations.	pass
5	Run fake news algorithm	Without loading the dataset	We cannot run the algorithm	The fake news detector algorithm run successfully.	Pass

Table 6.3 :Test Cases

Chapter 7
CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION

This study reviewed existing literature to detect aggressive behavior on social media. By using various machine learning approaches. In this paper, we analyzed a computerized model for checking the verification of news extracted from Twitter which gives general answers for information accumulation and expository demonstration towards fake news recognition. After having an idea from the supervised models, a deep learning-based model is proposed to identify fake news. The accuracy metric presumably would be altogether improved by methods for utilizing progressively complex model. It is worth noting, that even with the given dataset, only part of the information was used. The current project did not include domain knowledge related features, such as entity-relationships.

Future studies could extract name entities from each pair of news headline and news body and analyze their relationships through a knowledge base. The study demonstrated that even the very basic algorithms on fields like AI and Machine Learning may find a decent outcome on such a critical issue as the spread of fake news issues worldwide. Accordingly, the aftereffects of this examination propose much more, that systems like this might come very much handy and be effectively used to handle this critical issue. This work exhibits a programmed model for identifying fake news in well-known Twitter strings. Such a model could be important to a huge number of social media users by expanding their own credibility decisions.

The dataset in this examination is relied upon to be utilized for arrangements which utilized machine learning based statistical calculations, for example, Support Vector Machines (SVM), Naive Bayes (NB), Recurrent Neural Network (RNN), Logistic Regression (LR), Long Short Term Memory (LSTM). In this investigation, SVM performs best for characterization technique. We exploit the latent community structure in the global news network to improve the prediction of the viral cascades of news about events. The cascades which have early adopters in different communities have advantages in disseminating the contagion to these communities in parallel and therefore are more likely to result in the viral infections within a limited time period. Our model captures such property by inferring the community structure using the response times of nodes

7.2 FUTURE SCOPE

The future scope of detecting fake news system using machine learning encompasses several promising areas for development and enhancement. Improving model accuracy through advanced algorithms and hyperparameter tuning is a key focus, alongside integrating additional features. Real-time prediction capabilities and continuous learning methods will enable the system to adapt dynamically as new data emerges. Addressing class imbalances with advanced resampling techniques and cost-sensitive learning will further refine themodel's performance. Enhancing model interpretability and ensuring regulatory compliance will increase transparency and trust in the system. User experience can be improved through intuitive interfaces and personalized insights.

REFERENCES

1. J. Firmstone and S. Coleman, “The changing role of the local news media in enabling citizens to engage in local democracies,” Journalism Pract., vol. 8, no. 5, pp. 596–606, 2014.

2. Aphiwongsophon, S., & Chongstitvatana, P. (2018). Detecting Fake News with Machine Learning Method. 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 528-531. https://doi.org/10.1109/ECTICon.2018.8620051.

3. S. Della Vigna and E. Kaplan, “The Fox News effect: Media bias and voting,” Quart. J. Econ., vol. 122, no. 3, pp. 1187–1234, 2007.

4. M. Gentzkow and J. M. Shapiro, “Media bias and reputation,” J. Political Economy, vol. 114, no. 2, pp. 280–316, 2006.

5. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and resolution of rumours in social media: A survey. ACM Computing Surveys, 51(2), 1-36. https://dl.acm.org/doi/10.1145/3161603.

DATASET LINK:

https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets

GITHUB LINK

https://github.com/mahalakshmiaddal/Detecting-Fake-News