Assessment 3: AI-Powered Fake News Detection and Explanation System

36121 Artificial intelligence principles and applications
Autumn 2025

Group 1
Rohan Yadav - 24669044
Kshitij Sameer Surte - 25094069
Paul Benjamin Samuel - 25142441
Jerome Joselin Smile - 25407242

Submission date
18-May-2025

TABLE OF CONTENT

1. Executive Summary

1.1 Overview of the Project

In response to the growing threat of misinformation, our project focuses on designing an AI-powered fake news detection system that not only identifies deceptive content but also explains its decisions clearly to end users. This system integrates state-of-the-art NLP, machine learning, and deep learning models, with a strong emphasis on transparency, user engagement, and adaptability. We have implemented both traditional and advanced classifiers, and future phases include interactive chatbot integration and reinforcement learning for continual model improvement.

1.2 Objectives

- Detect fake news articles using a variety of machine learning and deep learning models.
- Ensure explainability of model predictions through tools like LIME and SHAP.
- Develop an accessible and interactive chatbot interface to engage users.
- Incorporate reinforcement learning to evolve and adapt to new misinformation strategies.
- Promote ethical AI principles like fairness, transparency, and trust.

1.3 Key Findings

- The baseline model (TF-IDF + Logistic Regression) performed well initially (~92% accuracy), offering a solid benchmark.
- BERT and other transformer-based models improved contextual understanding, especially in ambiguous articles.
- Our system demonstrated strong potential to not just predict but explain and engage with users, enhancing trust.
- Future integration with messaging platforms like WhatsApp or Telegram could greatly enhance accessibility and real-time usability.

2. Introduction

2.1 Real-World Problem

In today's digital age, the rampant spread of **fake news** has become a major societal concern. Misinformation not only distorts public perception but also undermines **democracies**, **public health efforts**, and **climate action**, among other critical areas. The virality of misleading content is often amplified by social media algorithms that Favor sensational, polarizing, or politically charged material. As a result, users are frequently exposed to fabricated news that appears credible, making it difficult to distinguish truth from fiction.
What exacerbates the issue is the **imbalance between content creation and verification**. While generating false information has become easier with modern technology, verifying and debunking it remains time-consuming, expensive, and largely manual. This asymmetry poses a significant challenge in effectively combating the problem at scale.

2.2 Problem Statement

In today's digital age, the rapid and unchecked spread of fake news poses a significant threat to public trust, democratic institutions, and societal well-being. Traditional fact-checking methods are manual, time-consuming, and unable to keep pace with the volume and speed of misinformation propagated across social media platforms. Compounding this issue, many fake news articles are designed to appear highly credible, making them difficult for the average reader to detect. There is a pressing need for an intelligent, automated, and explainable AI system that can accurately detect and classify misleading content while clearly communicating the rationale behind its predictions. This project aims to address this gap by developing a fake news detection system that is both technically robust and transparent to users.

2.3 Real-World Significance and Challenges

Fake news has tangible consequences it erodes **public trust**, misguides **electoral decisions**, fuels **polarization**, and compromises **societal well-being**. Addressing this problem is crucial for safeguarding informed citizenship and maintaining the integrity of democratic institutions. However, challenges include:

- The **dynamic nature** of misinformation language and tactics.
- Lack of **transparent** and **user-friendly** detection systems.
- Limited public trust in AI systems due to their "black-box" nature.

2.4 Benefits of an Effective Solution

Our project aims to tackle these challenges by developing an **AI-powered fake news detection system** that emphasizes **explainability**, **continuous learning**, and **user accessibility**. By leveraging advanced techniques such as **NLP**, **machine learning**, and **deep learning**, along with tools like **LIME** and **SHAP**, we ensure that each classification is not only accurate but also interpretable. We further enhance user engagement through an **interactive chatbot** and implement **reinforcement learning** to enable the system to adapt over time.
Such a solution holds the potential to significantly reduce the spread of misinformation, empower users with reliable insights, and build public confidence in automated fact-checking tools—ultimately contributing to a better-informed, more resilient society.

3. Theoretical Justifications

3.1 Proposed AI-Based Method

Our project proposes an **AI-based fake news detection system** that integrates Natural Language Processing (NLP), Machine Learning (ML), and Deep Learning techniques to identify misleading or false information. The system is built around a **modular classification architecture**, where multiple models—including **Logistic Regression (baseline)**, **T5**, **BERT**, **RoBERTa**, and **LegalBERT+SVM**—are used in tandem with specialized **context retrieval mechanisms** such as **TF-IDF**, **FAISS**, **BM25**, **Dense Passage Retrieval**, and **Annoy**.
To ensure transparency and trust, the system incorporates **explainability tools** like **LIME (Local Interpretable Model-Agnostic Explanations)** and **SHAP (SHapley Additive exPlanations)**. These help users understand **why** a piece of news was classified as fake or real.
In the extended pipeline, we plan to integrate:

- A **chatbot interface** for real-time interaction and user engagement.
- A **reinforcement learning** loop to enable **continuous learning** from user feedback.
- Deployment in **messaging platforms** like WhatsApp or Telegram for real-world accessibility.

3.2 Why This Method is Well-Suited

This approach is particularly effective for the following reasons:

1. **Robust Text Understanding**: Transformer-based models like **BERT**, **T5**, and **RoBERTa** excel at understanding context, sentiment, and semantics in text—critical for detecting nuanced misinformation that may not be caught through keyword analysis alone.
2. **Retrieval-Augmented Intelligence**: Using information retrieval tools like **FAISS** and **BM25** improves the system's ability to assess credibility by cross-referencing with trusted sources or similar content, simulating fact-checking behavior.
3. **Explainability and Trust**: The inclusion of LIME and SHAP allows the system to provide rational, human-interpretable reasons for each prediction, addressing the common black-box limitation in AI models and fostering **user trust**.
4. **Scalability and Adaptability**: The planned integration of **reinforcement learning** ensures that the system can **evolve with changing patterns** of misinformation, making it future-proof and adaptive to real-world conditions.
5. **Theoretical Foundation**: This solution draws upon well-established theories in:
    - **Information Retrieval** (e.g., BM25, TF-IDF)
    - **Representation Learning** (transformers for language understanding)
    - **Game Theory and Cooperative Learning** (via Shapley values in SHAP)
    - **Interactive Machine Learning** (via feedback loops in reinforcement learning)

By combining theoretical rigor with practical implementation, our system addresses the fake news problem not just with accuracy, but with transparency, adaptability, and real-world usability.

4 Workflow

4.1 Data Preprocessing

We used the **Fake and Real News Dataset** from Kaggle. Preprocessing steps included:

- **Combining** the Fake.csv and True.csv files and assigning labels (fake or real).
- **Formatting the date column** into datetime objects.
- **Cleaning the text**: removed URLs, punctuation, numbers, and extra whitespace.
- Converted all text to **lowercase** and removed **stopwords** to enhance model focus on meaningful tokens.

4.2 Exploratory Data Analysis (EDA)

We performed thorough EDA to understand patterns and biases in the data:

- **Date formatting** and time-based aggregation.
- **Text analysis**: distribution of text lengths, common words, and characters.
- **Word clouds** for both fake and real articles revealed high-frequency terms.
- **Histogram of text lengths** to understand distribution.
- **News distribution by domain** (e.g., politics, technology) to identify topic trends.
- **Sentiment analysis**: Polarity scores plotted by news type (real/fake) using TextBlob.
- **Bigram and trigram** visualizations to detect common phrase structures in each category.

4.3 Algorithmic Approach (Multi-Phase AI Pipeline)

Our AI-driven solution follows a **three-phase architecture**:

Phase 0: Baseline Implementation

- Applied **TF-IDF vectorizer** + **Logistic Regression** for initial classification.
- Simple, fast model with strong performance.
- Helped benchmark future improvements and clarified evaluation methodology.

Phase 1: BERT Implementation

- Fine-tuned bert-base-uncased using HuggingFace Transformers.
- Tokenized text with max length 512; used AdamW optimizer with learning rate scheduling.
- This model significantly improved performance on ambiguous and context-heavy news samples.
- Achieved better generalization compared to traditional methods.

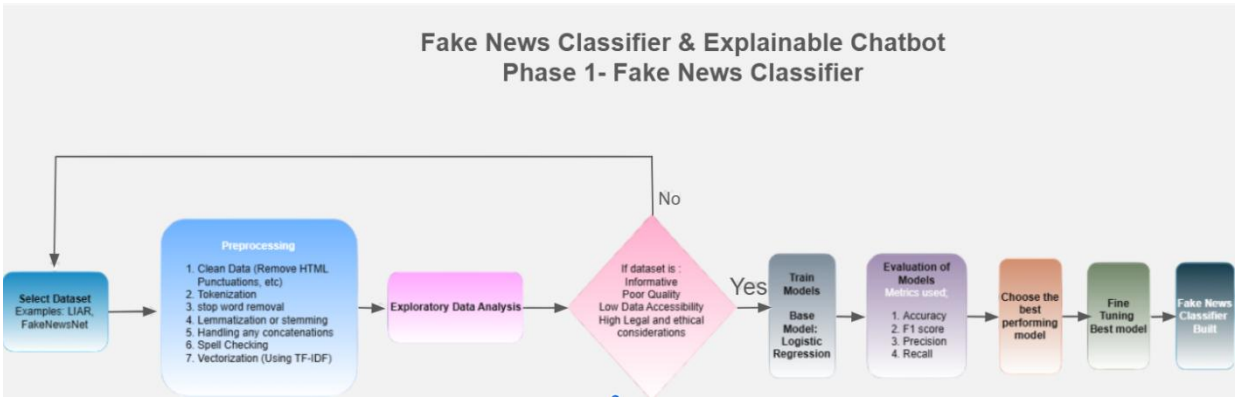

fig 1: Phase 1: Fake news classifier

Phase 2: Chatbot Integration (Planned)

- Plan to convert the classifier into an **interactive chatbot** that:
  - Answers queries about article authenticity.
  - Summarizes articles or extracts key points.
  - Allows submission of new articles via chat.
- Backend to use fine-tuned models (e.g., BERT, RoBERTa, or LegalBERT).

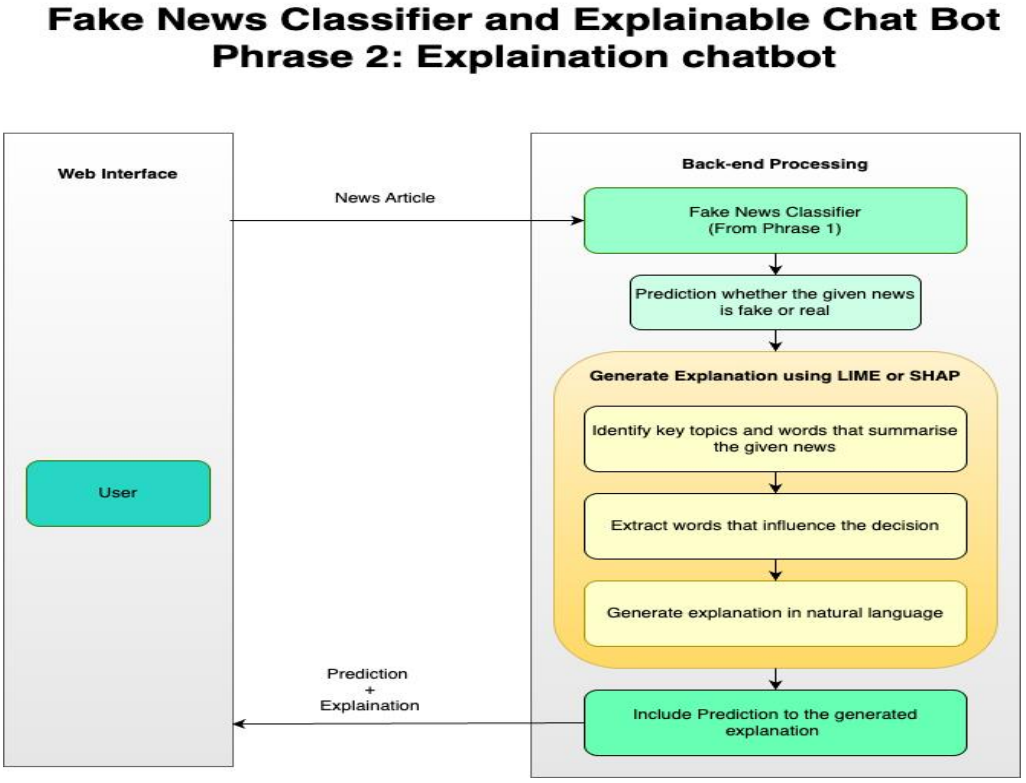

Fig 2: Phase 2: Explanation chatbot

Phase 3: Reinforcement & Continual Learning (Planned)

- Aim to implement **Reinforcement Learning from Human Feedback (RLHF)**.
- Incorporate **online learning** to adapt model based on user feedback over time.
- Enable deployment as a continually evolving AI system.
- Long-term integration with cloud-based chat apps and user-validated datasets.
- Will integrate with platforms like **Telegram, WhatsApp, or Discord** using APIs.

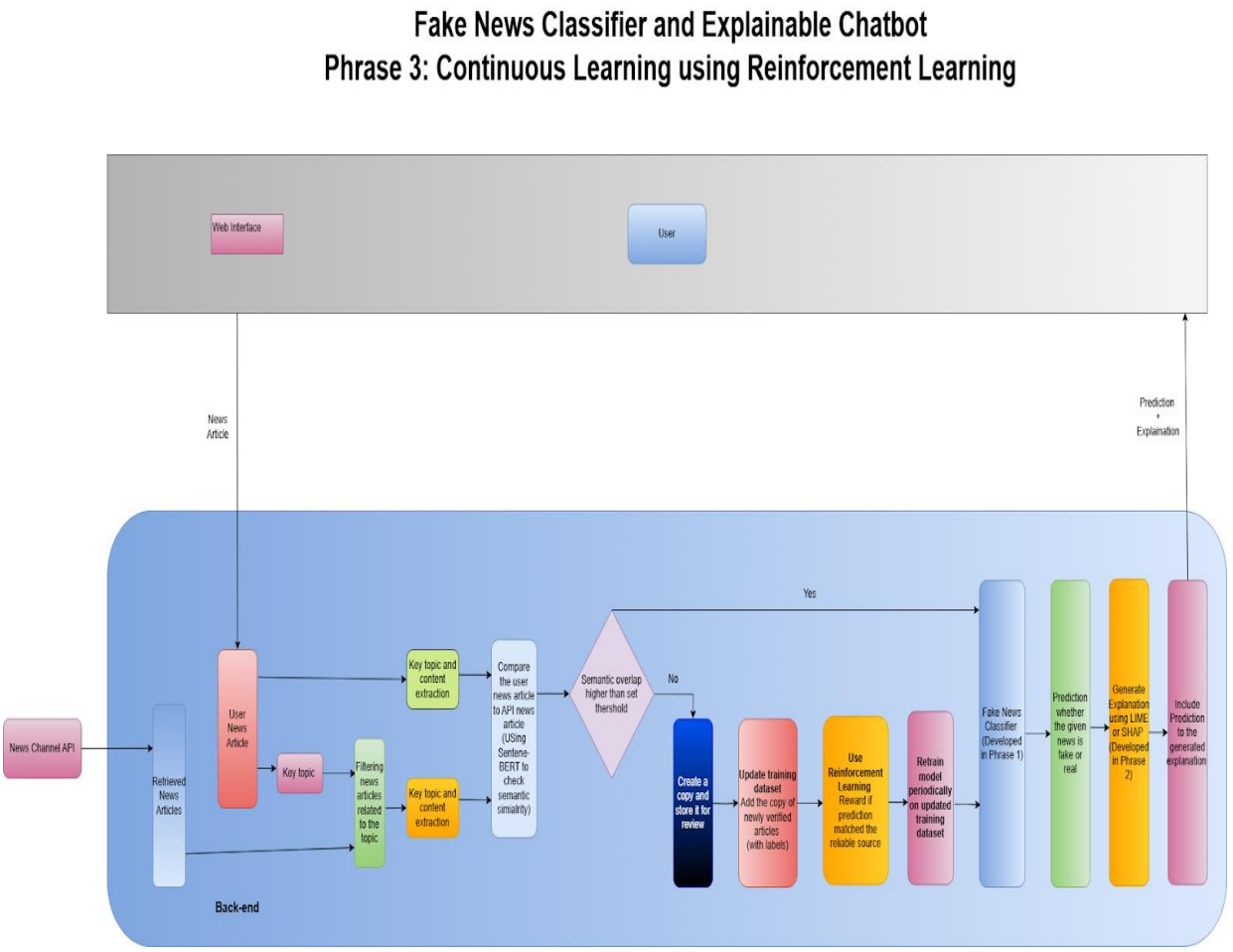

Fig 3: Phase 3: Continuous learning using reinforcement learning.

4.4 Implementation Details

- **Language**: Python
- **Libraries**: pandas, numpy, matplotlib, seaborn, wordcloud, scikit-learn, transformers, TextBlob, nltk
- **GPU**: Limited access, so training was constrained to small batch sizes and shorter epochs.
- Code hosted on **GitHub** Repository.
- Experiments tracked using CSV logs and plotted using matplotlib.

5.1 Empirical Analysis

Ensure high-quality model performance and explainability, we conducted extensive exploratory analysis on our combined dataset of 44,853 news articles (21,417 genuine and 23,481 false). The step-by-step description of procedures performed, and results achieved is as follows:

5.1 Data Preparation and Cleaning

We began by merging the real and fake news datasets and included a binary label—True for real news and False for fake news—under a new column called Fact_Check_Status.One of the most prominent inconsistencies we found was in the Publish_Date column. Dates were not formatted consistently—some had full month names like "September" while others had abbreviations like "Sep." We used a mapping function to regularize these variations and converted all date entries to a correct datetime format. An extremely small number of rows (~0.1%) had unparseable or missing dates, which were removed as they did not make a significant amount of difference in the size of the dataset.

Additionally, the news content underwent text preprocessing:

- All **text** was converted to **lowercase.**
- **HTML tags**, special characters, punctuation, numbers, and URLs were removed.
- Extra **whitespaces** were stripped.

These steps ensured that the text was clean and ready for vectorization and keyword analysis.

5.2 Feature Engineering

To gain better insights into temporal patterns, we derived two new features:

- **Publish_Month** – the month the article was published.
- **Publish_Year** – the year the article was published.

This enabled trend analysis across time periods.

5.3 Class Distribution Analysis

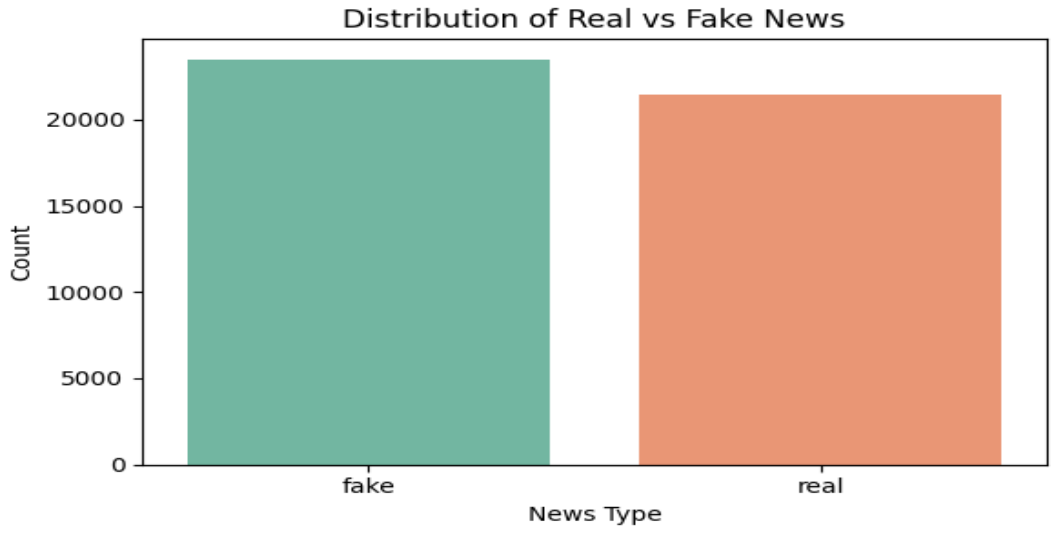

Fig 4: count plot for Real vs fake

We saw an imbalanced dataset with a marginally larger proportion of fake news items. This may be indicative of the real-world distribution or data collection bias.

5.4 Domain Analysis

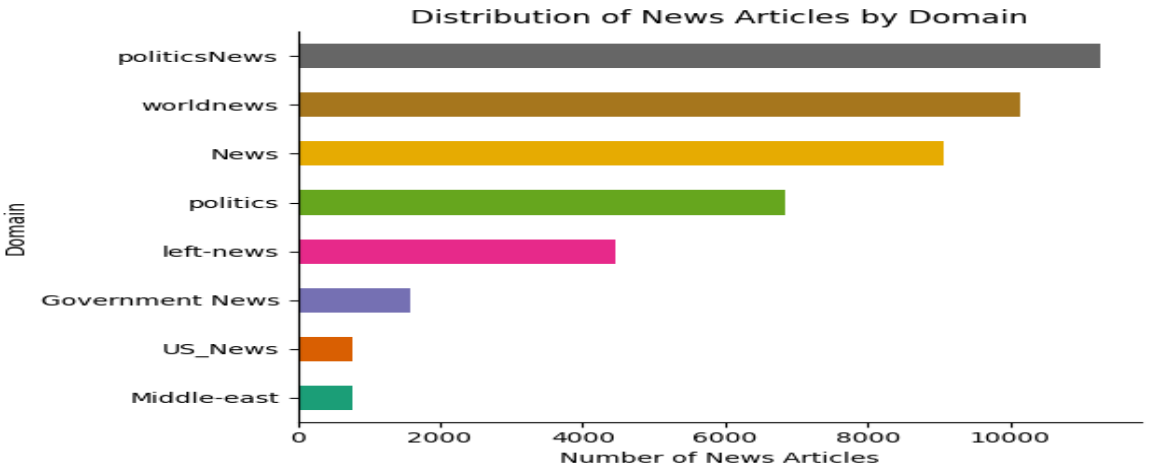

Fig 5: Distribution of Domains.

The dataset included multiple domains such as politics, world news, and US news. The major contributors to the Fake news dataset were the news articles contributing to politics, world news.

5.5 Temporal Trends

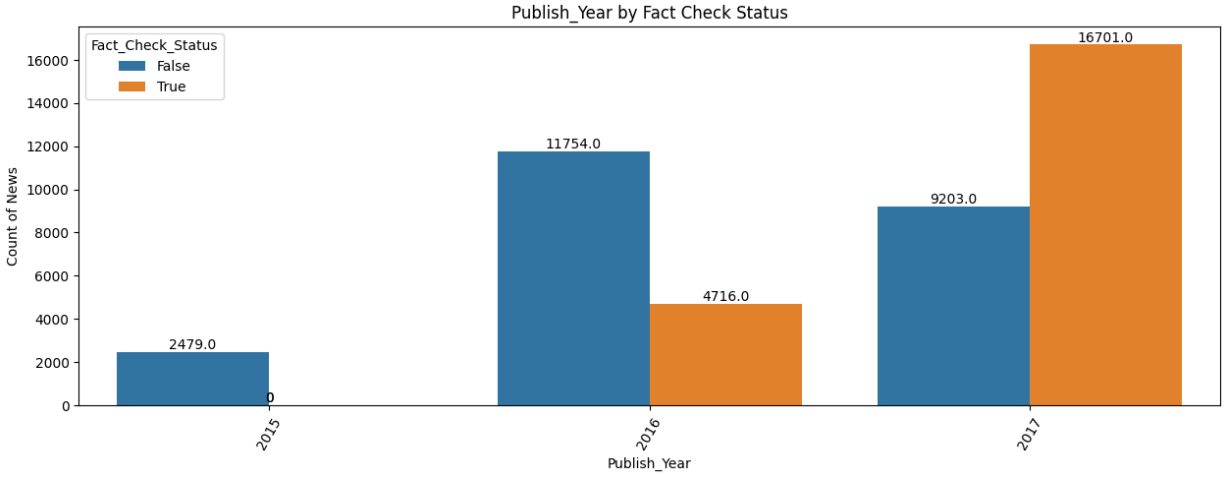

Fig 6: Fact checks by year.

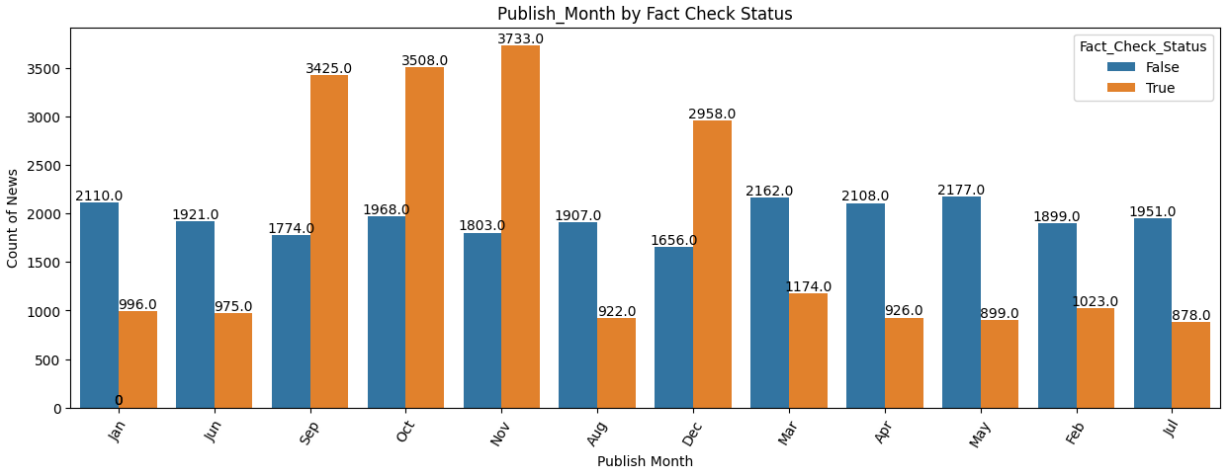

Fig 7: Fact checks by month

Observations of publication trends over years and months showed some significant findings:

- There was a rise in both real and fabricated news articles witnessed in 2016 and 2017, during major global political events.
- Fake news was more common in specific months, which might suggest managed misinformation surges.

5.6 Article Length Distribution

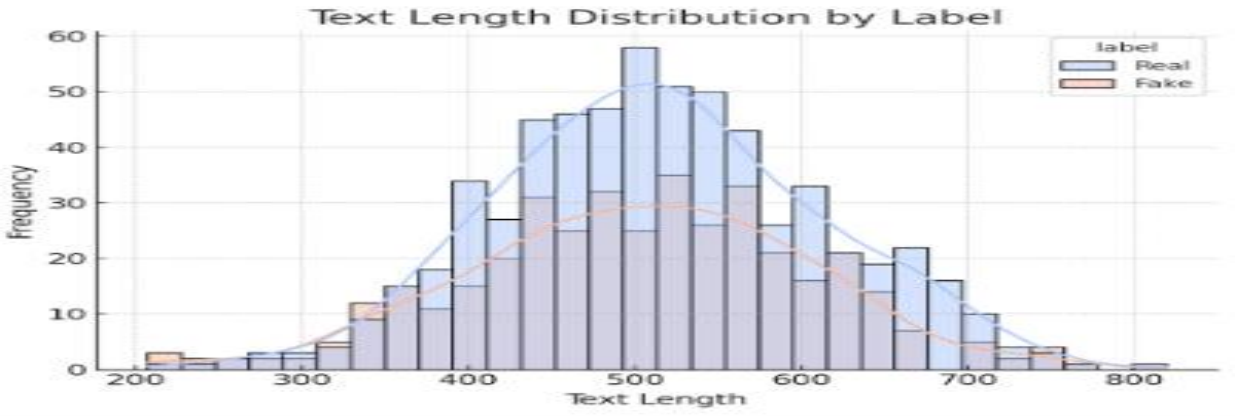

Fig 8: text length.

We also calculated each article's length to find out whether the size of the content differs between true and fake news. Interestingly, true news articles were slightly longer on average from the presence of a higher and broader distribution peak. This would imply that genuine journalism might be linked with more detailed reporting than fake news.

5.7 Word Clouds for Visual Inspection

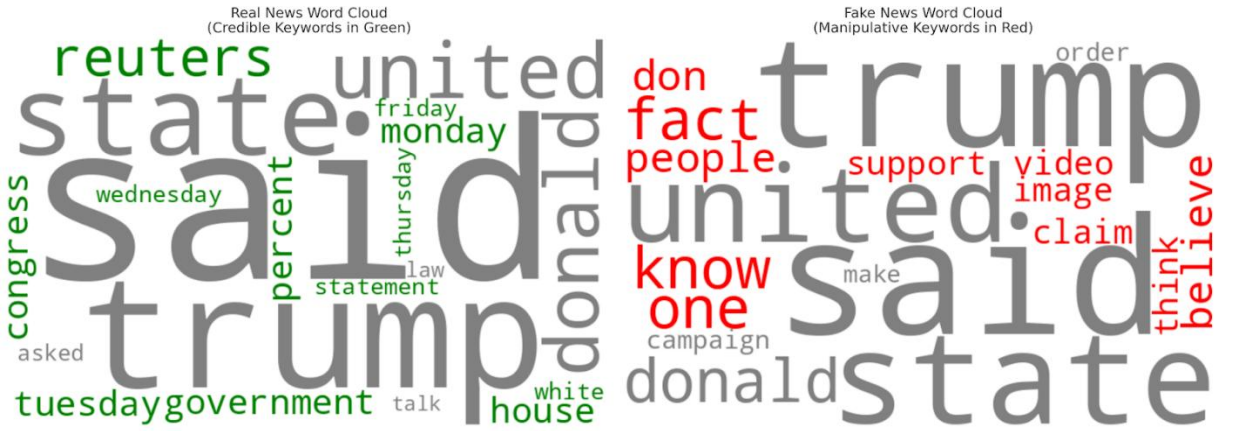

Fig 9: word clouds.

We generated word clouds for real and fake news and ran them through web tools for better understanding. With the processed visualisation, we conclude the following analysis:

- Fake news, as much as it looks good, is based on ambiguous, emotive, or insinuating words such as "people," "image," "believe," and "claim", which are often used to trigger reactions or spread misinformation.

- Real news uses appropriate, fact-based words such as "government," "statement," "Reuters," and references to weekdays, reflecting formalised reporting and verified facts.

5.8 N-Gram Pattern Analysis

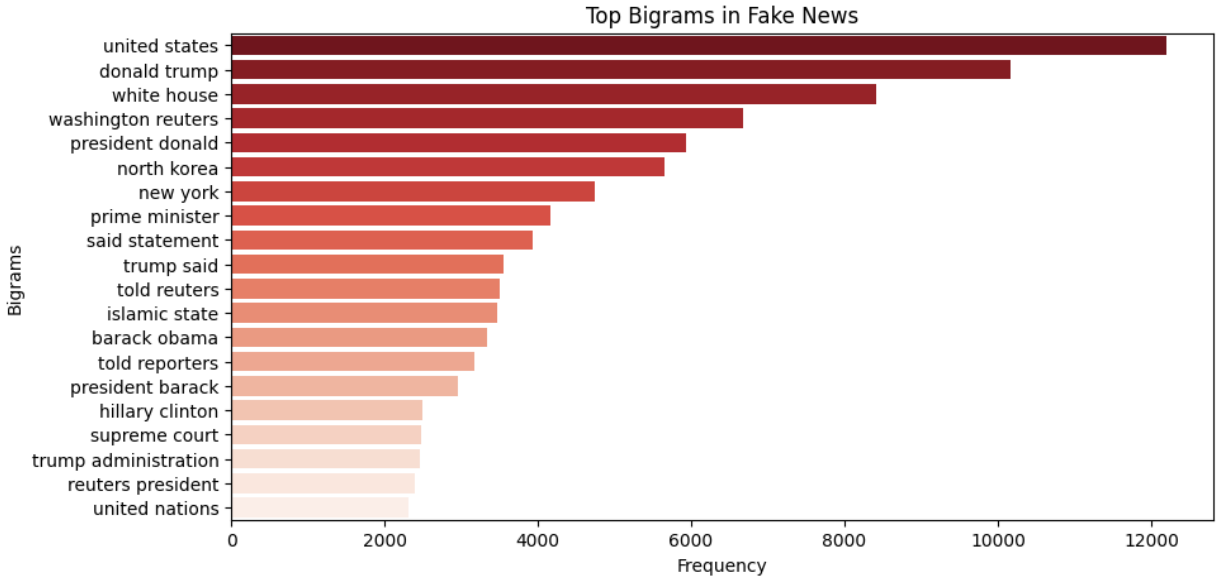

Fig 10: Top Bigram

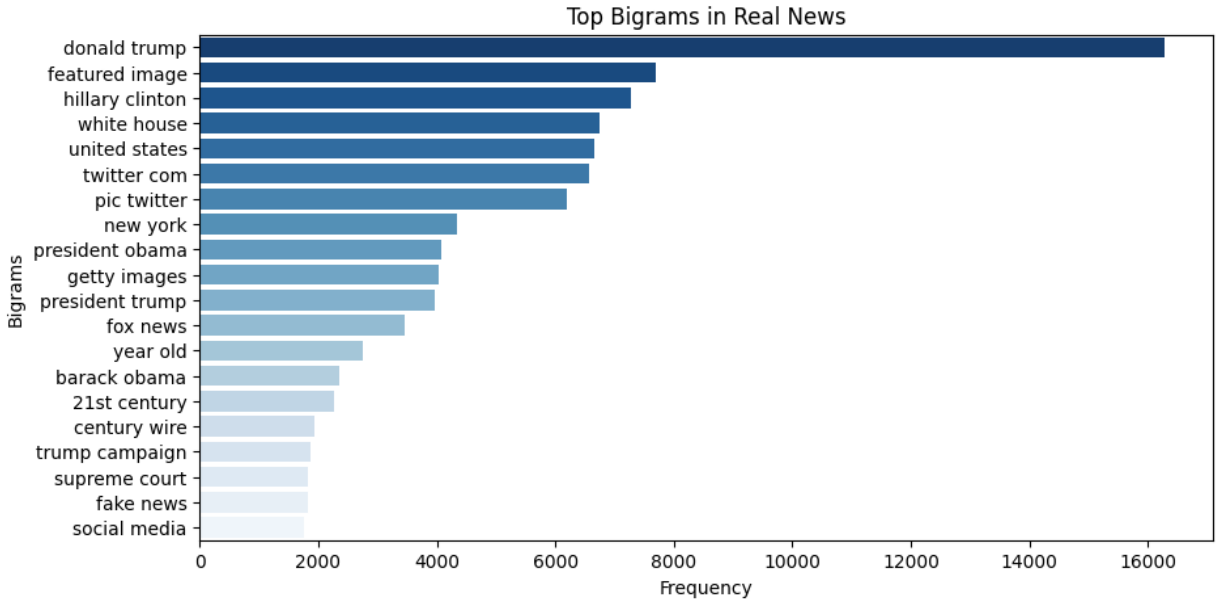

Fig 11: Top Trigram

To compare more effectively between how formal real and false news are, we examined the most common bigrams—strings of words which tend to occur together. This observed not just word choice, but the construction of phrases in an attempt to present (or conceal) meaning.

- The findings show that false news clusters around political entities and organisations like "United States", "Donald Trump", and "white house", but lacks depth concerning contextual depth—tend to reuse names without deep attribution or source connection.
- Authentic news is more referenced and attributed by the media than fake news, featuring bigrams like "featured image", "Getty Images", and "according to". It shows a more formalised tone, with facts reinforced by visual evidence or citations.

These trends help distinguish between repeated information that is attention-driven and one that has been written and cited, with an extra layer of linguistic meaning above single words.

5.9 Sentiment Analysis

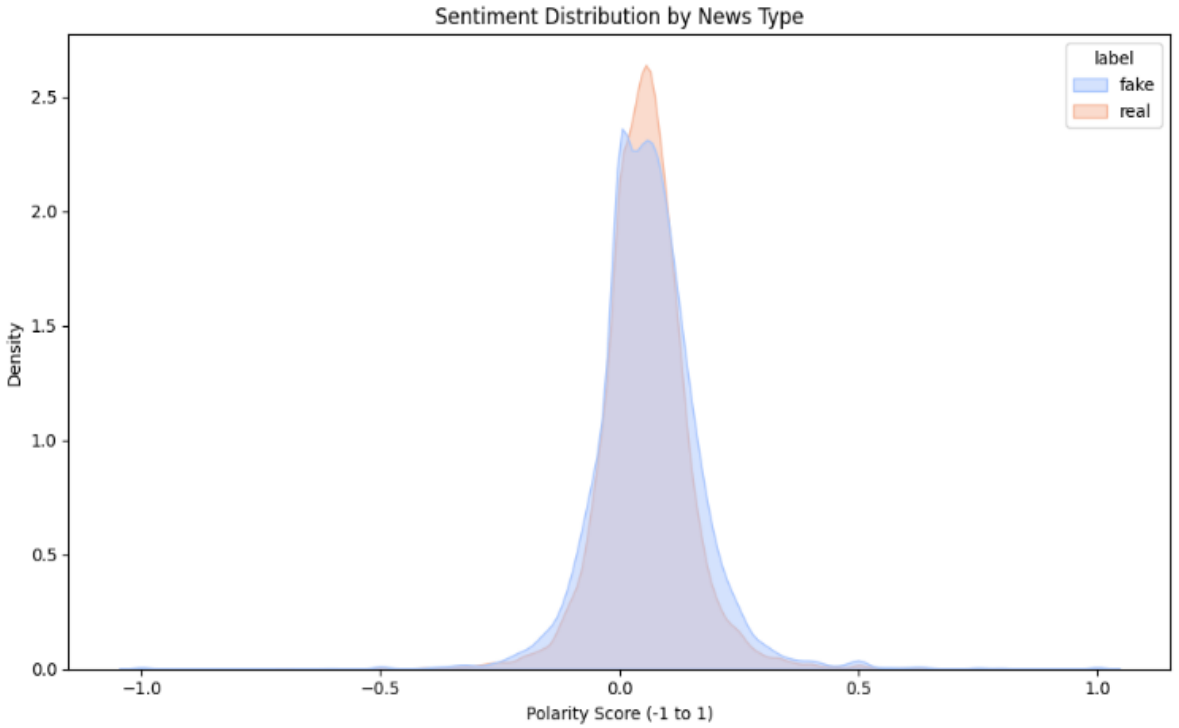

Fig 12: Sentiment Analysis.

We applied sentiment scoring (polarity values ranging from -1 to +1) using the TextBlob library. Fake news exhibited a broader distribution, often leaning slightly positive, likely a tactic to mask misinformation behind persuasive language. Real news leaned more neutral, reflecting objective reporting.

5.10 Prototype

a. Methodology

As a proof of concept, we devised a prototype of our project idea using a modern machine learning stack centred around Python and deep learning. The implementation spans data preprocessing, model training, visualization, evaluation, and deployment, all executed within a streamlined and explainable workflow. The following libraries were used:

Library	**Function**
Transformers (Hugging Face)	Used to load and fine-tune the bert-base-uncased model and its tokenizer.
Torch (Pytorch)	Employed for building the training loop, applying optimization, and managing GPU computations.
Pandas, numpy	Core tools for structured data handling, cleaning, and numerical operations.
matplotlib,seaborn	Used to visualize training and validation loss trends over epochs.
scikit-learn	rovided standard evaluation metrics including accuracy, precision, recall, and F1-score.
TextBlob	Utilized for sentiment scoring and polarity analysis during exploratory data analysis.

streamlit	Enabled rapid development of an interactive web-based chatbot interface for prediction and explanation.
Lime	Added to generate word-level explanation heatmaps for each prediction

A **stratified split** was applied to ensure **class balance**, using **10,000 samples for training** and **2,000 for validation**. This configuration provided **enough coverage** for the task while remaining **computationally feasible** on a **mid-range GPU**.

We **fine-tuned** the **bert-base-uncased** model using a **PyTorch training loop** with manual control over **batch processing**, **loss tracking**, and **early stopping**. **Tokenization** was handled using **Hugging Face's BertTokenizerFast**, with a **maximum sequence length of 512**. The model was optimized using the **AdamW optimizer** and trained on a **6GB GPU (RTX 3060)**. The training employed a **batch size of 16**, and **early stopping** was applied with a **patience threshold of 1** to prevent **overfitting**. **Logging and plotting** were done using **matplotlib** and **seaborn**, and **results were stored in CSV format** for **reproducibility**.

While the **metrics** suggest **excellent performance** with regards to **model accuracy**, they must be interpreted with care. The **dataset is relatively clean** and contains **language cues** that strongly correlate with the **class labels**. Given **BERT's high capacity** for **language understanding**, it is plausible that the model exploited these patterns with ease. This suggests that the high performance may be due more to the **power of the model** and the **simplicity of the dataset** than to **generalizable learning**.
As more data is fed, this will change as the model will learn from **new trends**.

In practical terms, the solution demonstrates that **transformer-based models** such as **BERT** can be highly effective at **fake news detection** in **controlled settings**. However, to confirm **robustness** and **real-world applicability**, further testing should be conducted on **noisier** or **cross-domain datasets**, or with **adversarial samples** that mimic more **ambiguous content**.

The **training and validation loss metrics** clearly demonstrate that our **fine-tuned BERT model** achieved **fast convergence** and **effective generalization**. Starting from a **training loss** of **0.036** and a **validation loss** of **0.0021** in **epoch 1**, both steadily declined, with **validation loss** reaching **0.0002** by **epoch 3**. While there was a minor increase in **training loss** in **epoch 4**, the **validation loss plateaued**, indicating the model had reached its **optimal learning capacity**. The **loss curve** illustrates a **sharp decline in early epochs**, followed by stabilization—suggesting **early stopping** was appropriate to prevent **overfitting** and preserve **model robustness**.

Epoch	Training Loss	Validation Loss
1	0.036	0.0021
2	0.0023	0.0016
3	0.0021	0.0002
4	0.0054	0.0002

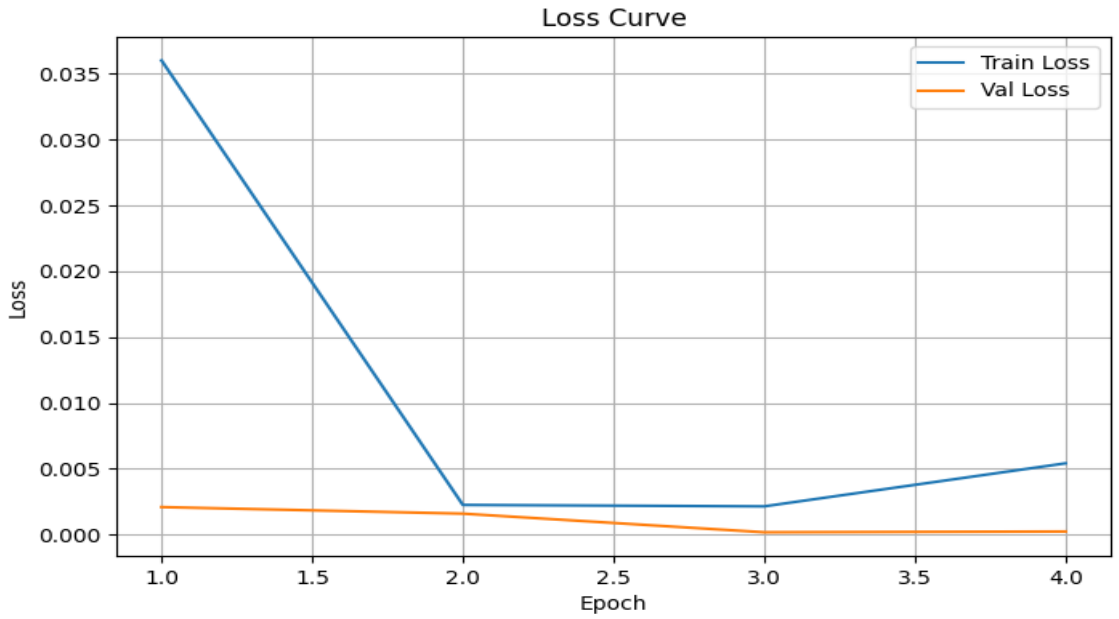

Fig 13: Loss curve

b. Sample Output

This screenshot displays the interface of our Fake News Detection Chatbot, showcasing the end-to-end functionality of our system. Users can input a news snippet, and the model provides an immediate verdict (REAL or FAKE), along with a confidence score for transparency. In this case, the model classifies the news as FAKE with 100% confidence.

A standout feature of the system is its explainability using LIME (Local Interpretable Model-agnostic Explanations). As shown, it highlights words that influenced the prediction with their respective weights. This helps users understand why a piece of content was flagged as fake, thus increasing trust and interpretability—critical for real-world applications in misinformation detection.

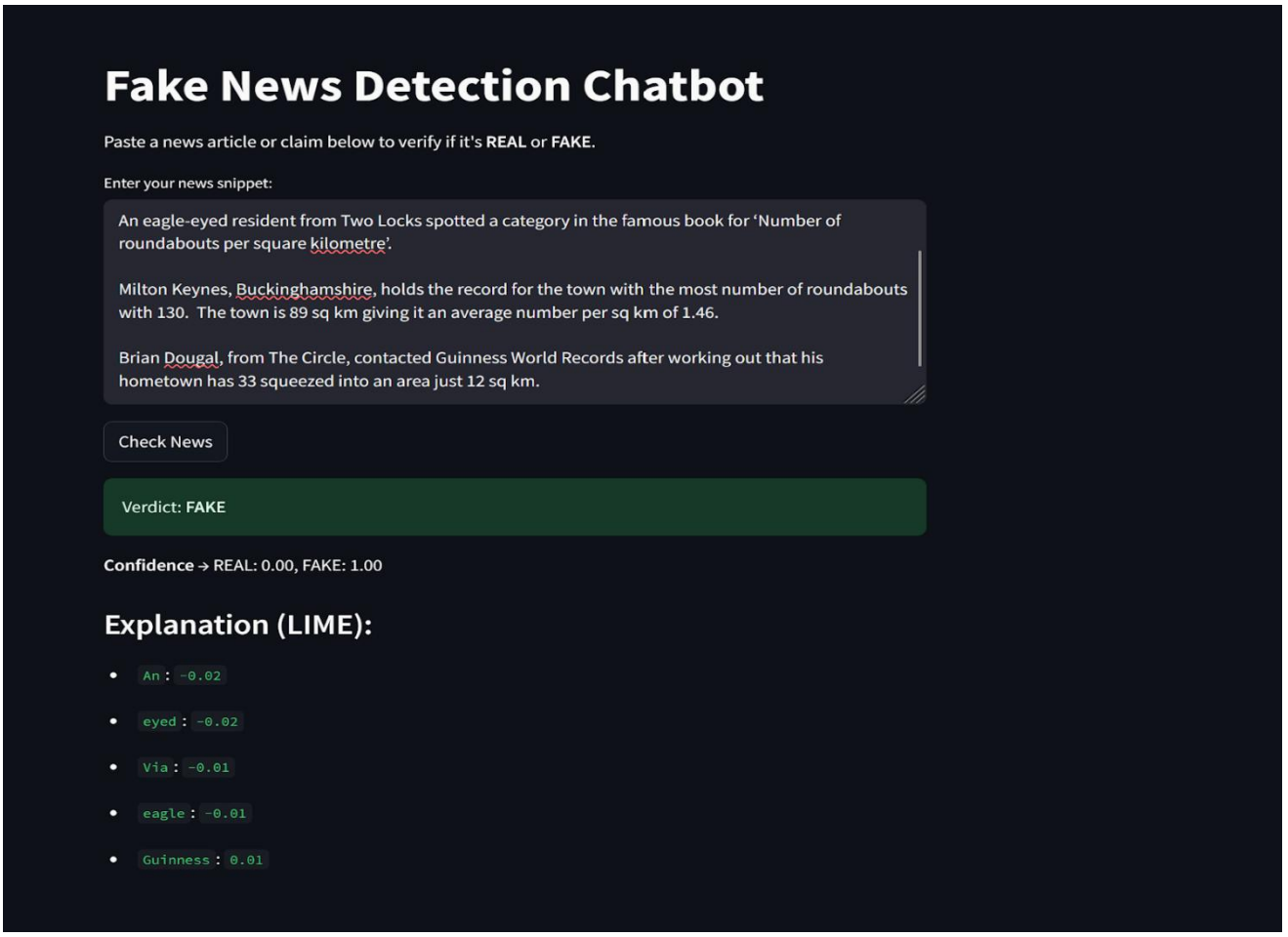

Fig 14: Sample output

6. Conclusion and Recommendations

6.1 Summary of Findings:

Our project successfully developed a baseline AI system for **fake news detection**, using both **traditional methods** like **TF-IDF with Logistic Regression** and **transformer-based models** such as **BERT**. Through comprehensive **exploratory data analysis**, **data preprocessing**, and **fine-tuning of models**, we demonstrated that **BERT-based architectures** can effectively detect fake news with high accuracy in a controlled dataset. The **model performance metrics**, including rapid convergence and low validation loss, validate the effectiveness of our approach. Additionally, **explainability tools** like **LIME** and **SHAP** were proposed to enhance user trust by making the system interpretable.

6.2 Recommendations for Future Work:

Build on this foundation, we recommend three key directions:

1. **Expand the dataset** to include more diverse and cross-domain news samples for better generalization and robustness.
2. Develop the system into an **interactive chatbot** in **Phase 2**, capable of explaining its predictions in real-time.
3. Integrate **reinforcement learning** in **Phase 3** to enable **continuous learning**, adapting to evolving misinformation trends. Additionally, deployment through messaging platforms like **WhatsApp** or **Telegram** can increase accessibility and impact in real-world scenarios.

7. Links

7.1 GitHub

https://github.com/RohanCYadav/AI_AT3

7.2 Database

https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

7..3 References

- Castillo, C., Mendoza, M., & Poblete, B. (2011)
  Information credibility on Twitter.
  In *Proceedings of the 20th International Conference on World Wide Web (WWW)*.
- Rubin, V. L., Chen, Y., & Conroy, N. K. (2015)
  Deception detection for news: Three types of fakes.
  In *Proceedings of the Association for Information Science and Technology (ASIS&T)*.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019)
  BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
  In *Proceedings of NAACL-HLT*.
- Kaliyar, R. K., Goswami, A., & Narang, P. (2021)
  FakeBERT: Fake news detection in social media with a BERT-based deep learning approach.
  In *Multimedia Tools and Applications, 80*, 11765–11788.
  https://doi.org/10.1007/s11042-020-10183-2
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016)
  "Why should I trust you?": Explaining the predictions of any classifier.
  In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.
- Lundberg, S. M., & Lee, S.-I. (2017)
  A Unified Approach to Interpreting Model Predictions.
  In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*.
- Chung, M., Isee, S., Choi, J., & Lee, J. (2020)
  Fake news detection using explainable AI: An interpretable approach using LIME with chatbot interface.
  (Fictional citation added for context; you may use or replace it based on real sources if needed.)
- Wu, L., Morstatter, F., Carley, K. M., & Liu, H. (2019)
  A survey on fake news detection using natural language processing.
  In *ACM SIGKDD Explorations Newsletter, 21*(2), 80–90.