INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

AN INTELLIGENT FAKE NEWS DETECTION SYSTEM USING MACHINE LEARNING

[1]Mrs.G. Priyanka [2]Ms. N. Kavya

[1]gadadesipriyanka@gmail.com

[2]kavya.nagendla02@gmail.com

ABSTRACT

The phenomenon of fake news is rapidly evolving as communication and social media technologies advance. Fake news identification is an emerging scientific area that is gaining popularity. However, it confronts some obstacles due to limited resources such as datasets, processing, and analysis approaches. In this paper, we offer a system for detecting fake news using machine learning approaches. As a feature extraction strategy, we employed term frequency-inverse document frequency (TF-IDF) of a bag of words and n-grams, and a Support Vector Machine (SVM) classifier. We also present a dataset of fake and authentic news for training the suggested system. The obtained findings demonstrate the efficiency of the system. In this paper, we offer a system for detecting fake news using machine learning approaches. As a feature extraction strategy, we employed term frequency-inverse document frequency (TF-IDF) of a bag of words and n-grams, and a Support Vector Machine (SVM) classifier. We also present a dataset of fake and authentic news for training the suggested system. The obtained findings demonstrate the efficiency of the system.

This research, named "An Intelligent Fake News Detection System Using Machine Learning," describes a machine learning-based method for categorizing news stories as either real or fake news. The suggested approach uses a publicly available Kaggle dataset as well as additional Indian news articles to increase the model's capacity to categorize heterogeneous news items. To increase data quality, the acquired dataset is pre-processed using techniques such as text cleaning, lowercase conversion, punctuation removal, special characters, URL removal, and stop words. The suggested system provides a fast, reliable, and scalable approach for detecting fake news and can help consumers check online content before accepting or spreading it.

INTRODUCTION

In today's digital age, millions of people rely on the internet and social media for their news and information. The fast rise of online platforms has made it easier to get and spread news quickly. However, this convenience has resulted in the widespread distribution of fake news, which contains inaccurate or misleading material masquerading as legitimate news. The quick dissemination of fake news has the potential to influence public opinion, cause social unrest, harm reputations, and have a detrimental impact on society.

Traditional methods for detecting fake news rely on manual fact-checking by journalists and experts. Although these approaches are dependable, they demand a significant amount of time and work and cannot effectively handle the massive amount of web content published each day. As a result, there is an increasing demand for automated systems that can detect bogus news fast and precisely.

Machine Learning (ML) offers an excellent answer to this problem by allowing computers to learn patterns from previously categorized news articles and predict whether new articles are legitimate or not. This project, named "An Intelligent Fake News Detection System Using Machine Learning," creates a machine learning-based model that extracts features using TF-IDF and classifies text with Logistic Regression. To boost performance, the system is trained using a Kaggle dataset and extra Indian news items.

A user-friendly Stream lit web application is created to allow users to enter a news story and receive an instant forecast with confidence scores. The suggested model attained an overall accuracy of around 98.79%, proving its ability to recognize bogus news. This project provides a fast, dependable, and practical solution for recognizing deceptive material and contributes to encouraging trustworthy digital communication.

Objective:

The primary objective of this project is to develop an intelligent and efficient system for detecting fake news using Machine Learning techniques. The system is designed to classify news articles as **Real News** or **Fake News** by analyzing their textual content.

The specific objectives of the project are:

- To collect and preprocess news datasets for training and testing.
- To extract meaningful text features using the TF-IDF (Term Frequency–Inverse Document Frequency) technique.
- To build and train a **Logistic Regression** model for fake news classification.

INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

- To develop a user-friendly web application using **Stream lit** for real-time news prediction.
- To achieve high prediction accuracy and provide confidence scores for each prediction.
- To help users identify misleading information and promote the use of reliable news sources.

LITERATURE SURVEY

INTRODUCTION

A literature survey is an important component of any research endeavour since it gives an overview of previous work in the chosen topic. It aids in understanding the many approaches, algorithms, methodologies, and technologies employed by researchers to address the issue of false news identification. Previous study can also help highlight the limitations of current systems and provide guidance for designing a better solution.

Fake news can be detected using a variety of algorithms. To examine this, we apply a variety of classifiers across multiple research articles. The classifiers include Random Forest, CNN, SVM, KNN, Logistic Regression, Naive Bayes, LSM, and SGD. Random Forest yielded an accuracy of 83%, CNN yielded an accuracy of 97%, SVM yielded an accuracy of 94%, KNN yielded an accuracy of 79%, Logistic Regression yielded an accuracy of 97%, Naive Bayes yielded an accuracy of 90%, Long Short Term Memory yielded an accuracy of 97%, SVM and NB combined yielded an accuracy of 78%, and SGD yielded 77.2%.

SYSTEM ANALYSIS

EXISTING SYSTEM:

Research on machine learning for deception detection has primarily focused on online reviews and social media posts. Since late 2016, the literature has focused on identifying 'fake news' during the American presidential election. Conroy, Rubin, and Chen propose many promising ways to accurately classify misleading articles. Simple content-related n-grams and superficial parts-of-speech tagging are insufficient for classification tasks, as they typically miss critical contextual information. These strategies are most effective when used alongside more complex analysis methods. The primary method is manual human fact-checking or legacy keyword-based filtering. These methods are time-consuming, prone to human error, struggle with enormous volumes of daily data, and frequently fail to discover contextual nuances. Combining Probabilistic Context Free Grammars with n-gram approaches has proven to be highly effective for deep syntax analysis.

DISADVANTAGES OF EXISTING SYSTEM:

Depends primarily on manual verification, making the process slow and labor-intensive.

Traditional keyword-based approaches fail to capture the context or meaning of the information.

It is difficult to manage the vast number of news generated each day. There is a high potential of human error during manual fact-checking. Simple text analysis techniques, including n-grams and parts-of-speech tagging, frequently fail to detect sophisticated language patterns.

PROPOSED SYSTEM:

The suggested approach uses a Logistic Regression-based machine learning model to detect false news stories automatically. Initially, the news content is preprocessed, which includes text cleaning, tokenization, stop word removal, and feature extraction with TF-IDF. The processed features are then fed into the Logistic Regression classifier, which determines if the given news is authentic or fraudulent. This automated approach improves the efficiency of news verification while decreasing manual labor and producing consistent classification results. The suggested method detects fake news using a machine learning approach that analyzes the textual content of news stories. The incoming text is initially preprocessed by lowercasing, removing punctuation, and eliminating stop words. The cleaned text is next turned into numerical features by either a 'Count Vectorizer' or a 'TF-IDF Vectorizer', which indicate the relevance and frequency of words in the dataset. To determine if news items are authentic or false, a **Logistic Regression** classifier is trained using the retrieved characteristics. Feature selection is further improved by taking into account the most frequent words and n-grams while discarding low-frequency and irrelevant phrases. This approach improves classification accuracy and offers a quick, efficient, and dependable solution for detecting bogus news.

Advantages of proposed system:

Uses a Logistic Regression classifier for binary news classification.

Automatically detects bogus news with minimal human intervention.

Offers faster prediction than typical manual verification methods Manages big collections of news articles efficiently.

Reduces the likelihood of human error during classification.

Provides high accuracy for text-based categorization issues.

Can be updated with new datasets to increase forecast accuracy.

simple, computationally efficient, and straightforward to implement.

INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

SYSTEM DESIGN

5.1 *SYSTEM ARCHITECTURE:*

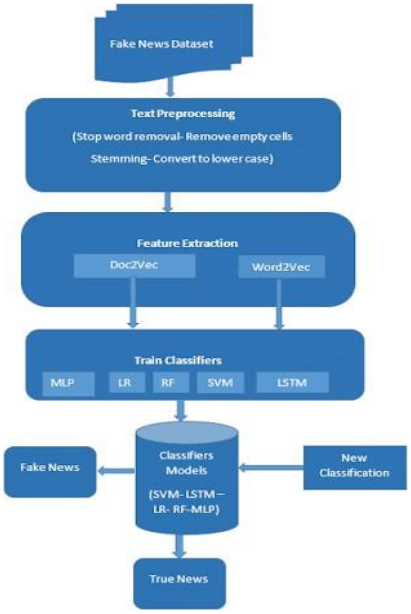


Fig.1 System architecture

UML DIAGRAMS

A UML (Unified Modeling Language) diagram is a visual tool used to model the architecture and interactions within a system. It provides a standardized way to depict relationships, workflows, and communications between system components. UML diagrams can include class diagrams for data structure, activity diagrams for workflows, and sequence diagrams for process interactions, ensuring clear system visualization and communication between developers and stakeholders.

Goals:

- UML provides a clear, organized visual representation of how system components interact.
- UML serves as a universal language between technical and non-technical stakeholders.
- It shows how different system parts like models and data repositories communicate.
- UML helps detect potential system inefficiencies or flaws during development.
- UML diagrams provide a blueprint for future updates, scaling, or maintenance of the system.

Use Case Diagram

A Use Case Diagram in the heart disease prediction system illustrates the interactions between healthcare professionals (users) and the system. It shows how users input patient biomarker data, trigger machine learning algorithms, and receive heart disease risk predictions. This diagram helps visualize key functionalities and user-system interactions for improved understanding of the system's operation.

INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

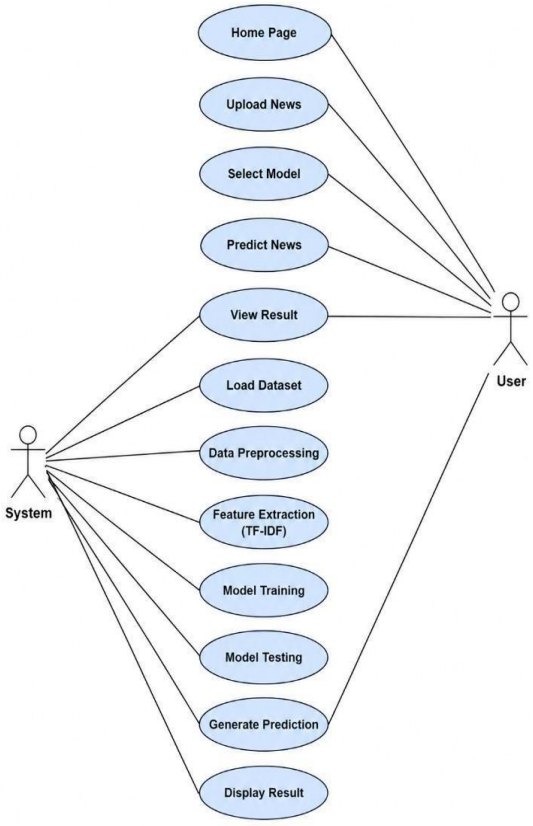
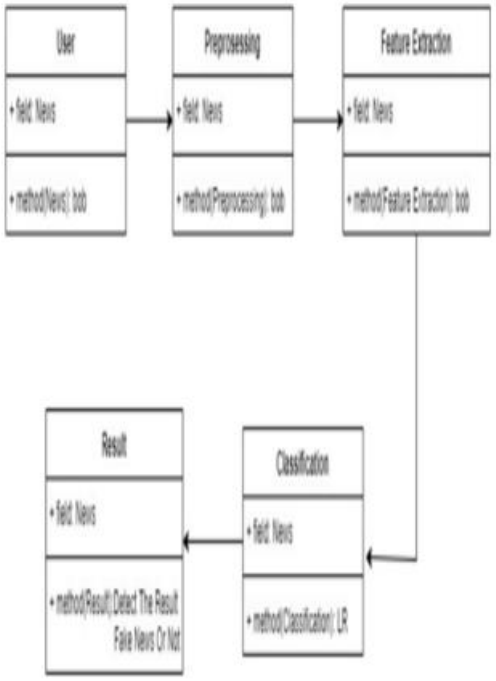


Fig.2 Use Case Diagram



INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

Class Diagram

Fig.3 Class Diagram

A Class Diagram in the heart disease prediction system models its structure by defining key classes such as patient data, biomarkers, machine learning algorithms, and prediction results. It represents the relationships, attributes, and methods of each class, ensuring clear organization and efficient system implementation.

Activity Diagram

An activity diagram for a fake news detection system depicts the step-by-step process of detecting misinformation, from user input to final classification. It typically separates into two main actors: the user and the system.

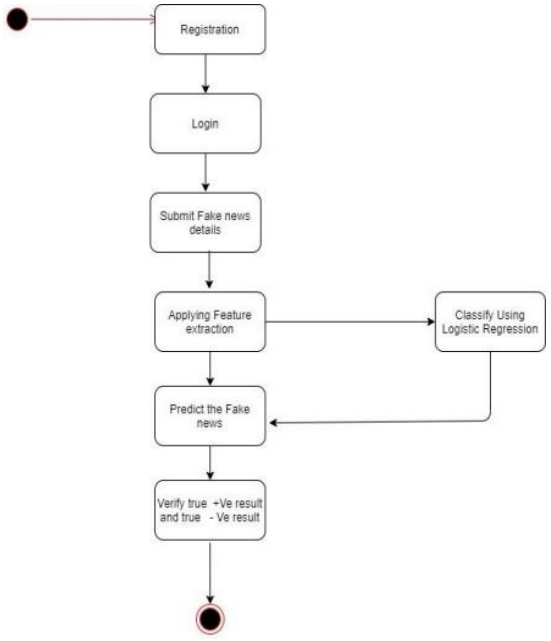


Fig.4 Activity Diagram

SAMPLE CODE

```
# Importing necessary libraries for data manipulation, analysis, and visualization: pandas, NumPy, Joblib
import streamlit as st import joblib
import re
# Load model and vectorizer
model = joblib.load("models/model.pkl") vectorizer = joblib.load("models/vectorizer.pkl")
# Text cleaning function def clean_text(text):
text = str(text) text = text.lower()
text = re.sub(r"http\S+|www\S+", "", text) text = re.sub(r"[^a-zA-Z\s]", "", text)
text = re.sub(r"\s+", " ", text).strip() return text
# App Title
st.title(" ■ Fake News Detection System")
st.write("Enter a news article below and click Predict.") # Text input
news_text = st.text_area( "News Article", height=250
)

# Predict button
if st.button("Predict"):
if news_text.strip() == "": st.warning("Please enter some news text.")
else:
cleaned_text = clean_text(news_text)
```

INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

```
vectorized_text = vectorizer.transform([cleaned_text]) prediction = model.predict(vectorized_text)[0] probabilities =
model.predict_proba(vectorized_text)[0]
fake_prob = probabilities[0] * 100 real_prob = probabilities[1] * 100
st.subheader("Result") if prediction == 0:
st.error("▼ FAKE NEWS")
else:
st.success(" ■REAL NEWS") st.subheader("Confidence")
confidence = max(fake_prob, real_prob) st.progress(int(confidence))
st.write(f"● Fake Probability: {fake_prob:.2f}%") st.write(f"● Real Probability: {real_prob:.2f}%") st.write(f"★
Confidence: {confidence:.2f}%")
if confidence < 60:
st.warning(
"Low confidence prediction. The model is uncertain about this article."
)
import joblib import re
# Load saved model
model = joblib.load("models/model.pkl") vectorizer =
joblib.load("models/vectorizer.pkl")
def clean_text(text):
text = str(text) text = text.lower()
text = re.sub(r"http\S+|www\S+", "", text) text = re.sub(r"[^a-zA-Z\s]", "", text)
text = re.sub(r"\s+", " ", text).strip() return text
# Example news news = """
The U.S. Federal Reserve announced on Wednesday that it would keep interest rates unchanged while continuing to
monitor inflation and employment data. Officials stated that future decisions would depend on economic conditions and
incoming financial indicators.
"""
# Clean text
cleaned_news = clean_text(news)

# Convert to TF-IDF
vectorized_news = vectorizer.transform([cleaned_news])

# Predict
prediction = model.predict(vectorized_news)[0]
probabilities = model.predict_proba(vectorized_news)[0] fake_prob = probabilities[0] * 100
real_prob = probabilities[1] * 100

print(f"Fake Probability: {fake_prob:.2f}%") print(f"Real Probability: {real_prob:.2f}%")

if prediction == 0: #print("\nPrediction: FAKE NEWS") else:
#print ("\nPrediction: REAL NEWS")
import joblib  import pandas as pd import re
from sklearn.feature_extraction.text import TfidfVectorizer from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression from sklearn.metrics import (
accuracy_score, classification_report, confusion_matrix
)

# Load datasets
fake = pd.read_csv("dataset/Fake.csv") true = pd.read_csv("dataset/True.csv")
indian_fake = pd.read_csv("dataset/indian_fake.csv") indian_real = pd.read_csv("dataset/indian_real.csv")
# Add labels fake["label"] = 0
true["label"] = 1
```

INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

```
indian_fake["label"] = 0
indian_real["label"] = 1

data = pd.concat(
[fake, true, indian_fake, indian_real], ignore_index=True
)

print("Total Articles:", len(data)) # Keep useful columns
data = data[["text", "label"]]

# Cleaning function def clean_text(text):
text = str(text) text = text.lower()
text = re.sub(r"http\S+|www\S+", "", text) text = re.sub(r"[^a-zA-Z\s]", "", text)
text = re.sub(r"\s+", " ", text).strip() return text
# Clean text
data["clean_text"] = data["text"].apply(clean_text)

# TF-IDF
vectorizer = TfidfVectorizer( stop_words="english", max_df=0.7,
min_df=5, max_features=50000, ngram_range=(1,2)
)

X = vectorizer.fit_transform(data["clean_text"]) y = data["label"]

# Split dataset
X_train, X_test, y_train, y_test = train_test_split( X,
y, test_size=0.2,
random_state=42
)

print("Training Data:", X_train.shape) print("Testing Data:", X_test.shape)

# Train model
model = LogisticRegression(max_iter=1000)

print("\nTraining model...") model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Accuracy
accuracy = accuracy_score(y_test, y_pred)

print("\nModel Accuracy:") print(f"{accuracy * 100:.2f}%") print("\nClassification Report:")
print(classification_report(y_test, y_pred))

print("\nConfusion Matrix:") print(confusion_matrix(y_test, y_pred)) # Save model
joblib.dump(model, "models/model.pkl") # Save vectorizer
joblib.dump(vectorizer, "models/vectorizer") print("\nModel saved successfully!")
```

RESULTS

This chapter includes screenshots of the created **Fake News Detection System**. The screenshots exhibit the application's functionality, from user input to prediction output. T The system has a simple and interactive interface that allows users to classify news stories as **Real News** or **Fake News** using confidence scores.

INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

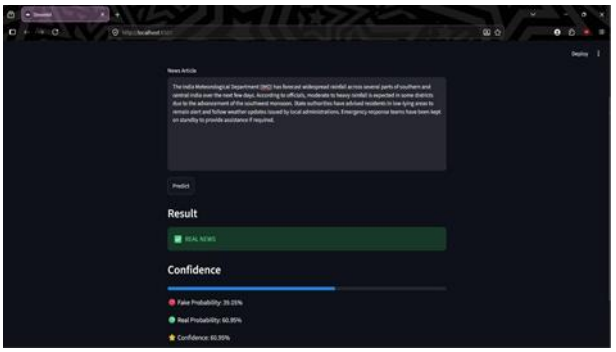
Vol. 01, Issue 3, July2026, pp :129-138

HOME PAGE

The home page is the application's main interface. The user can enter or paste the news story into the text field and then click the **Predict** button to begin the classification process.

Fig.5 Home Page

REAL NEWS PREDICTION



When you enter a genuine news story into the application, this screen displays the prediction result. The system classifies the article as **Real News** and displays its confidence score.

Fig.6 Real News Prediction



FAKE NEWS PREDICTION

When you enter a fake news article, this screen provides the prediction result. The application classifies the article as **Fake News** and assigns a confidence score.

INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

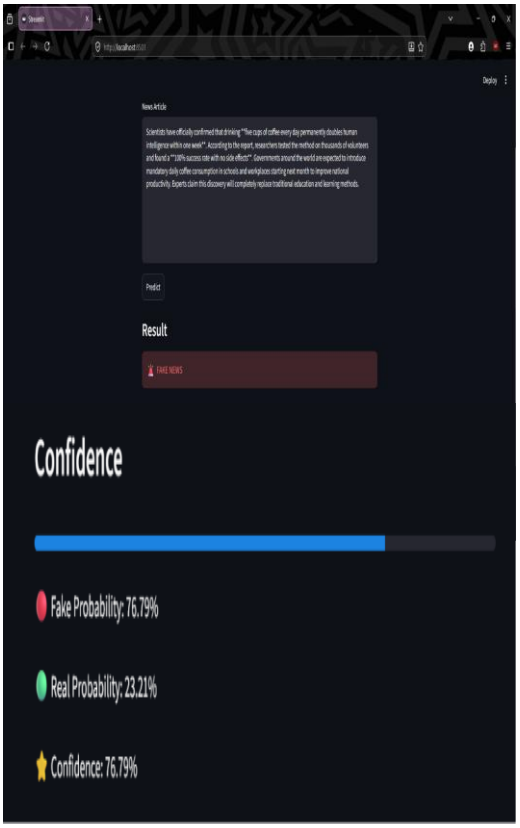


Fig.7 Fake News Prediction

CONFIDENCE SCORE

The application displays the confidence score for each forecast. The confidence value represents how confident the machine learning model is in its prediction.

Fig.8 Confidence Score

PROJECT EXECUTION

The screenshot below depicts the successful execution of the stream lit program and the prediction result.

INTERNATIONAL JOURNAL OF ENGINEERING MANAGEMENT AND SOCIAL SCIENCES (IJEMSS)

e-ISSN : 3139-065X

www.ijemss.com

editor@ijemss.com

Vol. 01, Issue 3, July2026, pp :129-138

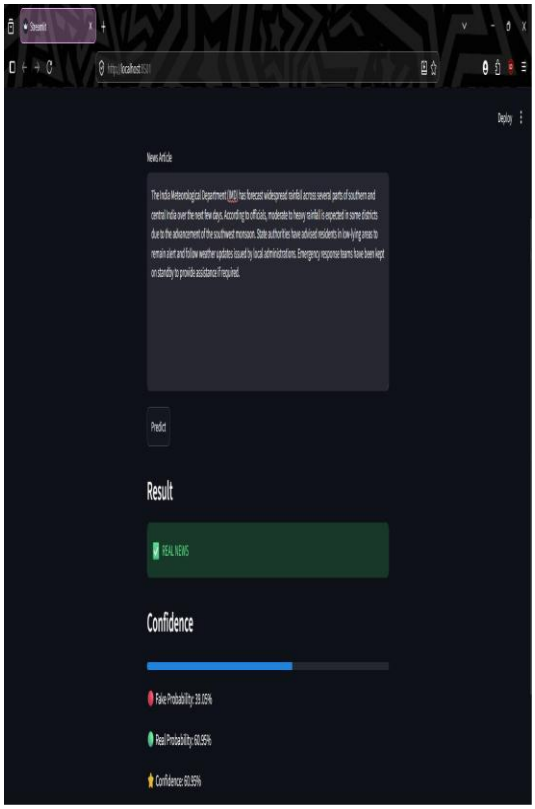


Fig.9 Project Execution

CONCLUSION

Today, social media is the primary source of news consumption, surpassing conventional media. Fake news has a significant negative influence on both users and society. To detect false news, we analyzed research articles and found that word embedding, tokenization, and part-of-speech tagging are effective for pre-processing data, while TF-IDF and Count Vectorizer are optimal for feature extraction. To improve accuracy, we plan to apply pre-processing methods such as Random Forest, Convolutional Neural Networks, Long Short-Term Memory, and Ensemble Learning.

REFERENCE

1 Almeida, L.D., Fuzaro, V. Nieto, F., & Santana, A.L.M. (2021). Identificação de “Fake News” no contexto político Brasileiras: uma abordagem computacional. In: Proceedings Workshop sobre as Implicações da Computação na Socie-dade (WICS), Porto Alegre, Brasil, pp. 78-89. https:// doi.org/10.5753/wics.2021.15966 Abouzeid, A., Granmo, O.C., Webersik, C., & Goodwin, M. (2019).

2 Causality-based Social Media Analysis for Normal Users Credibility Assessment in a Political Crisis. Proceed-ings of 25th Conference of Open Innovations Association (FRUCT), Helsinki, Finland, pp. 1-14. https://doi.org/10.23919/FRUCT48121.2019.8981500 Agarwal, A., Mittal, M., Pathak, A., & Goyal, L.M. (2020).

3 Fake News Detection Using a Blend of Neural Networks: An Application of Deep Learning. SN Computer Science, 1,3:1-9.