Project Report

Fake News Classifier

Goutam Mittal -S24CSEU0785

Gyanendra Prakash -S24CSEU0771

Dhruv Gupta -S24CSEU0819

A report submitted in part fulfilment of the degree of

BTech in Computer Science

Supervisor: Dr. Susmita Das

School of Computer Science Engineering and Technology

1. Introduction

This project builds a supervised classifier to detect fake news using combined public datasets (WELFake and ISOT). The goal is to design a practical pipeline that ingests raw news title and text, cleans and preprocesses the content, engineers text features, trains several baseline and deep models, and produces a deployable best model for inference. The dataset used contains ~62k cleaned articles after deduplication and basic cleaning, with binary labels (0: Fake, 1: Real).

Key objectives:

- Build a reliable preprocessing pipeline for noisy news text.
- Compare traditional ML classifiers (Logistic Regression, Random Forest, XGBoost, Naive Bayes) with transformer-based models augmented by an LSTM head.
- Select and export the best performing model for inference.

2. Proposed Works / Methodology

This section summarizes the end-to-end methodology implemented in the project.

Data collection & merging

- Two publicly available datasets were obtained and standardized: WELFake and ISOT (Fake.csv and True.csv).
- Relevant columns (title, text, label) were selected and concatenated into a single dataframe.
- Missing or duplicate records were removed, producing a cleaned dataset of 62,200 rows.

Preprocessing

- Lowercasing, URL and punctuation removal, numeric removal.
- Stopword removal and lemmatization using NLTK.
- Titles and article texts were cleaned separately and then concatenated to form a single combined_text field for modelling.

Feature engineering (traditional)

- TF-IDF vectorization with a vocabulary limit (max_features=5000) and English stopwords removed.

Modeling

Two parallel tracks were trained and compared:

1. **Traditional ML models** trained on TF-IDF features:
    - Logistic Regression
    - Random Forest Classifier
    - XGBoost
    - Multinomial Naive Bayes
2. **Transformer + RNN models (deep learning)**:
    - Pre-trained encoder (RoBERTa and DistilBERT) used as frozen feature extractors.
    - A lightweight RNN head (LSTM / Bidirectional LSTM with 128 units) placed on top of the encoder outputs.
    - Classification head: Dropout(0.3) → Dense(64, relu) → Dense(1, sigmoid).
    - Models trained with Adam (lr=2e-5), BinaryCrossentropy, early stopping on validation AUC.

Training and evaluation

- Data split: 80% training, 20% validation stratified by label.
- Evaluation metrics: Accuracy, ROC-AUC, Precision, Recall, F1; confusion matrices were computed.
- The best model (by ROC-AUC) was saved and its tokenizer exported for later inference.

Deployment readiness

- The final model and tokenizer were saved (best_fake_news_model.h5 and best_tokenizer/) and packaged into a ZIP for distribution.

3. Data structure and algorithms used

Data structure

- Primary data representation: Pandas DataFrame for tabular manipulation of title, text, processed_title, processed_text, combined_text, and label columns.
- Intermediate representations: TF-IDF sparse matrices (scipy sparse matrix) for traditional models; tokenized tensors (input_ids, attention_mask) for transformer models.
- Saved artifacts: model HDF5 files (.h5) and tokenizer folder containing tokenizer JSON/vocab files.

Algorithms and libraries

- **Text preprocessing:** NLTK (stopwords, WordNet lemmatizer) and Python re for regex cleaning.
- **Feature extraction (traditional):** sklearn.feature_extraction.text.TfidfVectorizer.
- **Traditional classifiers:** scikit-learn's LogisticRegression and RandomForestClassifier, xgboost.XGBClassifier, and sklearn.naive_bayes.MultinomialNB.
- **Transformer encoders:** Hugging Face transformers library — roberta-base and distilbert-base-uncased used as frozen encoders via TFAutoModel.
- **Neural head:** TensorFlow / Keras LSTM and Bidirectional LSTM layers for sequence aggregation, followed by dense layers for classification.

- **Training utilities:** tf.data for dataset batching and prefetching, Keras callbacks (EarlyStopping) and metrics (AUC, BinaryAccuracy).

Complexity notes

- TF-IDF + linear models are computationally cheap (time roughly linear in number of nonzero features × samples).
- Transformer + LSTM models are computationally heavy: encoder inference dominates memory and runtime (GPU recommended). Models were trained with encoder frozen to reduce compute and speed up training.

4. Result Analysis

This section reports the quantitative performance of the implemented models on the held-out validation set (12,440 samples).

Traditional models (TF-IDF)

- Logistic Regression — **Accuracy:** 0.9446
- Random Forest — **Accuracy:** 0.9382
- XGBoost — **Accuracy:** 0.9605
- Multinomial Naive Bayes — **Accuracy:** 0.8410

Transformer + LSTM models (validation metrics)

Model	Accuracy	ROC-AUC
RoBERTa + LSTM	0.9359	0.9815
RoBERTa + Bi-LSTM	0.9549	0.9914
DistilBERT + LSTM	0.9401	0.9825
DistilBERT + Bi-LSTM	0.9545	0.9908

Best model

- The RoBERTa + Bidirectional LSTM model achieved the highest ROC-AUC (0.9914) and strong accuracy (0.9549). It

was selected as the final model and exported with its tokenizer.

Confusion matrix / class performance

- RoBERTa + Bi-LSTM showed balanced performance across classes with high precision and recall for both fake and real classes (F1 scores ≈ 0.95). Detailed classification reports and confusion matrices were produced for each model during evaluation.

Observations

- Transformer-based models with a recurrent summarizer (Bi-LSTM) consistently outperformed traditional models in ROC-AUC, suggesting that contextual token representations plus sequence aggregation capture nuanced signals in news text.
- XGBoost on TF-IDF features remained a strong, lightweight alternative when GPU resources are limited.
- Naive Bayes performed poorest here, likely due to vocabulary and token dependencies in short news text that violate Naive Bayes assumptions.

Limitations

- Dataset bias: The merged datasets originate from specific sources which may introduce topical or stylistic biases.
- Generalization: Performance on out-of-distribution news sources (different publishers, languages, or writing styles) needs separate validation.

5. Conclusion

This project implemented and compared traditional and deep learning approaches for fake news detection. Key conclusions:

- Proper data cleaning, deduplication, and combining title and body text produced a robust dataset of 62k samples.

- Transformer encoders (RoBERTa, DistilBERT) combined with a Bi-LSTM head gave the best performance (RoBERTa + Bi-LSTM, ROC-AUC = 0.9914), and were chosen as the final model.
- For resource-constrained deployments, XGBoost on TF-IDF is a competitive and lightweight alternative (accuracy ≈ 0.9605).

Next steps (practical suggestions)

- Evaluate the chosen model on external datasets and time-split validation to measure generalization and temporal robustness.
- Consider lightweight distillation or quantization for faster inference on CPU.
- Add explainability (e.g., SHAP for tree models, attention visualizations for transformer models) to surface why a given article was classified as fake or real.