WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Fake News Detection Using TinyML

S. Pooja[1], A. Vidhya[2], K. Pavithra[3], E. Praveena[4], N. Yashema[5], S. Balaji[6]

[1,2,3,4] Department of Computer Science and Engineering Kingston Engineering College, Vellore, Tamil Nadu, India
[5]AP/CSE, [6]HOD/CSE
poojapoojas2412@gmail.com , vidhya2632004@gmail.com , pavithra4575@gmail.com , praveenakavi2517@gmail.com
yash.engineering@kingston.ac.in , balajicse.engineering@kingston.ac.in

ABSTRACT

In today's digital world, the rampant spread of fake news has emerged as a critical societal challenge. From political propaganda to health misinformation, deceptive news can shape public perception and incite social unrest. This research introduces an efficient, scalable solution leveraging Tiny Machine Learning (TinyML) for real-time fake news detection, optimized for deployment in resource-constrained environments. The system architecture integrates TF-IDF for textual feature extraction and a Random Forest Classifier for classification, subsequently converted into a TensorFlow Lite (TFLite) model to ensure minimal computational footprint. To enhance relevance, the dataset was augmented with current real-world news articles collected from BBC News, addressing the temporal limitations of older datasets. The model was deployed using a Flask API backend and a Streamlit web interface, providing an interactive and user-friendly platform. The proposed system achieved an impressive 99.23% accuracy, confirming its potential for lightweight, real-time misinformation detection.

Keywords Fake News Detection, TinyML, TF-IDF, TFLite, Flask API, Streamlit, Random Forest Classifier, Lightweight ML, Real-Time Detection.

Introduction

The evolution of digital media and the widespread usage of social platforms have led to an explosion of information—both real and fabricated. Fake news spreads faster than ever, often outpacing fact-based journalism, affecting everything from elections to public health. Machine learning models can detect such content, but they are typically designed for cloud-based deployment, requiring significant resources. This study proposes a **TinyML-based fake news detection system**, bridging the gap between high-performance classification and deployment feasibility on low-power devices such as mobile phones or embedded hardware. By combining **traditional NLP techniques** with **resource-efficient ML deployment**, we enable a reliable, portable solution for real-world applications.

Literature Survey

Early fake news detection relied heavily on manual verification or simple keyword-matching. As the field evolved, researchers employed **Natural Language Processing (NLP)** with machine learning models like **Naive Bayes**, **Logistic Regression**, and **Support Vector Machines (SVMs)**. Deep learning techniques such as **LSTM (Long Short-Term Memory)** and **CNNs (Convolutional Neural Networks)** showed promise but were computationally intensive. Despite their high accuracy, these methods are impractical for edge deployment. Literature on **TinyML**-enabled fake news detection is limited, making this study one of the pioneering attempts to create an **efficient, deployable model** suitable for lightweight environments.

Proposed System

The proposed framework is a **five-stage pipeline**:

1. **Text Preprocessing** – Cleaning news headlines/articles to remove punctuation, stop words, and perform tokenization.
2. **Feature Extraction** – Applying **TF-IDF vectorization** to convert text into numerical form.
3. **Model Training** – Using **Random Forest Classifier** with class_weight='balanced' to address class imbalance.
4. **Model Conversion** – Converting the sklearn model into a **TFLite format** compatible with TinyML standards.
5. **Deployment** – Exposing the model via a **Flask API**, and creating a user interface using **Streamlit** for interactive access.

Recent, real news data from **BBC, Times of India** was scraped and integrated into the dataset to ensure the model generalizes well on current trends and topics.

System Architecture

The architecture is structured into four major components:

- **Frontend (Streamlit UI):** Accepts user input (headline/news) and displays results.
- **Backend (Flask API):** Acts as a bridge between UI and the ML model.
- **Model Engine:** Pre-trained **TFLite model** that loads a vectorizer and performs prediction.
- **Storage Layer:** Includes the TF-IDF vectorizer, preprocessed dataset files, and model weights.

This modular architecture ensures separation of concerns and allows easy maintenance or enhancement.

Implementation

1.Data Collection & Preprocessing

- Base dataset: **Kaggle Fake and Real News Dataset** (True.csv and Fake.csv).
- Augmented with **41 recent real headlines** from **BBC News** to overcome data staleness.
- Labeled data: 1 = Real, 0 = Fake.

2.Feature Engineering

- Utilized **TF-IDF vectorizer** with max_features=1000.
- Saved the vectorizer using pickle for consistent feature mapping during inference.

3.Model Training

- Applied **Random Forest Classifier** with balanced class weights.
- Achieved **93.23% accuracy** on the validation dataset.

4.Model Conversion

- The trained sklearn model was converted into **TensorFlow format**.
- Then converted to **TFLite** ensuring compatibility with TinyML (e.g., float32 support).

5.Deployment

- Flask API handles the HTTP POST request containing user input.
- Streamlit frontend provides a simple UI for users to enter text.
- TFLite model loads within the API and returns predictions.

Results and Evaluation

	precision	recall	f1-score	support
0	0.93	0.94	0.93	4650
1	0.93	0.92	0.93	4330
accuracy			0.93	8980
macro avg	0.93	0.93	0.93	8980
weighted avg	0.93	0.93	0.93	8980

Evaluation used a **confusion matrix** and **manual tests** on current news headlines. The model showed consistent performance even on articles not seen during training.

Advantages and Limitations

Advantages:

- Compact model deployable in memory-constrained systems.
- High accuracy with minimal preprocessing.
- Web-accessible UI, no installation needed.
- Adaptability to newly added news sources.

Limitations:

- Model struggles with **ambiguous or sarcastic statements**.
- **Manual scraping** needed for dataset updates.
- Not yet optimized for **microcontroller deployment**.

Future Enhancements

- **Deploy on embedded hardware** like Arduino or Raspberry Pi using actual TinyML devices.
- **Automate data ingestion** using news APIs for live updates.
- **Explore Transformer models** like **DistilBERT + TFLite** for more context-aware predictions.
- **Add multilingual support** to detect regional fake news in local languages.
- Incorporate **explainability tools** to show why a news item is labeled fake.

Conclusion

This project proves the practicality of a lightweight fake news detector using **TinyML**. By combining TF-IDF, Random Forest, and TFLite, the model achieves high accuracy while remaining deployable in resource-limited environments. The modular deployment using Flask and Streamlit makes it both developer- and user-friendly. This work lays the foundation for future innovations in **lightweight misinformation detection systems** on edge devices.

References

- Scikit-learn Documentation – https://scikit-learn.org
- TensorFlow Lite Conversion Guide – https://www.tensorflow.org/lite/convert
- BBC News Headlines – https://www.bbc.com/news
- Kaggle Fake News Dataset – https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset
- TinyML Foundation – https://www.tinyml.org