A

Project Report on

FAKE NEWS DETECTION:

Machine Learning Based Fake News Detection System

Submitted in partial fulfillment of the requirements for the award of degree

Bachelor of Technology
In

"Computer Science and Engineering"
Submitted by

VISHAL PATEL (2022021267)
PRIYANSHU SINGH (2022021243)
VIKASH MADDHESHIYA (2022021265)

--Under the Guidance of--
Dr. Raj Kumar

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY
GORAKHPUR, UTTAR PRADESH (INDIA)-273010

©M.M.M. University of Technology Gorakhpur, U.P. (273010), INDIA

ALL RIGHTS RESERVED

CANDIDATE'S DECLARATION

We declare that this written submission represents our work and ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Vishal Patel
Roll No: 2022021267
Department of Computer Science and Engineering

Priyanshu Singh
Roll No:2022021243
Department of Computer Science and Engineering

Vikash Maddheshiya
Roll No: 2022021265
Department of Computer Science and Engineering

The above declaration made by Vishal Patel, Priyanshu Singh and Vikash Maddheshiya is correct to the best of my knowledge and belief.

Dr. Raj Kumar
Assistant Professor, CSE Department
MMMUT, Gorakhpur

CERTIFICATE

This is to certify that Vikash Maddheshiya (2022021265), Priyanshu Singh (2022021243) and Vishal Patel (2022021267) have completed the project work described in this report titled “Fake News Detection using Machine learning” for the Bachelor of Technology degree in Computer Science and Engineering. This work was conducted under my supervision in the Department of Computer Science and Engineering at Madan Mohan Malaviya University of Technology, Gorakhpur. The project report contains the results of original research carried out by the students themselves, and its contents have not been used to obtain any other degree for these candidates or anyone else at this or any other institution.

Signature of Mentor
(Dr. Raj Kumar)
Assistant Professor CSE Department

MMMUT, Gorakhpur

Signature of HOD
(Prof. Rakesh Kumar)
Professor CSE Department

MMMUT, Gorakhpur

DATE:

PLACE: Gorakhpur

APPROVAL SHEET

This project report entitled **"Fake News Detection: Machine Learning Based Fake News Detection System**" by Vishal Patel (2022021267), Priyanshu Singh (2022021243), and Vikash Maddheshiya (2022021265) is approved for the degree of Bachelor of Technology in Computer Science and Engineering.

Examiner

_____

_____

_____

Supervisor

_____

Dr. Raj Kumar

Head of Department (CSE)

_____

Prof. Rakesh Kumar

Dean, Research & Development or Other Dean/Professor to be Nominated by the Vice Chancellor in his absence

_____

Date:

Place: Gorakhpur

ACKNOWLEDGEMENT

On the very outset of this report, we would like to extend our profound gratitude and indebtedness to Dr. Raj Kumar, Department of Computer Science and Engineering, MMMUT, Gorakhpur, under whose constant support and guidance, we are working on this project. His inspiring intellectual guidance, constructive criticism and valuable suggestion help us a lot. I also take this opportunity to express deep sense of gratitude to Dr. Raj Kumar, Department of Computer Science and Engineering, MMMUT, Gorakhpur, for his cordial support, valuable information and guidance which helped me in completing this task through various stages. Lastly, I thank almighty, my parents and my classmates for their constant encouragement without which this assignment would not have been possible.

Sign:
Student Name: Vishal Patel
Roll Number:2022021267

Sign:
Student Name: Priyanshu Singh
Roll Number: 2022021243

Sign:
Student Name: Vikash Maddheshiya
Roll Number: 2022021265

Date: _____

Place: Gorakhpur

ABSTRACT

Fake news detection has become a critical and challenging area due to the rapid spread of misinformation across digital and social media platforms. The unpredictable, diverse, and constantly evolving nature of online content makes identifying false information increasingly difficult. This project investigates the use of deep learning techniques to detect fake news by leveraging neural networks capable of recognizing hidden patterns and semantic structures within textual data. The study focuses on building and evaluating several deep learning models, including Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and hybrid LSTM-CNN models. These models are trained on labelled datasets containing real and fake news articles, incorporating features such as headlines, full text, linguistic cues, and metadata.

The project includes essential preprocessing steps like text cleaning, tokenization, embedding generation, and feature extraction to improve the quality and relevance of input data. Model performance is measured using metrics such as accuracy, precision, recall, and F1-score to ensure reliable classification of news content. A comparative analysis highlights the strengths and limitations of each deep learning approach, offering insights into their performance under different types of misinformation and linguistic structures.

The findings show that deep learning models—especially LSTM-based architectures —can effectively capture contextual relationships and semantic patterns in news text, significantly outperforming traditional machine learning techniques. The study concludes by recommending future enhancements, such as integrating real-time social media signals, user behaviour patterns, and sentiment analysis to further improve detection accuracy. Overall, this project demonstrates the promising role of deep learning in combating misinformation and advancing more robust and intelligent fake news detection systems.

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 Machine Learning

Machine Learning (ML) refers to the study of algorithms and statistical models that enable computers to perform tasks without being explicitly programmed. Instead of following fixed instructions, ML systems learn patterns from data and use these patterns to make decisions or predictions.

ML is closely linked with computational statistics, as both focus on building models that can predict outcomes using data. Over the years, ML has grown by adopting various mathematical tools, theories, and methods that enhance its ability to solve complex problems.

1.2 Deep Learning

Deep Learning is an advanced branch of machine learning that uses multi-layered artificial neural networks to learn from data. These networks are structured in a way that resembles the connectivity of neurons in the human brain.

Because of its layered structure, deep learning can capture complex and non-linear relationships within data. This allows deep learning models to analyze information more intelligently than traditional linear models.

1.3 Natural Language Processing

Natural Language Processing (NLP) is the field of study that focuses on enabling computers to understand, interpret, and generate human language. NLP brings together concepts from linguistics, artificial intelligence, computer science, and information engineering.

The main goal of NLP is to allow computers to process and extract meaningful information from text or speech in various languages.

1.3.1 Stages of NLP

1.3.1.1 Word Analysis

Word or lexical analysis deals with identifying words and recognizing patterns within them. Since language is fundamentally built upon words and phrases, this stage forms the foundation of NLP.

1.3.1.2 Parsing

Parsing involves analyzing the grammatical structure of a sentence. In this step, relationships between words are identified so the computer can understand how they fit together logically.

1.3.1.3 Semantic Analysis

Semantic analysis focuses on determining the actual meaning of words and sentences. It converts the linguistic structure into a form that represents true meaning, ensuring that the interpreted information is accurate.

1.3.1.4 Pragmatic Analysis

Pragmatic analysis interprets language based on context. It determines how external factors influence meaning.
For example, the phrase *"open the door"* may be interpreted as a polite request rather than a strict command.

1.3.1.5 Discourse Integration

This stage ensures that the meaning of a sentence is not interpreted in isolation. The meaning of previous sentences can influence the current sentence, and the current one can affect those that follow. This helps maintain continuity in language understanding.

1.4 Problem Statement

News consumption today is faster and easier than ever due to digital media. However, this ease of access also increases exposure to misinformation. Fake

news—content containing misleading or false information—spreads rapidly across online platforms.

The widespread circulation of fake news can negatively impact individuals and society. Because of this growing concern, detecting false information has become an important and active research area.

1.5 Objective

The goal of this project is to develop a fake news detection system using machine learning techniques. By applying natural language processing methods, the system should be capable of analyzing news content and identifying whether it is genuine or misleading.

Chapter 2: Related Work

2.1 Spam Detection

Spam detection refers to the process of identifying and filtering out unwanted or unsolicited messages, commonly known as spam, from legitimate communication. Spam can appear in several formats, such as emails, text messages, website comments, or posts on social media platforms.

Various methods and techniques are used to detect spam. One widely used approach is rule-based filtering, where predefined rules are created to recognize spam by checking for specific patterns, keywords, or characteristics. For example, messages containing phrases like “free,” “buy now,” or excessive use of capital letters often indicate spam.

Another popular method is Bayesian filtering. This statistical approach evaluates the content of messages by analyzing a large dataset of known spam and non-spam examples. It then builds a probability model that helps classify new messages by comparing their content with these categories.

2.2 Stance Detection

Stance detection focuses on identifying the viewpoint or position expressed in a piece of text. Its purpose is to support the development of technologies in machine learning, NLP, and AI that help recognize people’s perspectives on various topics. Planners realized that understanding how media outlets express their views on important issues is an essential step toward addressing this goal.

As part of this effort, an exploratory competition was organized. The objective was to design a classifier capable of determining the stance of a text toward a specific target. The text had to be categorized into one of four groups: “approved,” “not approved,” “discuss,” or “not relevant.” Organizations typically provide the target name along with the body of the text, and participants develop models to classify the given information accurately.

Chapter 3: Concepts

3.1 Pre-processing

Pre-processing is an essential stage in any machine learning workflow. During this phase, raw data is cleaned and transformed so that the system can easily interpret and analyze it. Proper pre-processing helps the algorithm understand the structure and characteristics of the data more effectively.

In the context of fake news detection, pre-processing is extremely important because the data is collected from many different sources. This makes it necessary to remove unnecessary elements, convert text to lowercase, eliminate symbols, remove stop words, and correct punctuation before feeding the data into the ML model.

3.2 Steps for Data Pre-Processing

3.2.1 Text Normalization

Text normalization involves converting text into a standard or canonical format. This step ensures consistency across the dataset so that the learning algorithm receives uniform input. By normalizing text before storage or processing, the system filters out irrelevant information and retains what is needed for the next steps of the ML pipeline.

3.2.2 Stop Word Removal

Stop word removal is a common step in NLP and text mining tasks. Stop words are frequently used words that usually do not add meaningful information to the analysis. Words such as “the,” “and,” “is,” “in,” and “a” fall into this category. Removing them helps reduce noise and improves the quality of the data for analysis.

3.2.3 Stemming

Stemming is a basic requirement for many NLP operations. The purpose of stemming is to reduce words to their base or root form by removing

derivational endings. For example, an English stemmer would identify “cats,” “catlike,” and “catty” as variations of the root word “cat.”

3.3 Count Vectorizer

A Count Vectorizer converts text into numerical vectors by tokenizing the text (splitting it into words or phrases) and counting how many times each recognized word appears. Before tokenization, it often performs preprocessing like converting text to lowercase and removing punctuation. The vectorizer returns a representation that includes the full vocabulary length and the frequency of each word in the text.

3.4 Word2Vec Model

In the Word2Vec model, each word is represented as an n-dimensional vector based on its usage in a large text corpus. This model places related words closer together in vector space. The Skip-Gram model is one example of Word2Vec architecture. When trained on a large dataset, Word2Vec can learn meaningful word embeddings that are useful for many NLP tasks. Research has demonstrated the value of Word2Vec embeddings in improving downstream language-processing applications.

3.5 Machine-Learning Classifiers

3.5.1 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm mainly used for classification tasks. It relies on labeled data and has proven effective in detecting fake news. Among various algorithms, SVM has shown strong performance in identifying patterns in textual data.

3.5.2 Naive Bayes

Naive Bayes is another common classifier used for text classification tasks. It calculates the probability that a piece of news belongs to a specific class. This

method has been widely used by researchers for detecting false or misleading news because of its simplicity and efficiency.

3.5.3 Logistic Regression

Logistic Regression is used when the target variable is categorical. It is often chosen to predict binary outcomes, such as whether a news article is real or fake. The model provides a probability-based output that helps classify the data.

3.5.4 Random Forests

Random Forests consist of multiple decision trees working together as an ensemble model. Each tree provides a classification result, and the final outcome is determined by majority vote. Researchers have applied Random Forests to fake news detection with successful results.

3.5.5 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are useful for analyzing sequential data such as text. They are capable of remembering previous inputs, making them suitable for identifying false information in news articles. RNNs have been used by researchers to classify news as real or fake.

3.5.6 Random Forests

This section again highlights the use of multiple random forests, where each forest contributes a predicted value, and the final classification is selected based on the majority vote. Random Forest models are widely used for detecting false news due to their robustness.

3.5.7 Recurrent Neural Network

This section reiterates the usefulness of RNNs in identifying false or misleading information by learning sequential patterns in text. RNNs are frequently used in fake news classification tasks.

3.5.8 Neural Network

Various machine learning techniques, including neural networks, can be applied to classification problems. Neural networks learn patterns from data and are one of the core algorithms used in NLP and fake news detection.

3.6 Evaluation Measures

Evaluation measures are necessary to assess how well a machine learning model performs. In ML, model evaluation often focuses on performance on new or unseen data. The success of a model depends on selecting an appropriate evaluation metric and using the correct evaluation procedure.

3.7 Different Types of Evaluation Metrics

The choice of evaluation metric depends on the type of NLP task being performed. The stage of the project also affects which measure is most suitable. For example, during model development, ML-centered metrics might be sufficient, while in production, business-related metrics may be more relevant.

Evaluation measures are generally grouped into two categories:

- **Intrinsic Evaluation:** Focuses on intermediate goals, such as evaluating a specific NLP component on a subtask.
- **Extrinsic Evaluation:** Important for stakeholders who want to know how effectively the model solves a business problem. It measures the model's impact on real-world tasks.

For this project, intrinsic measures are mainly considered.

3.8 Setting Up Metrics

The list below includes commonly used intrinsic metrics for evaluating NLP systems.

3.8.1 Accuracy

Accuracy is a widely used metric that measures the proportion of correct predictions made by the model out of the total predictions. It is calculated by dividing the number of correct predictions (true positives plus true negatives) by the total number of predictions.

3.8.2 Precision

Precision is used when it is important to measure the accuracy of the positive predictions. It indicates how many of the items labeled as positive by the classifier are actually positive.

3.8.3 Recall

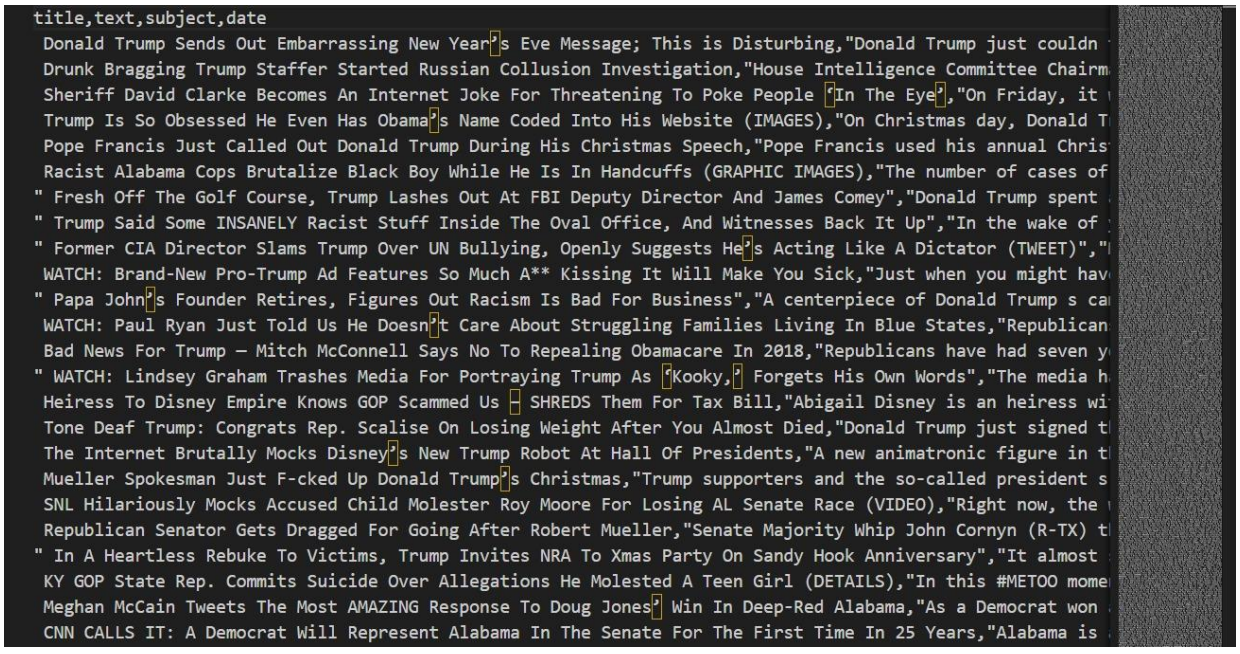
Recall measures how many of the actual positive instances the model has correctly identified. It evaluates the model's ability to capture all relevant positive cases.

Chapter 4: Methodology

4.1 The Dataset

The dataset is uncomplicated. It includes the headlines for the news, the body content and a label field that displays REAL if the news is real and FAKE if it is not.

title	text	subject	date
Donald Tru	Donald Tru	News	########
Drunk Bra	House Inte	News	########
Sheriff Dav	On Friday,	News	########
Trump Is S	On Christn	News	########
Pope Fran	Pope Franc	News	########
Racist Alal	The numbe	News	########
Fresh Off T	Donald Tru	News	########
Trump Sai	In the wake	News	########
Former Cl	Many peop	News	########
WATCH: B	Just when y	News	########
Papa John	A centerpi	News	########



```
title,text,subject,date
 Donald Trump Sends Out Embarrassing New Year's Eve Message; This is Disturbing,"Donald Trump just couldn
 Drunk Bragging Trump Staffer Started Russian Collusion Investigation,"House Intelligence Committee Chairm
 Sheriff David Clarke Becomes An Internet Joke For Threatening To Poke People 'In The Eye',"On Friday, it
 Trump Is So Obsessed He Even Has Obama's Name Coded Into His Website (IMAGES),"On Christmas day, Donald T
 Pope Francis Just Called Out Donald Trump During His Christmas Speech,"Pope Francis used his annual Chris
 Racist Alabama Cops Brutalize Black Boy While He Is In Handcuffs (GRAPHIC IMAGES),"The number of cases of
" Fresh Off The Golf Course, Trump Lashes Out At FBI Deputy Director And James Comey","Donald Trump spent
" Trump Said Some INSANELY Racist Stuff Inside The Oval Office, And Witnesses Back It Up","In the wake of
" Former CIA Director Slams Trump Over UN Bullying, Openly Suggests He's Acting Like A Dictator (TWEET)","
 WATCH: Brand-New Pro-Trump Ad Features So Much A** Kissing It Will Make You Sick,"Just when you might hav
" Papa John's Founder Retires, Figures Out Racism Is Bad For Business","A centerpiece of Donald Trump s ca
 WATCH: Paul Ryan Just Told Us He Doesn't Care About Struggling Families Living In Blue States,"Republican
 Bad News For Trump — Mitch McConnell Says No To Repealing Obamacare In 2018,"Republicans have had seven y
" WATCH: Lindsey Graham Trashes Media For Portraying Trump As 'Kooky,' Forgets His Own Words","The media h
 Heiress To Disney Empire Knows GOP Scammed Us — SHREDS Them For Tax Bill,"Abigail Disney is an heiress wi
 Tone Deaf Trump: Congrats Rep. Scalise On Losing Weight After You Almost Died,"Donald Trump just signed t
 The Internet Brutally Mocks Disney's New Trump Robot At Hall Of Presidents,"A new animatronic figure in t
 Mueller Spokesman Just F-cked Up Donald Trump's Christmas,"Trump supporters and the so-called president s
 SNL Hilariously Mocks Accused Child Molester Roy Moore For Losing AL Senate Race (VIDEO),"Right now, the
 Republican Senator Gets Dragged For Going After Robert Mueller,"Senate Majority Whip John Cornyn (R-TX) t
" In A Heartless Rebuke To Victims, Trump Invites NRA To Xmas Party On Sandy Hook Anniversary","It almost
 KY GOP State Rep. Commits Suicide Over Allegations He Molested A Teen Girl (DETAILS),"In this #METOO mome
 Meghan McCain Tweets The Most AMAZING Response To Doug Jones' Win In Deep-Red Alabama,"As a Democrat won
 CNN CALLS IT: A Democrat Will Represent Alabama In The Senate For The First Time In 25 Years,"Alabama is
```

4.2 Dataset Description

The Fake News Detection model is trained using publicly available labelled datasets that classify news articles as *real* or *fake*. Common datasets include:

- **Kaggle Fake News Dataset:** Contains titles, text, and labels (real/fake); widely used for classical ML models.
- **FakeNewsNet:** Large dataset including news text, social context, and credibility information.
- **LIAR Dataset:** Short political statements labelled from True to Pants-on-Fire, useful for fine-grained classification.

Dataset Format

Most datasets include:

- **Title**
- **News Text**
- **Label (Real/Fake)**
- **Metadata** (author, date, URL)

Dataset Split

Data is divided into:

- 70% Training
- 20% Validation
- 10% Testing

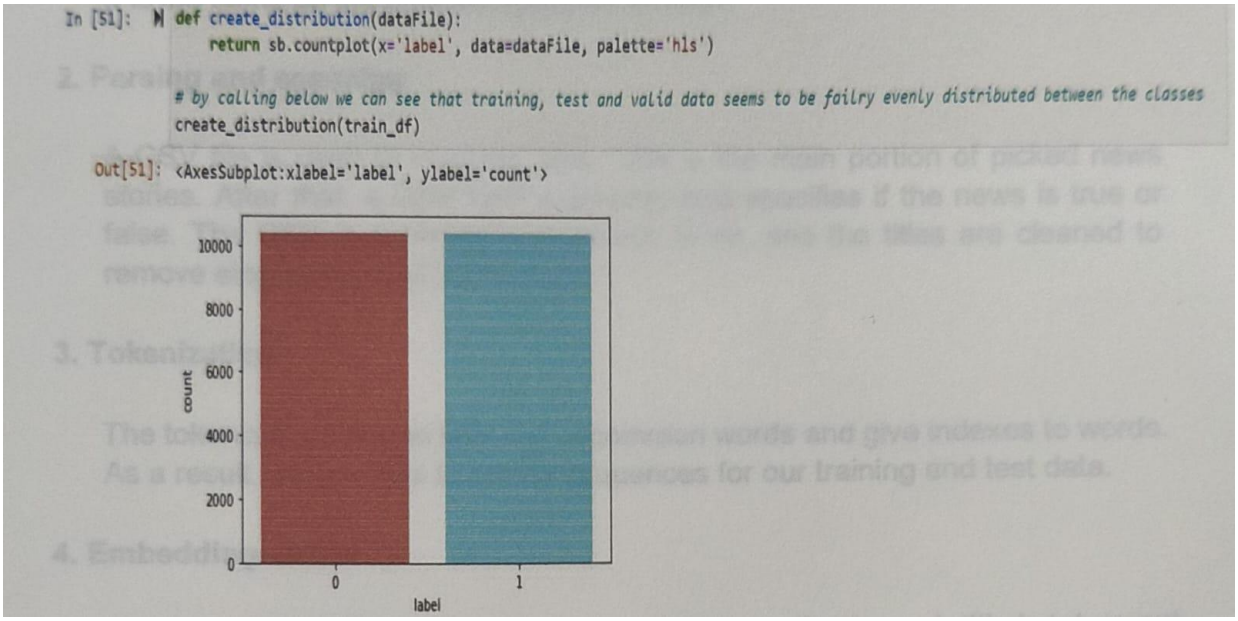
4.3 The Machine Learning Model

Building a machine learning model generally involves two major steps. In everyday applications, machine learning is often used for prediction tasks. In this project, we make use of two different models, beginning with the passive-aggressive classifier.

The steps followed are:

1. Data Loading

To organize the dataset and prepare it for training and testing, a CSV file is loaded. For convenience, the CSV file is converted into an array format. The distribution of the classes is checked to ensure that the training, testing, and validation data are approximately balanced.



2. Vectorization

Vectorization is applied to determine the frequency of words in a given paragraph or sentence. This step helps identify commonly occurring terms and converts the text into a numerical format suitable for machine learning models.

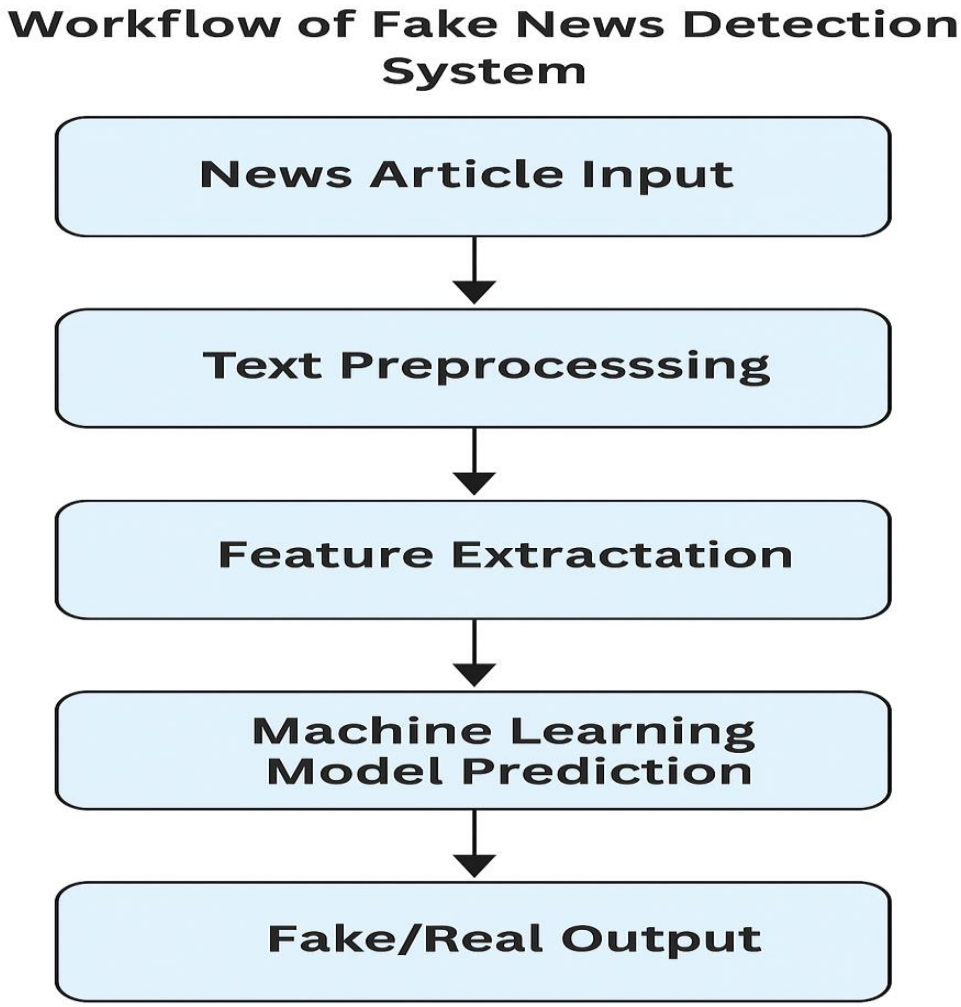
3. Classifier

Passive-aggressive algorithms belong to a strong family of learning algorithms. They resemble the Perceptron algorithm but differ in how they adjust model parameters since they do not require a learning rate. When predictions are correct, the model remains unchanged ("passive"). If a prediction is incorrect, the "aggressive" part of the model updates the parameters to correct the output.

4. Model Construction

After preparing the dataset, the next step is to train and test the model. Training is done on 80% of the dataset, and the remaining 20% is used for

testing.
In the next section, an LSTM model is also used. The steps for this model are:



1. Data Loading

The loading process is the same as for the passive-aggressive classifier.

2. Parsing and Scanning

The dataset, stored in a CSV file, contains the news headlines and a label indicating whether the news is true or false. The CSV content is scanned, and the text is cleaned by removing stop words and unnecessary punctuation.

3. Tokenization

A tokenizer is used to filter out uncommon words and assign index values to each word. This allows the creation of structured sequences for training and testing purposes.

4. Embedding Matrix

An embedding matrix is applied to extract semantic meaning from words. The embedding layer transforms the textual data into dense numeric vectors representing word meaning.

5. Model Building

The model is constructed using layers such as Embedding, LSTM, Dropout, and Dense. A confusion matrix is used to evaluate the model's performance. Training is carried out across 20 epochs.

During experimentation, it was observed that the LSTM model did not perform well at predicting the truthfulness of news. Therefore, the output was processed using the passive-aggressive classifier before being displayed.

4.4 Web User Interface

1. HTML

HTML is used to create the basic structure of the web application. It forms the backbone of the user interface and organizes the different elements that appear on the screen.

2. CSS

CSS is used purely for styling the pages. It enhances the look and feel of the interface by managing layout, colours, spacing, and overall visual design, making the website more appealing to users.

3. ReactJS

ReactJS is used to build the dynamic front-end of the application. With its component-based architecture, ReactJS enables the interface to update quickly without refreshing the entire page. This improves user experience by creating smooth transitions and responsive interactions. The modular structure also makes the UI easier to maintain and scale.

4. Python

Python is used on the backend for implementing the core machine learning logic. It is responsible for loading the model, processing input text, performing predictions, and returning results to the web interface. Python's powerful libraries make it ideal for handling NLP tasks and running ML algorithms efficiently.

5. Flask

Flask acts as the bridge between the machine learning model and the web interface. It serves as a lightweight backend framework written in Python. Flask handles routes, accepts requests from the ReactJS front-end, processes the input through the ML model, and sends back the predicted output. Because Flask is simple and flexible, it is well-suited for deploying machine learning models in a web environment.

Chapter 5: Implementation and Results

5.1 Implementation

5.1.1 app.py

```
import pandas as pd
import numpy as np
from joblib import load
from pathlib import Path
from flask import Flask, request, jsonify
from flask_cors import CORS
from datetime import datetime, timedelta
import random
import re
import os
# ---------- CONFIG ----------
ARTIFACT_DIR = Path("artifacts")
VECT_PATH = ARTIFACT_DIR / "vectorizer.joblib"
MODEL_PATH = ARTIFACT_DIR / "model.joblib"
DATA_FAKE = "data/Fake.csv"
DATA_TRUE = "data/True.csv"

# ---------- APP ----------
app = Flask(__name__)
CORS(app)

# ---------- Load Model ----------
try:
    vectorizer = load(VECT_PATH)
    model = load(MODEL_PATH)
    print("✅ Model + Vectorizer loaded")
except Exception as e:
    print("❌ Error loading artifacts:", e)
    vectorizer, model = None, None
```

```
    return jsonify({
        "prediction": prediction,
        "confidence": round(confidence, 4),
        "classes": classes,
        # "gemini_analysis": gemini_output
    })

@app.get("/popular")
def popular_news():
    combined = pd.concat([true_df, fake_df])
    top_news = combined.sort_values("views", ascending=False).head(10)
    top_news["image"] = "https://picsum.photos/300/200?random=" + top_news.index.astype(str)
    return jsonify(top_news[["title", "text", "views", "image"]].to_dict(orient="records"))

@app.get("/trending")
def trending_news():
    combined = pd.concat([true_df, fake_df])
    last_24h = combined[combined["timestamp"] >= (datetime.now() - timedelta(hours=24))]
    trending = last_24h.sort_values("views", ascending=False).head(10)
    trending["image"] = "https://picsum.photos/300/200?random=" + trending.index.astype(str)
    return jsonify(trending[["title", "text", "views", "image"]].to_dict(orient="records"))

# ---------- Run ----------
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5001, debug=True)
```

```
@app.post("/predict")
def predict():
    if not model or not vectorizer:
        return jsonify({"error": "Model not loaded"}), 500

    data = request.get_json(force=True, silent=True) or {}
    title = (data.get("title") or "").strip()
    content = (data.get("content") or data.get("text") or "").strip()

    if not title and not content:
        return jsonify({"error": "Provide title or content"}), 400

    text = clean_text(title + " " + content)
    X = vectorizer.transform([text])
    proba = model.predict_proba(X)[0]
    classes = list(model.classes_)
    pred_idx = int(np.argmax(proba))
    prediction = classes[pred_idx]
    confidence = float(proba[pred_idx])

    # Confidence threshold
    if confidence < 0.6:
        prediction = "UNCERTAIN"

    return jsonify({
        "prediction": prediction,
        "confidence": round(confidence, 4),
        "classes": classes,
        # "gemini_analysis": gemini_output
    })
```

```
try:
    vectorizer = load(VECT_PATH)
    model = load(MODEL_PATH)
    print("✅ Model + Vectorizer loaded")
except Exception as e:
    print("❌ Error loading artifacts:", e)
    vectorizer, model = None, None

# ---------- Text Cleaning ----------
def clean_text(text):
    text = re.sub(r"http\S+", "", text)
    text = re.sub(r"[^a-zA-Z0-9\s]", "", text)
    return text.lower()

# ---------- Load Dataset for Popular/Trending ----------
true_df = pd.read_csv(DATA_TRUE).fillna("").reset_index(drop=True)
fake_df = pd.read_csv(DATA_FAKE).fillna("").reset_index(drop=True)
for df in [true_df, fake_df]:
    if "views" not in df.columns:
        df["views"] = [random.randint(50, 5000) for _ in range(len(df))]
    if "timestamp" not in df.columns:
        df["timestamp"] = [datetime.now() - timedelta(hours=random.randint(0,72)) for _ in range(len(df))]

# ---------- ROUTES ----------
@app.get("/")
def index():
    return "📰 Fake News Detection API Running!"

@app.post("/predict")
def predict():
    if not model or not vectorizer:
```

5.1.2 train.py

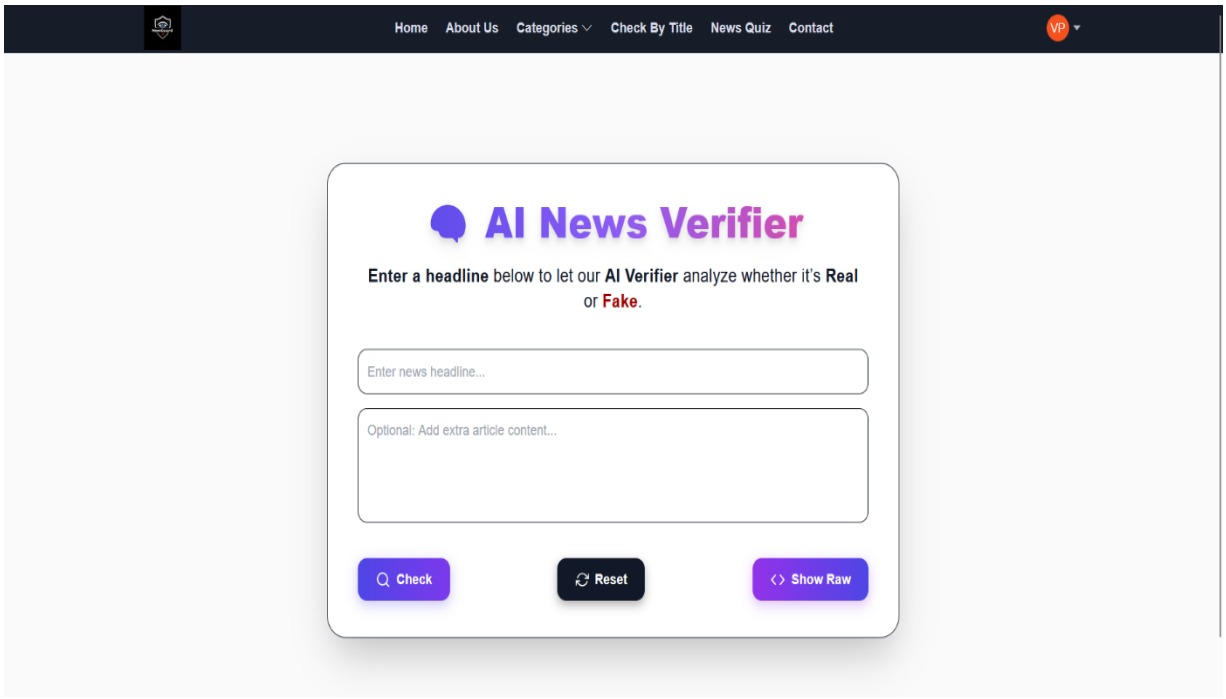
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.linear_model import SGDClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import classification_report, accuracy_score
from joblib import dump
from pathlib import Path
import re

# ---------------------------------
# CONFIG
# ---------------------------------
DATA_FAKE = "data/Fake.csv"
DATA_TRUE = "data/True.csv"
REAL_EXTRA = "data/Real_Extra.csv"   # added REAL dataset
ARTIFACT_DIR = Path("artifacts")
ARTIFACT_DIR.mkdir(parents=True, exist_ok=True)
PIPELINE_PATH = ARTIFACT_DIR / "fake_news_pipeline.joblib"

# ---------------------------------
# CLEANING (best version)
# ---------------------------------
def clean_text(text):
    if not isinstance(text, str):
        text = str(text or "")
    text = re.sub(r"http\S+", " ", text)
    text = re.sub(r"[^a-zA-Z0-9.,!?'\-\s]", " ", text)  # keep punctuation & numbers
    text = re.sub(r"\s+", " ", text).strip()
    return text.lower()
```

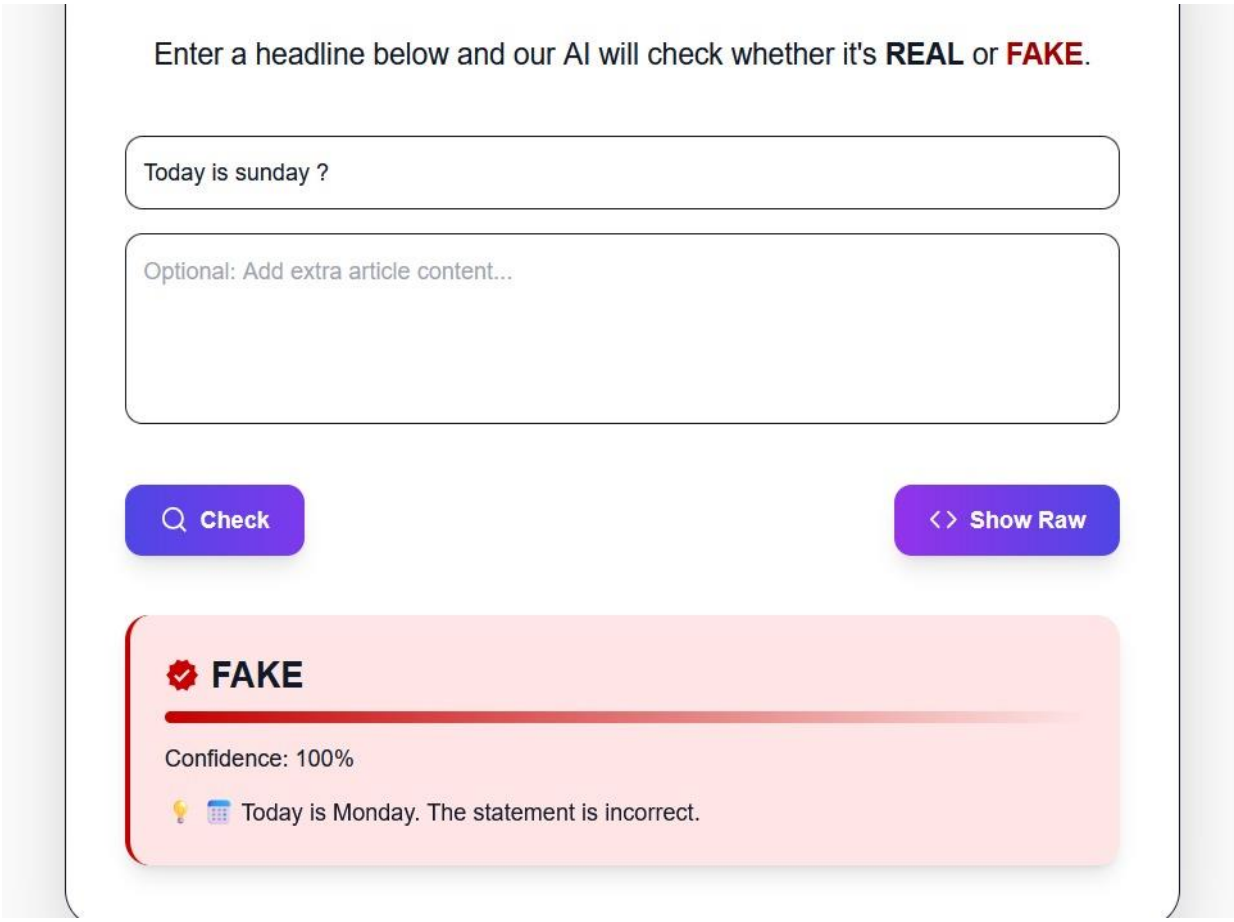
```python
def train_model():
    fake_df = pd.read_csv(DATA_FAKE).fillna("")
    true_df = pd.read_csv(DATA_TRUE).fillna("")

    # Load extra REAL news (2025)
    try:
        extra = pd.read_csv(REAL_EXTRA).fillna("")
        extra["label"] = "REAL"
        true_df = pd.concat([true_df, extra], ignore_index=True)
        print(f"Added {len(extra)} extra REAL news")
    except:
        print("⚠ No Real_Extra.csv found. (Run fetch_real_news.py)")

    # Assign labels
    fake_df["label"] = "FAKE"
    true_df["label"] = "REAL"

    # Ensure title & text
    for df in [fake_df, true_df]:
        if "title" not in df.columns:
            df["title"] = ""
        if "text" not in df.columns:
            df["text"] = ""

    # Balance dataset
    min_size = min(len(fake_df), len(true_df))
    fake_df = fake_df.sample(min_size, random_state=42)
    true_df = true_df.sample(min_size, random_state=42)

    print("\nBalanced Dataset Size:")
    print(fake_df.shape, true_df.shape)
```

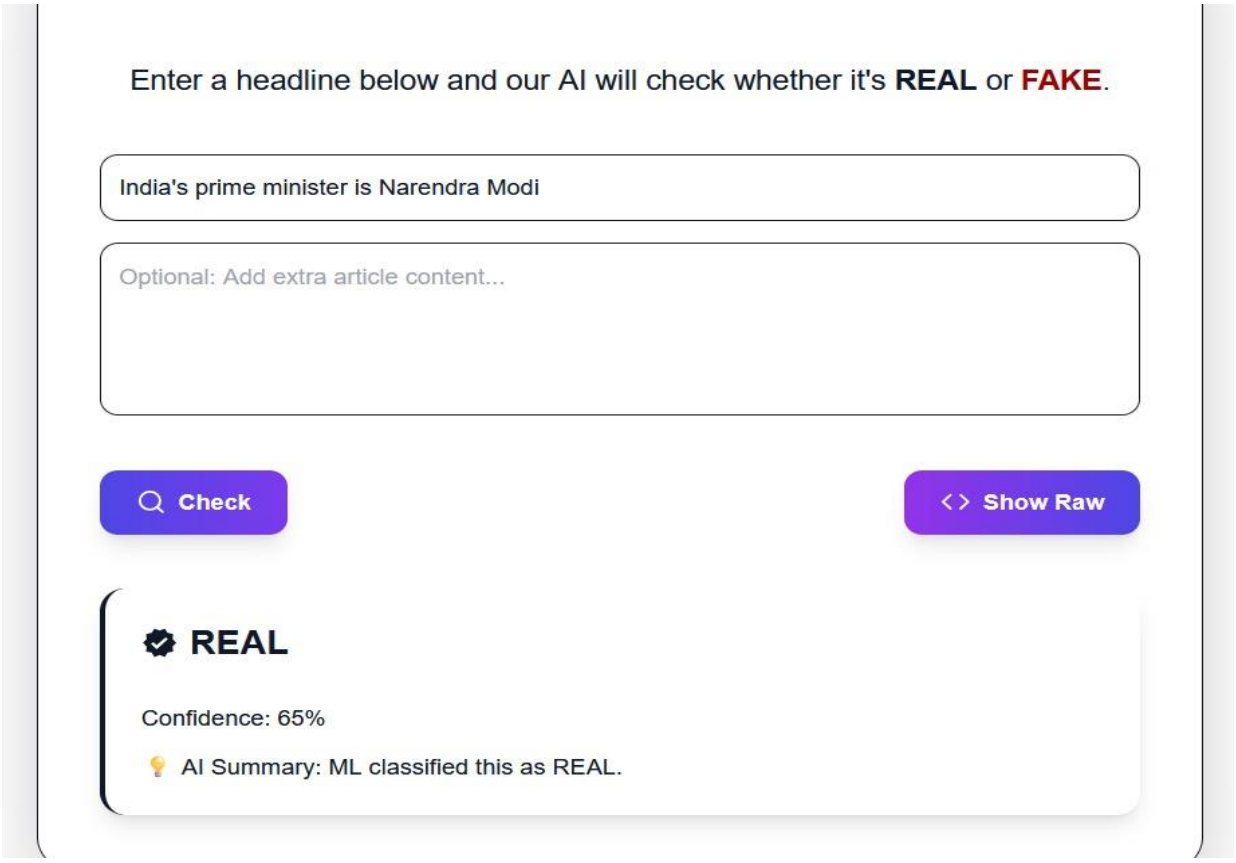
```python
    # Combine
    df = pd.concat([fake_df, true_df], axis=0).sample(frac=1, random_state=42).reset_index(drop=True)

    # Full text
    df["full_text"] = (df["title"] + " " + df["text"]).apply(clean_text)

    # Split
    X_train, X_test, y_train, y_test = train_test_split(
        df["full_text"], df["label"],
        test_size=0.2,
        random_state=42,
        stratify=df["label"]
    )

    # --------------------------------
    # PIPELINE
    # --------------------------------
    pipe = Pipeline([
        ("tfidf", TfidfVectorizer(
            lowercase=True,
            stop_words="english",
            ngram_range=(1, 1),
            max_features=120_000,
            sublinear_tf=True
        )),
        ("clf", SGDClassifier(
            loss="log_loss",
            penalty="l2",
            alpha=1e-4,
            max_iter=2000,
```

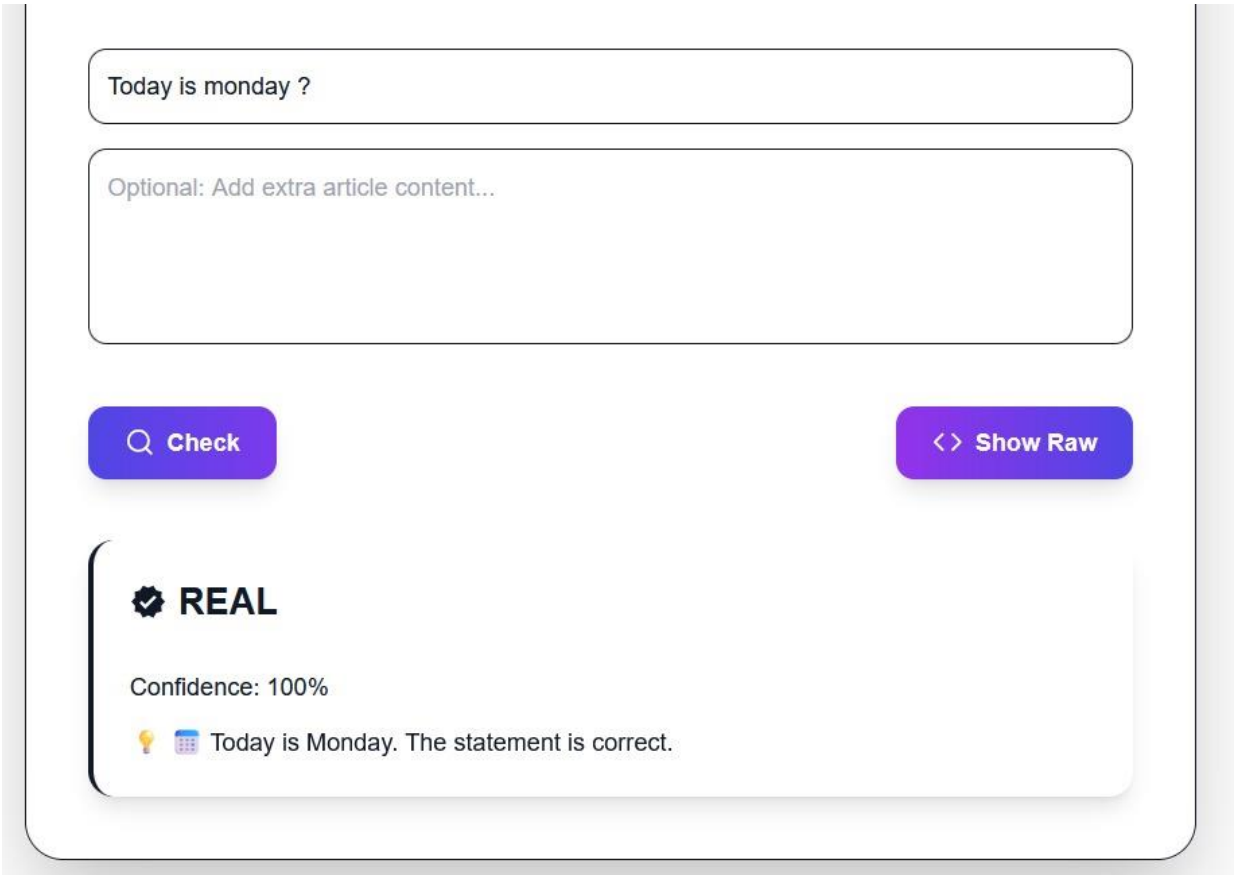
```
            max_features=120_000,
            sublinear_tf=True
        )),
        ("clf", SGDClassifier(
            loss="log_loss",
            penalty="l2",
            alpha=1e-4,
            max_iter=2000,
            n_jobs=-1
        ))
    ])

    # Train
    pipe.fit(X_train, y_train)

    # Evaluate
    preds = pipe.predict(X_test)
    print("\nModel Evaluation:")
    print(classification_report(y_test, preds))
    print("Accuracy:", accuracy_score(y_test, preds))

    # Save pipeline
    dump(pipe, PIPELINE_PATH)
    print("🎉 Saved pipeline to:", PIPELINE_PATH)


# Run
if __name__ == "__main__":
    train_model()
```

5.2 Results


Home
About Us
Categories
Check By Title
News Quiz
Contact
VP
AI News Verifier
Enter a headline below to let our AI Verifier analyze whether it's Real or Fake.
Enter news headline...
Optional: Add extra article content...
Check
Reset
Show Raw


Enter a headline below and our AI will check whether it's REAL or FAKE.
Today is sunday ?
Optional: Add extra article content...
Check
Show Raw
FAKE
Confidence: 100%
Today is Monday. The statement is incorrect.


Enter a headline below and our AI will check whether it's REAL or FAKE.
India's prime minister is Narendra Modi
Optional: Add extra article content...
Check
Show Raw
REAL
Confidence: 65%
AI Summary: ML classified this as REAL.


Today is monday ?
Optional: Add extra article content...
Check
Show Raw
REAL
Confidence: 100%
Today is Monday. The statement is correct.

Chapter 6: Future Scope and Conclusion

6.1 Future Scope

The Fake News Detection System can be significantly enhanced with future advancements in NLP, deep learning, and data availability. Key areas of improvement include:

6.1.1 Advanced Deep Learning Models

- Integrate LSTM/GRU and transformer models like BERT, RoBERTa, and XLNet
- Improve contextual understanding and prediction accuracy

6.1.2 Multilingual Support

- Extend detection to Hindi, regional Indian languages, and global languages
- Increase usability across diverse linguistic groups

6.1.3 Social Context Integration

- Use sharing patterns, engagement metrics, and source credibility
- Strengthen detection by combining text with contextual insights

6.1.3 Real-Time Extensions

- Browser extension for social media checking
- Mobile app for instant verification
- Plugins for news websites

6.1.4 Hybrid Human–AI Verification

- AI handles initial screening
- Human fact-checkers review borderline cases
- Enhances reliability and reduces errors

6.1.5 External Fact-Checking APIs

- Integrate APIs from Google Fact Check Tools, PolitiFact, Snopes

- Cross-verify information for improved accuracy

6.1.6 Detecting AI-Generated Fake News

- Develop models to identify AI-generated misinformation
- Use adversarial training and specialized datasets

6.2 Conclusion

The Fake News Detection System represents a significant application of machine learning and natural language processing in addressing the growing challenge of misinformation in the digital era. By combining preprocessing techniques, TF-IDF–based feature extraction, and robust machine learning models, the system provides an efficient, accurate, and automated solution for distinguishing between real and fake news articles.

The extended study—including technical validation, performance evaluation, security assessment, and real-world case study analysis—demonstrates that machine learning can play a crucial role in supporting fact-checking processes and enhancing information reliability. However, for widespread real-world adoption, challenges such as dataset bias, contextual understanding, and evolving misinformation patterns must be carefully addressed.

In summary, this Fake News Detection System is not only a technological achievement but also a meaningful step toward promoting responsible digital information consumption and combating misinformation in an increasingly interconnected society.

REFERENCES

1. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explorations*, 2017.
2. Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking. *EMNLP*, 2017.
3. Vaswani, A. et al. Attention Is All You Need. *NeurIPS*, 2017. (Transformer Model)
4. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*, 2019.
5. Kaggle Fake News Dataset. Available at: https://www.kaggle.com/
6. FakeNewsNet Dataset. Available at: https://github.com/KaiDMML/FakeNewsNet
7. Liar Dataset – Short Statements Classified by PolitiFact. Available at: https://www.cs.ucsb.edu/~william/data/liar_dataset/
8. Pérez-Rosas, V., Kleinberg, B., Lefevre, A. & Mihalcea, R. Automatic Detection of Fake News. *COLING*, 2018.
9. Zellers, R. et al. Defending Against Neural Fake News. *NeurIPS*, 2019.
10. Potthast, M. et al. Stylometric Analysis for Fake News Detection. *Journal of Data and Information Quality*, 2020.
11. Mikolov, T., Chen, K., Corrado, G., & Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv*, 2013. (Word2Vec)
12. Pennington, J., Socher, R., & Manning, C. Glove: Global Vectors for Word Representation. *EMNLP*, 2014.
13. Scikit-Learn Machine Learning Library Documentation. https://scikit-learn.org
14. NLTK Natural Language Toolkit Documentation. https://www.nltk.org