## FakeGuard :An AI-Powered Fake News Detection System

***A project report submitted to SWECHA***

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

**BACHELOR OF TECHNOLGY**

**in**

**COMPUTER SCIENCE & ENGINEERING (AI & ML)**

**Submitted by**

**S.Bhavaja**

**P.Bhavani**

**P.Sri Sathwik**

**V.Meghana**

**S.M.Ram Prakash**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AI & ML)**

Abstract

The proliferation of fake news threatens public awareness, societal calm, and democracy more than ever, especially considering the ever-evolving digital world. Given the millions who consume unverified news content every day, there is a special need for a reliable and verifiable news-checking service for low-resource and multiple languages regions.

FakeGuard is an application aimed to counter the spread of fake news and is powered by artificial intelligence. It leverages Natural Language Processing (NLP) and machine learning to classify news. It is built on Streamlit and uses lightweight scikit-learn models. The system analyzes text submitted by users and predicts if the news is real or fake. FakeGuard is also optimized for offline use and accepts English, Hindi, and Telugu as inputs, thus serving users from different Indian regions.

FakeGuard is different from traditional fact-checking solutions as it uses prediction models to classify news without external queries to the internet or APIs. It also functions as a corpus collection engine where prediction inputs and user inputs with their consent are captured. This enables the creation of a corpus for real and fake news in regional languages that can be used to train more robust culturally aware AI models in the future.

Collecting region-specific low-resource languages news corpus shows the potential that AI has to help counter misinformation, which our team demonstrates through this project.

Introduction

**Problem Definition**

The rapid development of social media has massively transformed how news is consumed, created, and disseminated. On the downside, however, the surge in news content has also on a quick rise to the dissemination of fake news, content that is purposefully and completely fabricated and disguised as legitimate news. Misinformation of such news content not only has the potential to skew public perception but also create chaos and amazing scepticism regarding the legitimate news sources.

The current methods in place for fake news detection still seem to focus on manual fact verification. This is a painstaking and time-intensive process that is severely restricted by language, as well as, most regional and rural users. Many AI detection systems depend on the internet and require cloud APIs. This is a huge problem for millions of users in India who consume and share content in regional languages.

The absence of structured real-world datasets from non-mainstream languages makes building AI informal and regional text pattern comprehension models extremely difficult. This creates not only a gap in detection tools but also a lack of corpus for training future models in NLP.

The problem needs to be solved from the ground level, creating a flexible and AI supported.

**Objective of Project**

FakeGuard’s main goal is to create a lightweight application that functions offline, utilizing NLP to detect and classify fake news in multiple languages, working even in areas with low connectivity.

This project is designed to accomplish the following goals:

Assist Users in Combating Misinformation: Help mitigate the proliferation of conflicting narratives by empowering users with tools that provide news authenticity checks in real-time, especially on social media networks that support swift news sharing.

Enhance Access to Non-English Speakers: Support the Indian regional languages of Hindi and Telugu in addition to English, expanding the scope of the application to non-English speaking populations.

Aid Regions with Low Connectivity: Work offline first with capabilities to operate in low-bandwidth areas where fact-checking tools and internet connectivity is scarce.

Aid in Corpus Collection: Document news snippets and their predicted labels, with users’ approval, to enrich the corpus for robust multilingual AI model training.

Enhance Integration of AI: Illustrate the use of TF-IDF and Logistic Regression AI models in real-life scenarios through the use of Streamlit.

Encourage Community Participation: Create a platform that SWECHA and viswam.ai wish to fulfill with their collaborative open-source aim, especially for the educational sector.

**Scope of the Project**

The coverage of the FakeGuard project dives into the technology, language and society integration. It aims to serve beyond an “anti-fake-news” tool to serve also as an NLP application that automates the creation of corpuses as well as AI implementation in challenging environments.

The main sections of the project are as follows:

Fake News Classification

Machine learning algorithms enable the primary functionality of FakeGuard, which is to classify news as real or fake based on user input. It accepts both short and long texts in various languages.

Multilingual NLP Support

With the ability to process Hindi and Telugu alongside English, FakeGuard broadens its reach to more users. The model and UI have been designed in a way that makes expansion to other indian regional languages straightforward.

Offline-First and Low-Bandwidth Design

The application’s ability to operate offline is enhanced by its ability to load a pre-trained model which greatly benefits users in rural and remote regions with little or no internet.

Data Storage and Collection

The system structures news submissions along with their predictions into a dataset. With user consent, news inputs and predictions are logged. This contributes to the expanding corpus of classified fake and real news in multiple languages, enabling more advanced AI training and research in Indian languages.

Literature Survey

Literature Survey 1

**Tittle:** “Fake News Detection on Social Media: A Data Mining Perspective”  
**Authors:** Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, Huan Liu  
**Published in:** ACM SIGKDD Explorations Newsletter, 2017  
  
**Summary:**  
This paper presents an in-depth survey of machine learning and data mining-based methods for detecting fake news. It categorizes the existing methods with respect to content-based features (writing style, text semantics) and user-based features (credibility of the user, likes, shares) with a strong focus on the significance of a hybrid approach in order to enhance detection performance.  
  
**Conclusion:**  
The paper proves that NLP-based models such as TF-IDF and semantic analysis are central to identifying misleading linguistic patterns. It lends evidence toward the utilization of interpretable models for real-time fake news detection in use cases such as FakeGuard.  
  
**Literature Survey 2**

**Title:** "Automated Fake News Detection: Are We There Yet?"  
**Authors:** R. Oshikawa, J. Qian, K. Uchiumi  
**Published in:** IEEE Transactions on Computational Social Systems, 2020  
  
**Summary:**  
This paper presents a comparative analysis of various models detecting fake news: classical machine learning, deep learning, and hybrid architectures. It analyzes the performance of these architectures on benchmark datasets and discusses challenges such as data bias, multilingualism, and a lack of standardization.

**Conclusion:**  
It highlights that transformer-based models may look powerful, yet much simpler models, such as Logistic Regression with effective feature extraction (e.g., TF-IDF) turn out to be pretty reliable and deployable in low-resource settings, as shown in the MVP of FakeGuard.

Existing System

There are a number of fake news detection platforms and tools, but they are either manual, internet-based, English-only, or non-open-source. Here is a comparison of systems and why FakeGuard is an enhancement.

1. Snopes.com

Description: Snopes is a popular manual fact-checking site that verifies and explodes rumors and viral stuff.

Features:

Large fact-checked database, Human-checked articles, High credibility

Limitations:

Manual and labor-intensive, English-only, Needs full internet connectivity

Relevance to FakeGuard:

FakeGuard has AI-powered automated fact-checking, regional language support, and offline capability—improving on Snopes' weaknesses.

2. FactCheck.org

Description:

A non-profit site that confirms the truthfulness of political and public statements in America.

Features:

Editorial supervision, Daily updates on going-viral claims

Weaknesses:

English-language and U.S.-centric news, No user contribution or prediction functionality,No data contribution interface

Relevance to FakeGuard:

FakeGuard offers an interactive AI-based prediction system with Indian language support and logs user contributions for corpus development.

3. Google Fact Check Tools

Description:

Google's set of fact-checking tools.

Google offers fact-checking features through aggregating review of claims from verified fact-checking entities.

Features:

Aggregated reviews of claims, Structured metadata used

Limitations:

Based on external verified sources, No prediction or classification model, Internet-dependent

Relevance to FakeGuard:

FakeGuard conducts real-time fake news detection based on a local model without the need for third-party data, thus being applicable in remote and low-bandwidth areas.

**METHODOLOGY**

**Proposed System**

The proposed system, **FakeGuard**, is an **AI-powered, offline-compatible, multilingual application** designed to detect fake news based on the text input provided by users. It uses lightweight NLP and machine learning techniques to classify news articles as **real or fake** in real time.

Unlike traditional fact-checking methods that rely on manual research or online databases, FakeGuard uses a **pre-trained AI model** embedded directly into the application, making it usable in **low-bandwidth or offline environments**. It is built using **Streamlit** as the front-end and **scikit-learn** for machine learning.

**Key Features:**

* **Multilingual Support:** Accepts news input in **English**, **Hindi**, and **Telugu**.
* **Offline-First Design:** Loads a pre-trained model locally using joblib, enabling use without internet after initial setup.
* **Corpus Collection:** With user consent, it logs all input and predictions to build a valuable dataset for future AI model training in Indian languages.
* **Lightweight and Fast:** Optimized for low-resource systems; the model is based on **TF-IDF vectorization** and **Logistic Regression** for quick predictions.

**System Architecture:**

1. **User Interface (Streamlit Web App):**  
   Simple and responsive front-end where users can enter news content, choose language, and view predictions.
2. **Text Preprocessing & Feature Extraction (NLP):**  
   Uses **TfidfVectorizer** to convert raw text into numeric form suitable for model prediction.
3. **Fake News Classifier (AI Model):**  
   A Logistic Regression classifier trained on the Kaggle Fake and Real News dataset is used to label news as “FAKE” or “REAL”.
4. **Corpus Logger (CSV Logging):**  
   User inputs and model results are stored in a secure .csv file for future supervised model improvements.
5. **Offline Model Loading:**  
   The model and vectorizer are saved as .pkl files and loaded during app startup, ensuring **no internet dependency** once deployed.

This system architecture allows FakeGuard to serve as both a **practical tool for misinformation detection** and a **data collection engine** that supports SWECHA’s goal of building culturally inclusive AI through regional language data.

Modules

The **FakeGuard** system is designed using modular architecture to ensure clarity, scalability, and maintainability. Below are the major modules of the system:

**🔹 1. User Input Module**

* **Function**: Accepts user-submitted news content and selected language.
* **Features**:
  + Multilingual input support (English, Hindi, Telugu).
  + Language selection dropdown.
  + Handles input validation and submission.
* **Purpose**: Collects raw text for preprocessing and classification.

**🔹 2. Text Preprocessing & Classification Module**

* **Function**: Converts text into numeric format and classifies it using an AI model.
* **Features**:
  + TF-IDF Vectorization to extract meaningful features from text.
  + Logistic Regression classifier to predict fake or real.
  + Works offline using pre-trained model files.
* **Purpose**: Performs the core task of fake news detection using NLP and machine learning.

**🔹 3. Result Display Module**

* **Function**: Shows prediction result to the user in an interactive format.
* **Features**:
  + Displays output as “FAKE” or “REAL” with appropriate styling.
  + Optional: Confidence score and retry option.
* **Purpose**: Communicates results clearly and allows repeated use.

**🔹 4. Corpus Logging Module**

* **Function**: Stores user inputs and model predictions into a local .csv file.
* **Features**:
  + Records input text, language, and prediction (with user consent).
  + Helps build a labeled fake/real news dataset for future training.
* **Purpose**: Contributes to multilingual corpus growth for responsible AI development.

**🔹 5. Streamlit Front-End UI Module**

* **Function**: Provides the user interface for input, output, and interaction.
* **Features**:
  + Clean and mobile-responsive design.
  + Fast loading and easy navigation.
* **Purpose**: Makes the system accessible to users of all backgrounds, even in low-bandwidth areas.

**Methods and Algorithms**

The core of **FakeGuard** relies on Natural Language Processing (NLP) and classical machine learning to detect fake news based on linguistic features. The system is designed to be lightweight, interpretable, and compatible with low-bandwidth environments, while still delivering effective predictions. Below are the key methods and algorithms used:

**1. Natural Language Processing (NLP)**

**Objective:**  
To clean, process, and extract meaningful patterns from user-submitted news text.

**Technique Used: TF-IDF Vectorization**

* **TF (Term Frequency):** Measures how often a word appears in a document.
* **IDF (Inverse Document Frequency):** Reduces the importance of common words across multiple documents.
* **TF-IDF:** Combines the two to assign higher weights to informative, context-specific words.

**Result:**  
Converts raw news text into a structured vector of numbers that reflects the importance of each word in the context of fake/real classification.

**2. Machine Learning Classification**

**Algorithm Used: Logistic Regression**

* **Type:** Supervised binary classification algorithm
* **Why Used:**
  + Lightweight and efficient
  + High interpretability
  + Performs well on text classification with TF-IDF vectors
  + Suitable for offline use

**How It Works:**

* The model is trained on labeled data (news text + fake/real labels).
* It learns decision boundaries based on word patterns and their weights from the TF-IDF vectors.
* During prediction, it outputs the probability of the news being fake or real.

**Output:**

* Final label: “FAKE” or “REAL”
* Optional: Confidence score or probability value

**3. Model Serialization**

**Tools Used:**

* joblib to serialize the trained **TF-IDF vectorizer** and **Logistic Regression model** into .pkl files

**Purpose:**

* Enables fast model loading in the Streamlit app
* Supports **offline use** by avoiding runtime retraining or API calls

**4. Corpus Logging & Dataset Building**

**Method:**

* Each user input and the corresponding model prediction is saved into a local .csv file
* Includes language metadata and timestamp (optional)

**Purpose:**

* Builds a **multilingual dataset** of real and fake news
* Supports future model retraining with real-world, culturally diverse data

Conclusion

**Project Conclusion**

The **FakeGuard** project successfully demonstrates how open-source Artificial Intelligence and Natural Language Processing (NLP) can be effectively applied to address one of the most pressing challenges of the digital age — the spread of fake news. Designed with simplicity, accessibility, and social relevance in mind, FakeGuard provides a fast, lightweight, and multilingual solution that empowers users to verify the authenticity of news content, even in low-bandwidth and offline environments.

By using **TF-IDF vectorization** and **Logistic Regression**, the system delivers accurate results while maintaining transparency and interpretability — crucial for responsible AI usage. The integration with **Streamlit** ensures that the user interface is clean, responsive, and friendly, making the tool usable by individuals from both technical and non-technical backgrounds.

Beyond functionality, FakeGuard also serves as a **corpus collection engine**, enabling the creation of a diverse dataset of news texts in regional languages like **Hindi** and **Telugu**. This aligns perfectly with SWECHA’s mission to build AI that understands and reflects India’s linguistic and cultural diversity.

Through this project, we have not only built a practical tool for misinformation detection but also laid the foundation for future development in the area of **regional language NLP**, **low-resource AI**, and **data democratization**. FakeGuard exemplifies the potential of AI for social good, proving that powerful solutions can also be lightweight, ethical, and inclusive.

**Futurescope**

The FakeGuard project has significant potential for future development and scalability. One of the most important extensions is to support a wider range of Indian languages such as Tamil, Bengali, Marathi, and Kannada, ensuring deeper linguistic inclusivity. Integrating advanced transformer-based NLP models like IndicBERT or mBERT can enhance the system’s ability to understand complex context and improve classification accuracy across languages. Additionally, introducing voice-to-text input will allow users to speak news snippets in their regional languages, making the tool more accessible for low-literacy populations.

Future versions could also include browser extensions or mobile Progressive Web Apps (PWAs), enabling real-time verification directly on social media or websites, even offline. Integration with fact-checking APIs and sentiment analysis tools could further strengthen the detection of bias, sensationalism, or misleading content. A gamified contribution system, featuring leaderboards and rewards, could motivate users to help build a rich corpus of fake and real news in various dialects. Collaborations with journalists, fact-checking NGOs, and educational institutions could add credibility to the data and promote widespread adoption. Finally, publishing the collected corpus as an open-source dataset would support ongoing research in Indian language NLP and responsible AI development, helping to create a more informed and digitally resilient society.