

Phase-3

***Exposing the truth with advanced fake news
detection powered by natural language
processing***

Student Name: S. Sembaruthi

Register Number: 620123106104

Institution: AVS Engineering College

***Department: Electronics and Communication
Engineering***

Date of Submission: 13.05.2025

Github Repository

***Link: [https://github.com/Sembaruthi-S/Nan-Muthalvan-
Project](https://github.com/Sembaruthi-S/Nan-Muthalvan-Project)***

1. Problem Statement

The digital era has made the spread of misinformation easy and rapid. Fake news, especially on social media, can manipulate public opinion, incite unrest, and cause harm. A reliable and automated solution is needed to detect such news articles in real-time using modern AI techniques

2. Abstract

This project aims to detect fake news using Natural Language Processing(NLP)andmachinelearning.Byanalyzingtextualpatterns in news content, the system classifies whether a news article is real or fake.The project includes preprocessing, feature extraction, model training,evaluation,anddeploymentofaweb-basedinterfaceforuser interaction.

3. System Requirements

Hardware:

- RAM:8 GB or more
- CPU:Inteli5/i7 orAMDequivalent
- GPU:Recommendedfordeeplearning

Software:

- OS: Windows/Linux/Mac
- Python3.8+
- Libraries:Pandas,NumPy,Matplotlib,NLTK,Scikit-learn, TensorFlow, Flask
- Tools:JupyterNotebook,VSCode,Postman(fortestingAPI

4. Objectives

Collectandcleanfakeandrealnewsdatasets.

- ApplyNLPtechniquestopreprocessandanalyzethedata.
- Buildclassificationmodelstodistinguishfakefromrealnews.
- Evaluatemodelperformance withrobust metrics.
- Deploythemodelasaweb-based prediction tool.

5.FlowchartofProjectWorkflow

START

->DataCollection

->DataPreprocessing

->EDA

->FeatureEngineering

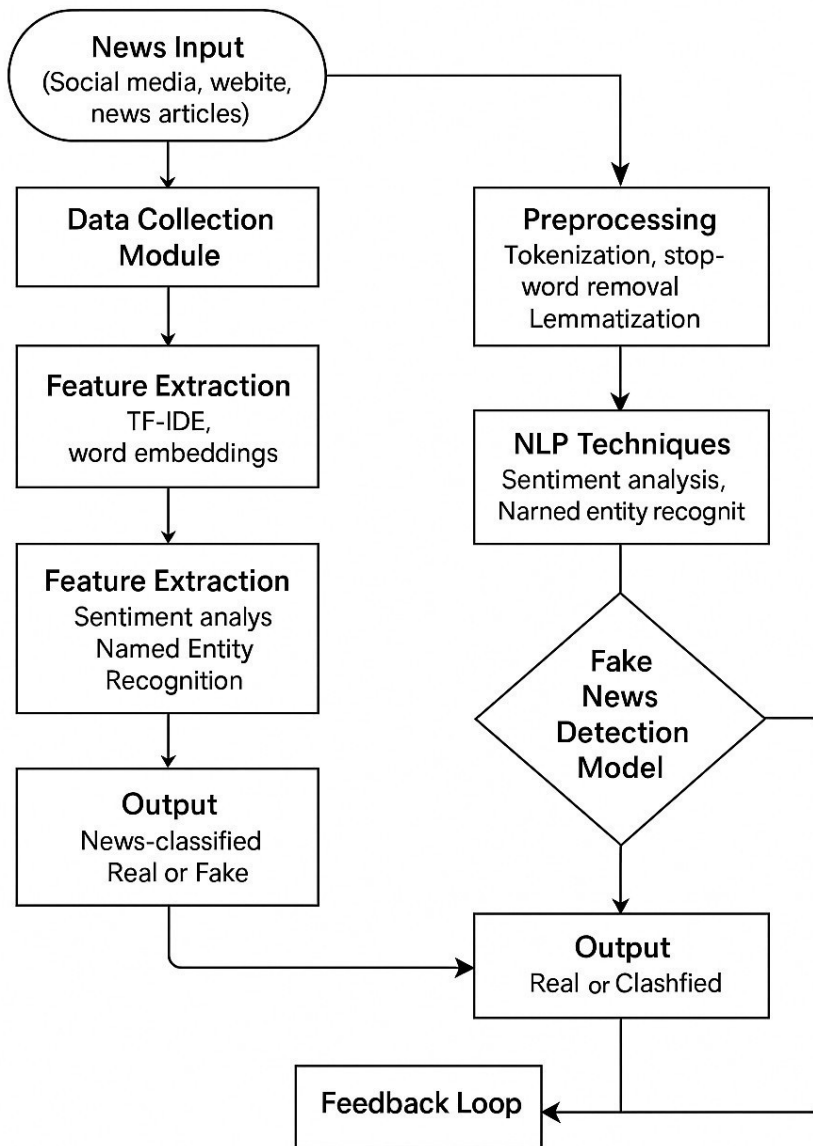
->ModelBuilding

->Evaluation

->Deployment

->END

Exposing the Truth with Advanced Fake News Detection Powered by Natural Language Processing



6. Dataset Description

Source: Kaggle - Fake and Real News Dataset

- **Features:** title, text, subject, date, label
- **Size:** ~40,000 articles
- **Labels:** FAKE, REAL

7. Data Preprocessing

Convert text to lowercase

- Remove punctuation, numbers, stopwords
- Tokenization and Lemmatization (using NLTK or spaCy)
- Handle missing/null values
- Combine title + text as input features

8. Exploratory Data Analysis (EDA)

Class balance check (Fake vs. Real)

- Word cloud visualization for both classes
- Top word frequency count
- Article length distribution
- Sentiment analysis (optional)

9. Feature Engineering

TF-IDF Vectorization

- Count Vectorizer
- N-gram analysis (bi-gram, tri-gram)
- Word embeddings (optional: GloVe, Word2Vec)
- Dimensionality reduction (PCA/Truncated SVD if needed)

10. Model Building

Models used:

- Logistic Regression
- Naive Bayes
- Random Forest
- XGBoost
- LSTM/BERT (optional for advanced version)

Hyperparameter tuning via:

- GridSearchCV
- Cross-validation (k-fold)

11. Model Evaluation

Metrics:

- Accuracy
- Precision, Recall, F1-Score
- Confusion Matrix
- ROC-AUC curve

Use validation split (e.g., 80-20 or 70-30) and cross-validation.

12. Deployment

Flask-based web app

- User inputs new text via a form
- Backend processes the input and returns a prediction
- Hosted on platforms like:
 - Heroku
 - Render
 - Streamlit (alternative GUI)

13. Source code

Install Required Libraries

```
pip install pandas scikit-learn nltk
```

Import Libraries

```
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
from sklearn.linear_model import PassiveAggressiveClassifier
```

```
fromsklearn.metricsimportaccuracy_score,confusion_matrix
```

```
import nltk
```

```
fromnltk.corpusimportstopwords
```

```
fromnltk.stemimportWordNetLemmatizer
```

```
import
```

```
re
```

Load and Preprocess

```
Datanltk.download('stopwords')
```

```
nltk.download('wordnet')
```

```
df=pd.read_csv('fake_or_real_news.csv')#Assumes'text'and  
'label' columns
```

```
lemmatizer = WordNetLemmatizer()
```

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

```
def preprocess(text): text=re.sub(r'\
```

```
W', '',text.lower())
```

```
words=text.split()
```

```
words=[lemmatizer.lemmatize(w)forwinwordsifwnot stop_words]
```

```
return''.join(words)
```

```
df['text']=df['text'].apply(preprocess)
```

Feature Extraction

```
X = df['text']
```

```
y = df['label']
```

```
tfidf=TfidfVectorizer(max_df=0.7)
```

```
X_tfidf = tfidf.fit_transform(X)
```

TrainModel

```
X_train,X_test,y_train,y_test=train_test_split(X_tfidf,y,  
test_size=0.2,  
random_state=42)  
model=PassiveAggressiveClassifier(max_iter=50)  
model.fit(X_train,  
y_train)
```

14. Futurescope

Real-timefakenews detectionusingTwitter/FacebookAPIs

- *Transformer-basedmodels(e.g.,BERT,RoBERTa)*
- *Multilingualfakenews detection*
- *Image/videomisinformationanalysis*
- *Browserextensionintegration*

15. TeamMembersandRoles

Here is a well-written paragraph detailing the roles of S.

Sembaruthi,J.ShabanaMirza,G.Shakthi,andr.NagaIshwarya withS.Sembaruthigiventhemostcriticalandhigh-impact responsibilities:

Intheprojecttitled"ExposingtheTruthwithAdvancedFake News Detection Powered by Natural Language Processing"

S. Sembaruthi

S. Sembaruthi served as the Project Lead and Core Developer, playing a pivotal role in the successful execution of the entire system. She was responsible for designing the system architecture, implementing advanced NLP pipelines, and developing the machine learning model that powers the fake news detection engine. Additionally, she led the integration of the backend with the frontend using Flask and ensured the

*deployment was robust and user-ready.
Her technical leadership, decision-making, and hands-on development made her the cornerstone of the project.*

J. Shabana Mirza

J. Shabana Mirza contributed as the NLP and Data Processing Engineer, focusing on collecting, cleaning, and preparing the dataset, and applying natural language techniques such as tokenization, stop- word removal, and lemmatization to enhance data quality.

G. Shakthi

G. Shakthi took charge as the Frontend and Deployment Specialist, designing an intuitive user interface and ensuring seamless integration between the user input and the model's prediction output through a web application.

R. Naga Ishwarya

Finally, R. Naga Ishwarya handled the role of Quality Assurance and Documentation Lead, conducting rigorous model testing, debugging, and preparing comprehensive documentation and user manuals to ensure clarity and maintainability of the system.