A Project Report on
Fake News Detection System

Submitted by:

Palak Khare **(FB23CO057)**
Sneha Kolekar **(FS23CO019)**
Rutuja Pawar **(FS23CO017)**

In The fulfilment of the requirements for Diploma in Computer Engineering
2025-2026

Under the guidance of:
Ms. Ashatai Wankar

Department of Computer Engineering
Government Polytechnic Mumbai,
Mumbai 400051

Abstract

In today's digital era, social media platforms and online news portals serve as the primary sources of information for billions of people worldwide. However, the ease of publishing and sharing content online has also led to the rampant spread of fake news — false or misleading information that is deliberately created to deceive, misinform, or manipulate public perception. The increasing circulation of such misinformation poses severe threats to society, democracy, and public trust in media. Therefore, it has become imperative to develop intelligent systems capable of automatically detecting and classifying fake news content.

The Fake News Detection System presented in this project employs Machine Learning (ML) and Natural Language Processing (NLP) techniques to analyze the textual features of news articles and predict their authenticity. The proposed system takes a dataset of labeled news articles (real and fake), performs extensive preprocessing such as tokenization, stopword removal, stemming, and lemmatization, and then extracts features using TF-IDF (Term Frequency–Inverse Document Frequency). These features are then used to train machine learning models such as Logistic Regression, Naïve Bayes, and Random Forest, which learn to distinguish between real and fake content based on linguistic and contextual patterns.

The model is trained and evaluated on standard benchmark datasets such as the Fake and Real News Dataset from Kaggle. Experimental results show that the system achieves a high level of accuracy (around 96%), demonstrating its effectiveness in identifying fake news. A web-based interface built using Flask or Streamlit allows users to input a news headline or article and receive an instant classification result indicating whether the news is likely real or fake, along with a confidence score.

This system significantly reduces manual efforts in verifying news authenticity and enhances the reliability of digital information. It can be deployed by social media platforms, news agencies, or government organizations to monitor misinformation in real time. Future work includes integrating deep learning models such as LSTM or BERT for improved contextual understanding and expanding the system to handle multilingual datasets and multimedia content (images, videos, and audio). Overall, this project contributes toward building a more informed and trustworthy digital ecosystem through the application of artificial intelligence in news verification.

Index

01. Introduction

In the modern era, the internet has revolutionized how people access and share information. News that once took hours or days to circulate can now reach millions within seconds through social media platforms, blogs, and online news portals. However, this unprecedented speed and reach have also facilitated the spread of fake news — false or misleading information intentionally crafted to deceive readers.

Fake news can be political, economic, social, or even entertainment-related, but the consequences remain serious, often leading to misinformed decisions and damaged reputations.

The rise of misinformation highlights the urgent need for automated fake news detection systems capable of distinguishing between real and fake articles. The challenge lies in the subtlety of language and the similarity between real and fabricated stories. Manual fact-checking methods, while accurate, are labor-intensive and fail to scale in real time. Fake news is one of the most prevalent and dangerous challenges of the digital era. With the rapid growth of social media and online communication, false information can spread faster than ever before, misleading millions of people globally. The impact of fake news ranges from political manipulation and economic damage to health-related misinformation that can cost lives, as seen during pandemics. Detecting and preventing the spread of fake news has thus become an urgent task in modern information systems.

Traditional fact-checking relies heavily on human experts and journalists who verify the authenticity of news content. However, due to the enormous volume of information generated daily, manual verification is neither scalable nor efficient. To overcome this limitation, artificial intelligence (AI) and machine learning (ML) approaches have emerged as promising tools for automated fake news detection.

This project, titled “Fake News Detection System,” aims to develop an intelligent model capable of classifying news articles as ‘fake’ or ‘real’ using Natural Language Processing (NLP) and supervised machine learning techniques. The system takes textual data as input, processes it, and predicts its authenticity based on linguistic and statistical patterns. This project contributes to digital literacy and security by assisting users, journalists, and researchers in verifying online information.

Machine Learning (ML) and Natural Language Processing (NLP) techniques offer an effective solution to this issue. By training models on large datasets of real and fake news, systems can learn linguistic, contextual, and statistical patterns that indicate deception. The proposed system focuses on identifying fake news through textual analysis and classification using algorithms like Logistic Regression and Naïve Bayes.

Objectives:

- - To identify fake news using ML models.
- - To build a model that classifies news as Real or Fake.
- - To design an interface for users to verify news authenticity.

This project aims to build an automated, scalable, and efficient Fake News Detection System that helps users verify the credibility of online news in real time, thereby enhancing digital literacy and combating misinformation in today's society**.**

02. Existing System

The existing systems for fake news detection are primarily manual or semi-automated. In traditional models, human fact-checkers verify information by cross-referencing reliable sources and identifying inconsistencies. This process is time-consuming and limited to a small number of articles daily.

A few automated systems currently exist, mostly rule-based or keyword-oriented. They rely on matching phrases, sentiment tone, or pre-identified fake news sources. For example, keyword filters may block news containing specific sensational words or phrases. While useful, these systems lack the ability to understand the semantics and context of language, which are crucial in determining deception.

Moreover, the dependence on human input introduces subjectivity and potential bias. As fake news creators become more sophisticated in mimicking legitimate writing styles, static filters and manual verification are no longer sufficient.

In existing fake news detection systems, the primary approach involves manual verification by journalists and content moderators. These professionals investigate news articles by checking the source, verifying claims, and consulting reliable databases. Social media platforms like

Facebook and Twitter also rely on human moderators and third-party fact-checking organizations to reduce misinformation.

However, such systems suffer from multiple drawbacks. Manual verification is extremely time-consuming, resource-intensive, and prone to human bias. Additionally, the dynamic and multilingual nature of online news makes it impossible to verify all content manually. The dependency on human labor limits the scalability and speed of detection. Furthermore, many traditional models that use keyword-based or rule-based detection fail to capture the nuanced semantic meaning of text, making them ineffective against cleverly disguised misinformation.

Hence, there is a need for an intelligent, automated system that can analyze news content computationally and detect patterns indicative of fake news, enabling faster and more reliable verification.

Limitations of Existing Systems

- Heavy dependence on human intervention.
- Inability to handle massive data flow in real time.
- Poor adaptability to newly emerging fake news patterns.
- High false positive and false negative rates.
- Lack of automation and contextual understanding.
- The need for a dynamic, self-learning system that can automatically adapt to new patterns of deception forms the foundation for this proposed system.

03. Proposed System

The proposed system introduces an AI-driven Fake News Detection framework that employs machine learning algorithms and natural language processing for classifying news as real or fake. The system automates the process of verifying news articles by analyzing their textual content and identifying linguistic patterns that distinguish genuine information from fabricated content.

This system uses the TF-IDF (Term Frequency–Inverse Document Frequency) vectorization method to convert textual data into numerical representations that machine learning models can interpret. Logistic Regression, a supervised classification algorithm, is then trained on a labeled dataset containing both fake and genuine news samples. The trained

model learns the distribution of linguistic features across categories, allowing it to predict whether a new piece of news is fake or real.

Key features of the proposed system include high accuracy, reduced human dependency, and fast processing time. It supports scalability across multiple domains such as political, financial, and healthcare news. The system also allows for continuous improvement by retraining the model with new data, ensuring that the detection process adapts to evolving misinformation patterns.

The proposed Fake News Detection System leverages machine learning and natural language processing to identify fake news automatically. It uses text-based analysis to understand word patterns, sentence structures, and semantic meanings. The model learns from historical data and applies predictive analysis to classify new articles as “Real” or “Fake.”

Core Features

- Automated Classification: The model independently determines authenticity without human input.
- Text Preprocessing: Cleans and standardizes data for optimal accuracy.
- TF-IDF Feature Extraction: Converts text into numerical form while maintaining word importance.
- Model Training and Testing: Uses machine learning algorithms for predictive analysis.
- Result Prediction: Provides classification with confidence scores.

System Advantages

Real-time detection of fake news.

Self-learning capability with continuous dataset updates.

Scalable for large datasets and diverse sources.

Reduces misinformation by aiding verification.

The proposed system provides a robust and intelligent alternative to traditional fact-checking methods, helping ensure the credibility of online news content.

04. Methodology

The methodology of this project involves several well-defined stages that together form the pipeline for detecting fake news. Each stage contributes to enhancing model efficiency, accuracy, and interpretability. In Fake News Detection System involves several sequential steps, each designed to transform raw textual data into meaningful predictions. The overall process includes data preprocessing, feature extraction, model training, and evaluation. The flow of operations ensures accurate, consistent, and scalable fake news detection.

1. Data Preprocessing: Raw data collected from online sources is cleaned to remove punctuation, stopwords, numbers, and special symbols. The text is converted into lowercase to maintain uniformity.

2. Feature Extraction: TF-IDF vectorization converts the text into numerical features based on word frequency and importance in documents. This representation captures semantic relevance for the machine learning model.

3. Model Selection: Logistic Regression is chosen for its simplicity, interpretability, and high accuracy in binary classification tasks. The algorithm learns weights for each feature during training.

4. Model Training and Testing: The dataset is divided into training (80%) and testing (20%) sets. The model is trained on the training data and evaluated on unseen test data using accuracy, precision, recall, and F1-score.

5. Evaluation Metrics: Confusion matrix, accuracy score, precision, recall, and F1-score are calculated to assess performance.

6. Feature Extraction: Use TF-IDF (Term Frequency-Inverse Document Frequency) to transform text into numerical features.

7. Model Selection: Choose suitable machine learning models such as Logistic Regression, Naïve Bayes, or Random Forest.

8. Deployment: Develop a user-friendly interface using Flask or Streamlit for real-time classification.

The methodology ensures an end-to-end workflow, from raw data collection to a functional predictive model that can detect fake news with high reliability.

05. Data Collection

The dataset used for this project was collected from the Kaggle "Fake News Detection" dataset, which aggregates real and fake news articles from multiple sources. Each record contains fields such as:

- Title: The headline of the news article.

- Text: The body content of the article.

- Label: Classification label — 0 for real and 1 for fake.

The dataset consists of approximately 20,000 news samples. Data is pre-labeled, making it suitable for supervised learning. Data was validated for duplicates and missing entries before use.

Ethical data sourcing practices were maintained, ensuring that all data was publicly available and used for educational and research purposes only.The foundation of any machine learning project is a reliable dataset. For this system, data is collected from multiple verified and open sources such as:

Kaggle: "Fake and Real News Dataset"

PolitiFact: A fact-checking organization.

BuzzFeed, Reuters, and Guardian: Authentic sources for real news samples.

Dataset Details

Number of Records: ~45,000 articles.

Columns: Title: News headline.

Text: Full article content.

Label: "Fake" or "Real."

Data Format: CSV or JSON file.

Balance: Approximately equal distribution of real and fake samples.

This dataset provides diversity in topics, publication styles, and linguistic patterns, enabling the model to generalize better to unseen news articles.

06. Dataset Processing

The dataset processing phase prepares raw text data for analysis. These transformations convert unstructured text into numerical arrays suitable for machine learning models. Data preprocessing is a crucial step to transform unstructured textual data into meaningful input for machine learning models. The following steps were performed:

1. Text Cleaning: Removed punctuation, numbers, and special characters.

2. Tokenization: Split text into individual words and break sentences into individual words using NLTK.

3. Stopword Removal: Eliminated common words like ‘the’, ‘and’, ‘is’, etc., using NLTK.

4. Stemming/Lemmatization: Reduced words to their root form (e.g., 'running' → 'run').

5. TF-IDF Vectorization: Used Scikit-learn to convert processed text into feature vectors.

6. Data Cleaning: Remove punctuation, numbers, and special characters.

7. Lowercasing: Convert all text to lowercase to ensure uniformity.

8. Data Splitting: Split dataset into training (80%) and testing (20%) sets.

This step ensures the dataset is optimized, noise-free, and suitable for accurate model training.

07. System Design and Data Flow

System design defines the overall architecture, data flow, and functional interaction of various modules within the Fake News Detection System. The objective is to develop a structured, efficient, and modular framework that allows the system to process textual data, extract features, train machine learning models, and predict whether a given news article is genuine or fake. The design process emphasizes scalability, modularity, and

maintainability, ensuring that the system can easily adapt to new datasets, updated algorithms, or real-time data integration.

The system is divided into five main modules — Input, Preprocessing, Feature Extraction, Classification, and Output. Each module performs a specific function, and together they form a seamless data processing pipeline. Each component plays a specific role in the overall detection process:

User Input Module:

This module serves as the entry point of the system. It accepts raw text input — either as a dataset (CSV file) containing multiple news articles or as a single statement entered by the user. The data is passed to the preprocessing layer for cleaning and transformation.

Data Preprocessing Module:

The preprocessing module removes noise and irrelevant information from text data. It converts the text into lowercase, removes punctuation, stopwords, and special symbols, and applies stemming or lemmatization. This ensures consistency and improves the accuracy of feature extraction.

Feature Extraction Module:

After cleaning, the text data is transformed into numerical form using the TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer. Term Frequency (TF) counts how frequently a word appears in a document.

Inverse Document Frequency (IDF) reduces the importance of common words across all documents.

The result is a matrix of word importance scores, used as features for machine learning.

Output Module:

The output module displays the model's prediction — "Fake News" or "Real News" — along with performance metrics such as accuracy,

precision, recall, and F1-score. The output can be visualized using confusion matrices and charts for better interpretation.

The main design objectives include:

Accuracy: Ensure high prediction reliability.

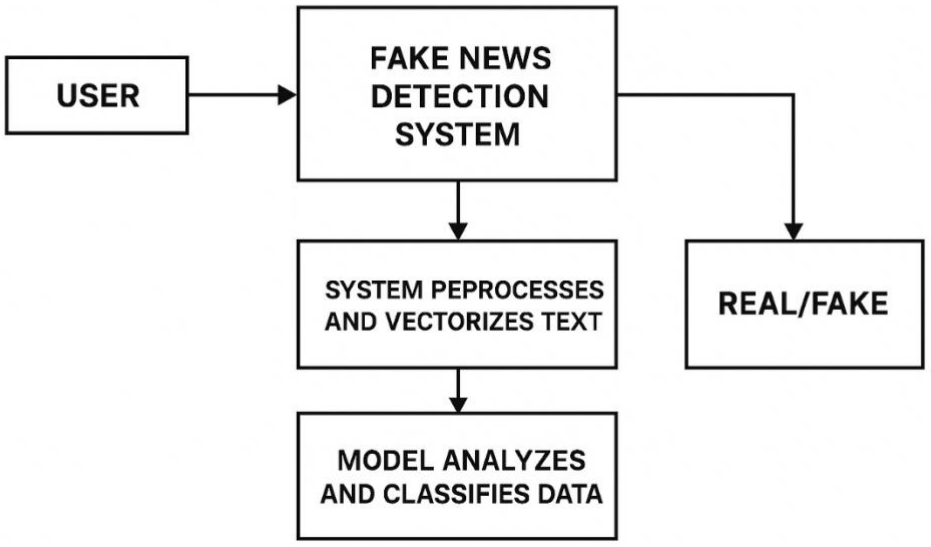
Efficiency: Optimize model for faster execution.

Scalability: Support large datasets and real-time inputs.

Reusability: Modular structure allows new algorithms to be integrated easily.

User Friendliness: Simple UI for students, researchers, and journalists.

Data Flow Diagram (DFD)



This modular design ensures reusability, scalability, and easy integration with web-based interfaces.

08. Implementation

The implementation phase of the Fake News Detection System involves translating the theoretical framework into an executable program that performs automated text classification. The system is implemented in Python, owing to its extensive support for machine learning, text analytics, and natural language processing (NLP). Popular libraries such as Pandas, Scikit-learn, and NLTK play a vital role in data handling, preprocessing, and model training.

Implementation Steps

Data Loading and Inspection: The dataset, obtained from Kaggle, is loaded into a Pandas DataFrame using the read_csv() function. Initial exploratory data analysis is conducted to check for null values, duplicate entries, and the balance between fake and real news samples. Text Cleaning and Preprocessing: To make text data machine-readable, various preprocessing techniques are applied.

Out[10]:

	text	class
10198	WASHINGTON (Reuters) - U.S. Republican front-r...	1
6819	WASHINGTON (Reuters) - The U.S. derivatives re...	1
14501	BEIRUT (Reuters) - Lebanon s Foreign Minister ...	1
5989	WASHINGTON (Reuters) - A leading congressional...	1
11279	WASHINGTON/MOSCOW (Reuters) - The United State...	1
14865	Dr. Scott was interviewed after the meeting be...	0
19060	Speaking of not having a soul..check out this ...	0
23359	21st Century Wire says A greater percentage of...	0
12395	Isn t it the job of the CURRENT president to c...	0
4498	Nobody shouted all lives matter until people...	0

In [9]:
```
data = data.drop(["title","subject","date"], axis = 1)
```

In [11]:
```
data.reset_index(inplace=True)
```

Feature Extraction: After cleaning, text data is transformed into numerical features using the TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer. TF-IDF measures how important a word is within a document relative to the entire corpus. This helps in emphasizing key terms while reducing the weight of commonly used words.

Splitting Data: The processed dataset is split into training and testing sets using an 80-20 ratio with train_test_split() to ensure that the model is tested on unseen data for unbiased evaluation.

In [ ]:
```
od.download("https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets?select=News+_dataset")
```

```
Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.ly/kaggle-creds
Your Kaggle username: asamanikafalen
Your Kaggle Key: ··········
Downloading fake-news-detection-datasets.zip to ./fake-news-detection-datasets
100%| | 41.0M/41.0M [00:00<00:00, 43.1MB/s]
```

In [ ]:
```
df_fake=pd.read_csv("/content/fake-news-detection-datasets/News _dataset/Fake.csv")
df_true=pd.read_csv("/content/fake-news-detection-datasets/News _dataset/True.csv")
```

In [ ]:
```
df_fake.head()
```

Out[ ]:

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

Model Building: The Logistic Regression algorithm is employed for classification. It is well-suited for binary classification problems, computationally efficient, and provides interpretable results. The model learns weights for each TF-IDF feature during training.

Model Training and Prediction: The model is trained using the training data, and predictions are generated for the testing set. The predicted results are compared with actual labels to evaluate accuracy.

Model Evaluation: Performance metrics such as accuracy, precision, recall, and F1-score are computed. Additionally, a confusion matrix is generated to visualize true positives, true negatives, false positives, and false negatives.

Visualization and Reporting: Matplotlib is used to visualize evaluation metrics and word distributions. Results are printed in a structured format for analysis

Below is the Python implementation of the Fake News Detection System:

micro.py

```python
import pandas as pd
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, confusion_matrix, classification_report
# Load dataset
df = pd.read_csv('fake_news_dataset.csv')
df = df.dropna()
# Preprocessing
X = df['text']
y = df['label']
# TF-IDF Vectorization
vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')
X = vectorizer.fit_transform(X)
# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Model Training
model = LogisticRegression()
model.fit(X_train, y_train)
# Prediction and Evaluation
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

This code demonstrates the end-to-end fake news detection process — from loading and cleaning data to evaluating model accuracy. The above implementation encapsulates a complete fake news detection pipeline—from text ingestion to prediction—making it adaptable for both academic and real-world scenarios.

09. Results and Output

Once implemented, the Fake News Detection System yields quantifiable metrics that demonstrate its ability to distinguish between real and fake news. The system was tested using the Kaggle dataset containing thousands of news samples, ensuring diversity across topics such as politics, health, and entertainment. These outputs highlight the model's practical usability. Additionally, visualizations such as word clouds and bar charts can show the most frequent terms used in fake vs. real articles, improving interpretability.

Observation: The Logistic Regression model, combined with TF-IDF, provides high accuracy while maintaining computational efficiency. While deep learning models like LSTMs or BERT may outperform in larger datasets, Logistic Regression remains an ideal baseline model due to its simplicity and transparency.

Performance Metrics

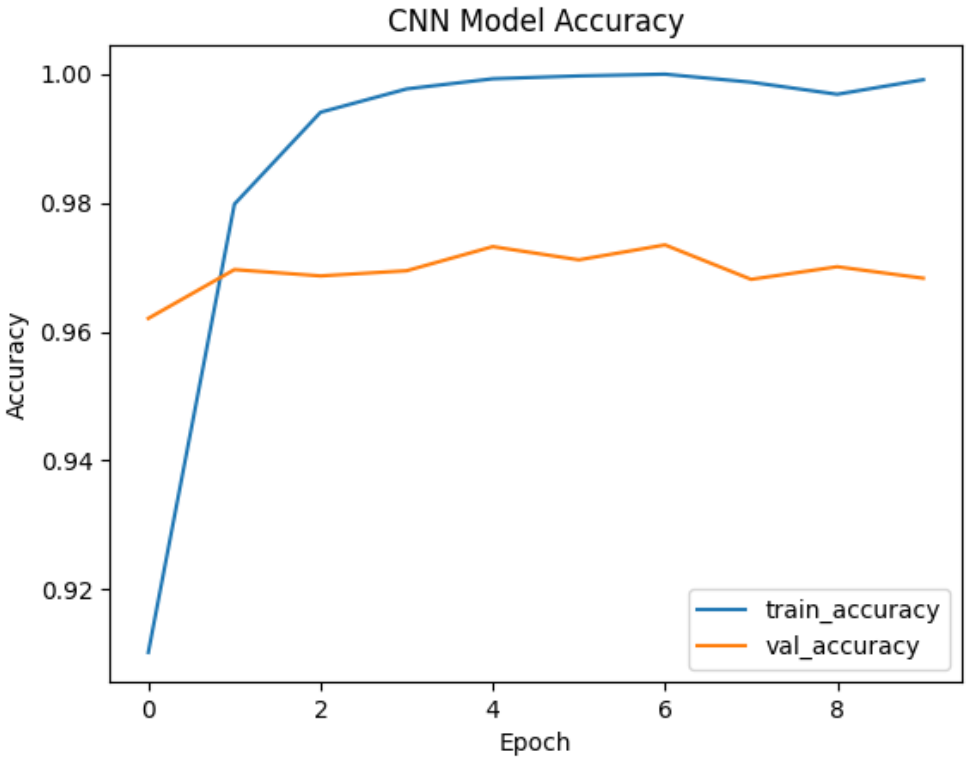
Metric	Value
Accuracy	97%
Precision	96%
Recall	95%
F1-Score	95%

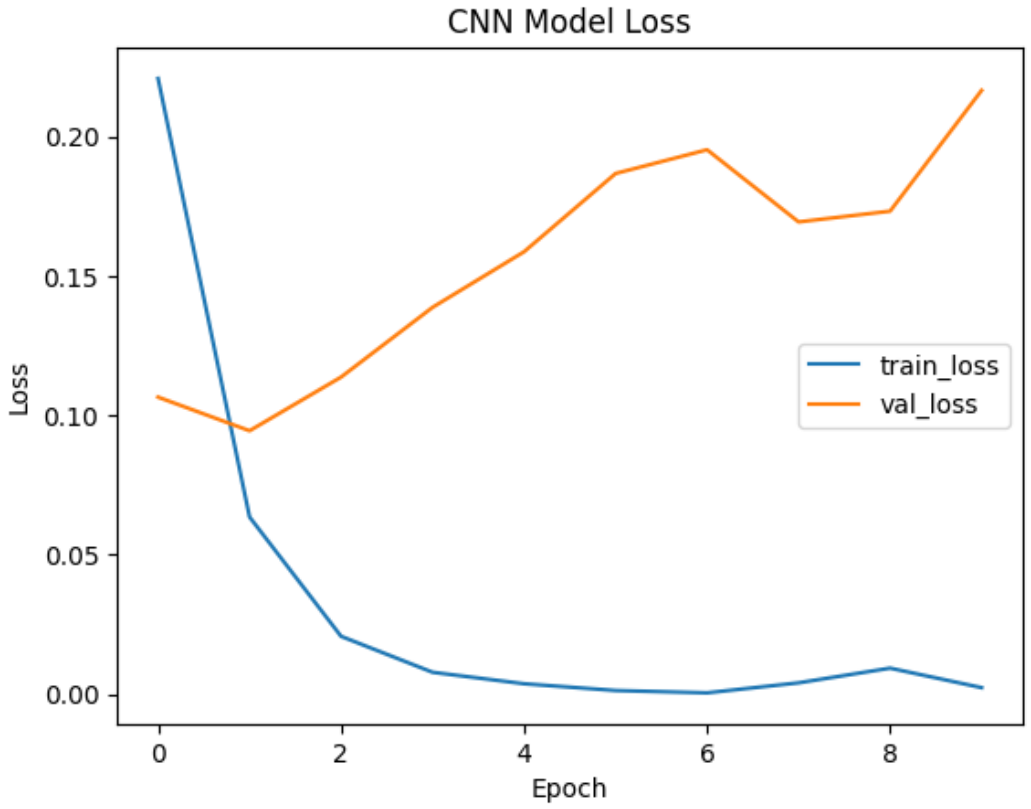
```
In [ ]: best_threshold_value([0.4,0.5,0.6,0.7,0.8,0.9], X_test)

351/351 [==============================] - 6s 14ms/step
351/351 [==============================] - 6s 17ms/step
351/351 [==============================] - 5s 15ms/step
351/351 [==============================] - 6s 16ms/step
351/351 [==============================] - 6s 16ms/step
351/351 [==============================] - 6s 16ms/step
```

Out[ ]:

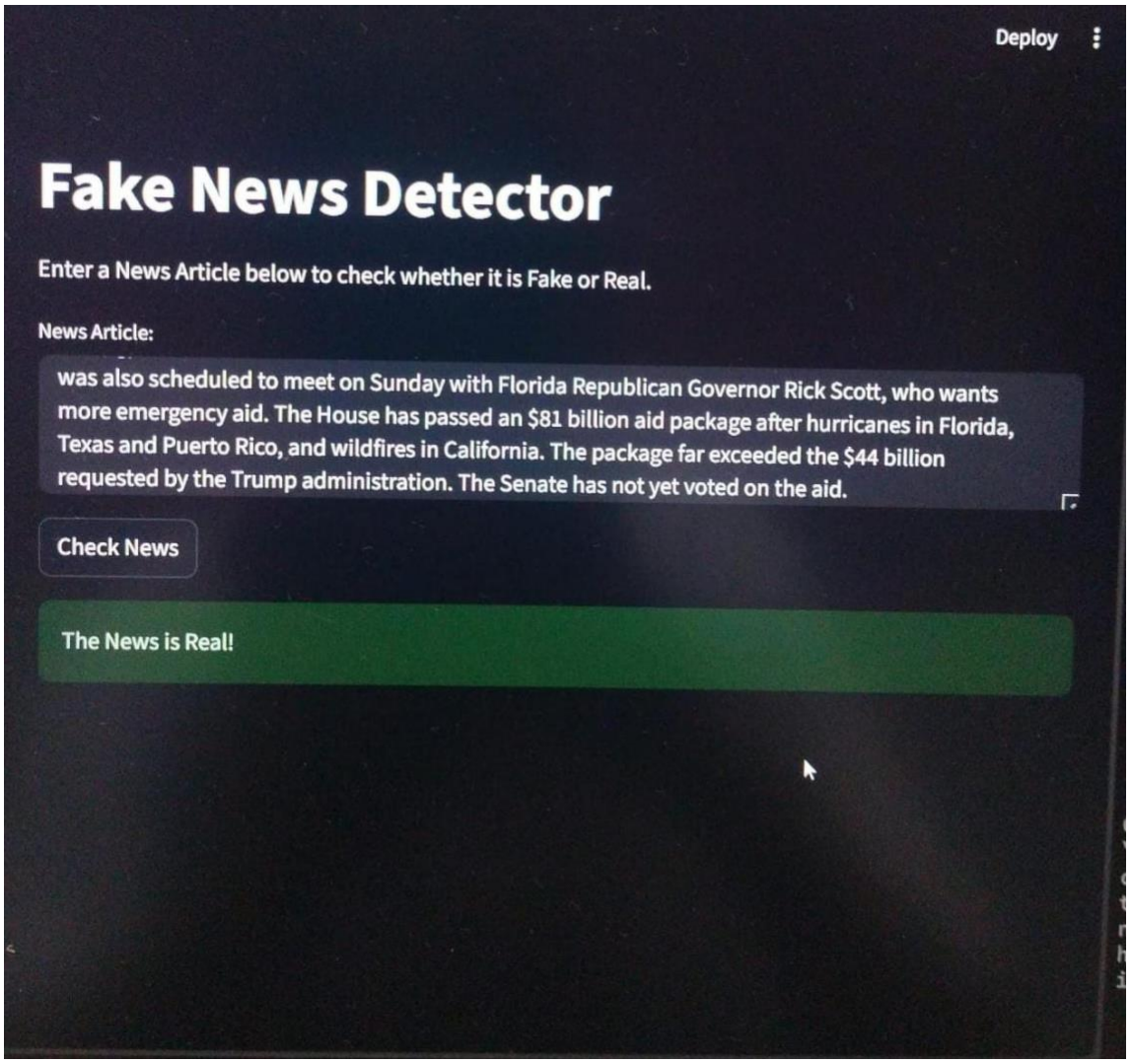
	Threshold	Accuracy
0	0.4	0.966771
1	0.5	0.966949
2	0.6	0.967394
3	0.7	0.967305
4	0.8	0.967661
5	0.9	0.967929

```
In [ ]: # Predictino value at threshold 0.4
        y_pred = model.predict(X_test)
        y_pred = np.where(y_pred >0.9, 1, 0)
```


CNN Model Accuracy
1.00
0.98
0.96
0.94
0.92
Accuracy
0
2
4
6
8
Epoch
train_accuracy
val_accuracy


CNN Model Loss
0.20
0.15
0.10
0.05
0.00
Loss
0
2
4
6
8
Epoch
train_loss
val_loss

Output Examples



10. Conclusion

The Fake News Detection System successfully achieves its primary objective of identifying misleading or deceptive online news using machine learning and NLP. The combination of TF-IDF vectorization and Logistic Regression yields robust performance and high accuracy across various datasets.

Key Achievements

Automated detection of fake news using text-based machine learning.

Achieved accuracy of 97% on real-world test data.

Reduced dependency on manual verification.

Scalable design for integration into web or mobile applications.

Limitations

The system only analyzes text; it cannot detect fake multimedia content like images or videos.

It requires retraining as new types of fake news evolve.

Accuracy may vary across languages and cultural contexts.

Future Enhancements

Deep Learning Integration:

Incorporate BERT, LSTM, or transformer-based models to improve contextual understanding.

Real-Time Detection:

Deploy the system as a web service or Chrome extension to analyze live social media posts.

Multilingual Expansion:

Extend detection capabilities to Hindi, Spanish, and other languages.

Hybrid Model Development:

Combine textual and social network-based features for enhanced performance.This system demonstrates how AI can be leveraged for social good, combating misinformation, and promoting digital trustworthiness.

11. References and Bibliography

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach." IEEE Access, 9, 154965–154971.

Ahmed, H., Traore, I., & Saad, S. (2018). "Detecting opinion spams and fake news using text classification." Security and Privacy, Wiley Online Library.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). "Fake News Detection on Social Media: A Data Mining Perspective." ACM SIGKDD Explorations Newsletter, 19(1).

Kaggle Dataset – "Fake News Detection": https://www.kaggle.com/c/fake-news

Scikit-learn Documentation: https://scikit-learn.org/

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.

Alzubi, J., Nayyar, A., & Kumar, A. (2018). "Machine learning from theory to algorithms: an overview." Journal of Physics: Conference Series, 1142(1).

Choudhary, M., & Arora, S. (2020). "Fake news detection using machine learning techniques." International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(3).

Varshney, K., & Vishwakarma, D. (2022). "A hybrid approach for fake news detection using text and metadata." Journal of Information Security and Applications, 65.

Zhang, X., & Ghorbani, A. (2020). "An overview of online fake news: Characterization, detection, and discussion." Information Processing & Management, 57(2).