Phase-1

Student Name: Pargunan

Register Number: 410723104058

Institution: Dhanalakshmi College Of Engineering

Department : Computer Science Engineering

Date of Submission: 27-04-2025

1.Problem Statement

In today's digital era, the rapid spread of misinformation and fake news across online platforms has become a critical global issue. False information not only misleads the public but can also incite fear, manipulate opinions, destabilize societies, influence important decisions like elections, public health measures.

Despite efforts to manually fact-check content, the sheer volume of news being produced every second makes it impossible to rely solely on human efforts. Current detection systems often struggle with speed, accuracy, or bias, leading to false positives and missed threats.

*Thus, there is an urgent need for an **automated, intelligent system powered by advanced Natural Language Processing (NLP)** that can accurately detect fake news in real-time, explain its decisions transparently, and help promote a more informed, truthful digital society.*

2.Objectives of the Project

***Develop an automated fake news detection system** using advanced Natural Language Processing (NLP) techniques.*

1. ***Accurately classify news articles*** *as "Real" or "Fake" based on their content, writing style, and linguistic features.*

2. ***Achieve high performance*** *in terms of accuracy, precision, recall, and F1-score through optimized machine learning or deep learning models.*

3. ***Implement explainable AI (XAI) methods*** *to provide understandable reasons behind the system's classification decisions (e.g., highlighting suspicious keywords or phrases).*

4. ***Create a user-friendly interface*** *where users can input news content and immediately receive authenticity feedback.*

5. ***Generate insights*** *into common linguistic patterns, emotional tones, or biases often present in fake news content.*

6. ***Promote digital literacy*** *by helping users recognize traits of misinformation on their own through transparency and education.*

3.Scope of the Project

- ***Analyze*** *news articles using NLP techniques like TF-IDF, Word2Vec, and BERT embeddings.*

- ***Build*** *machine learning and deep learning models (e.g., Logistic Regression, LSTM, BERT) to classify news as real or fake.*

- ***Implement*** *explainable AI tools (like LIME or SHAP) to justify predictions.*

- ***Develop*** *a basic web app prototype for user interaction (if time permits).*

Limitations:

- *Focus only on* ***English-language*** *articles.*

- ***Dependence*** *on available public datasets .*

- ***Limited*** *by hardware resources for training large models.*

- ***Prototype-level*** *deployment only; no large-scale real-time system.*

4.Data Sources

The project will use publicly available datasets from ***Kaggle*** *and other sources:*

- ***Fake News Detection Dataset (Kaggle):***
  Link: https://www.kaggle.com/c/fake-news/data

- ***LIAR Dataset (Fake news dataset from Politifact):***
  Link: https://www.cs.ucsb.edu/~william/data/liar_dataset.zip

- *(Optional)* ***FakeNewsNet Dataset:***
  Link: https://github.com/KaiDMML/FakeNewsNet

- *These datasets are* ***public*** *and* ***freely accessible*** *for academic and research purposes.The datasets are* ***static*** *— they are downloaded once and not updated in real-time.*

- *If needed, additional synthetic samples may be generated for data balancing (e.g., via text augmentation techniques).*

5.High – Level Methodology

- ***Data Collection:***
  Data will be obtained by ***downloading*** *public datasets from* ***Kaggle*** *and* ***other open repositories****. No web scraping or API access is planned initially. Minor synthetic data generation may be done to balance classes if needed.*

- ***Data Cleaning:***
  Handle ***missing values*** *by either removing or imputing them,* ***remove duplicates*** *to avoid bias, and* ***standardize text*** *(lowercasing, punctuation removal, stemming/lemmatization) to ensure uniformity.*

- ***Exploratory Data Analysis (EDA):***
  Use ***word clouds****,* ***frequency plots****,* ***bar charts****, and* ***correlation heatmaps*** *to discover common words, patterns in writing styles, sentiment trends, and class imbalances in the dataset.*

- ***Feature Engineering:***
  Extract features using ***TF-IDF****,* ***Bag of Words (BoW)****, and* ***word embeddings*** *like* ***Word2Vec*** *and* ***BERT****. Additional engineered features like* ***sentiment scores*** *or* ***readability scores*** *may also be included to improve model performance.*

- ***Model Building:***
  Experiment with traditional models like **Logistic Regression***,* **Random Forest***,* **Support Vector Machine (SVM)***, and deep learning models like* **LSTM** *and* **BERT fine-tuning***. These models are well-suited for text classification tasks because they handle sequence patterns.*

- ***Model Evaluation:***
  Use metrics such as **Accuracy***,* **Precision***,* **Recall***,* **F1-Score***, and* **ROC-AUC** *to evaluate performance. Apply* **k-fold cross-validation** *to ensure model generalization and avoid overfitting.*

- ***Visualization & Interpretation:***
  Present key insights using **matplotlib***,* **seaborn***, and* **plotly** *for interactive charts. Also, use* **LIME** *or* **SHAP** *to visually explain model predictions and highlight important words influencing classification.*

- ***Deployment:***
  If time allows, deploy the model as a simple **web application** *using* **Flask** *or* **Streamlit** *where users can paste a news article and see whether it is classified as "Real" or "Fake."*

6.Tools and Technologies

Programming Language:

  - ***Python*** *will be the primary language used for all stages of the project.*

- ***Notebook/IDE:***

  - *Development will be carried out mainly in* **Google Colab** *for easy access to GPU resources, and* **Jupyter Notebook** *for local experimentation.*

- ***Libraries:***

  - ***Data Processing:*** *pandas, numpy*

  - ***Data Visualization:*** *matplotlib, seaborn, plotly*

  - ***Modeling and Machine Learning:*** *scikit-learn, TensorFlow, Keras,* *PyTorch (for BERT models), transformers (Hugging Face)*

- ***NLP Processing:*** *nltk, spaCy*
- ***Explainability:*** *LIME, SHAP*

- ***Optional Tools for Deployment:***
  - ***Streamlit*** *or* ***Flask*** *will be used to deploy a simple web application if a front-end interface is developed.*

7.Team Members and Roles

Member	Role Title	Responsibilities
Vaidhyanathan	Data Collection & Preprocessing Lead	- Collect datasets from public sources (e.g., Kaggle). - Clean and preprocess the data (handle missing values, remove duplicates, normalize text).
Pargunan	EDA & Feature Engineering Specialist	- Perform Exploratory Data Analysis (EDA) to uncover trends and patterns. - Create and implement feature extraction methods (TF-IDF, Word2Vec, BERT).
Tharunkumar	Model Building & Evaluation Expert	- Build and fine-tune machine learning and deep learning models. - Evaluate model performance using accuracy, precision, recall, F1-score.
Vijay	Visualization & Deployment Manager	- Create visualizations (matplotlib, seaborn, plotly) to present findings. - Deploy the model using a simple web app (Streamlit or Flask).