

# Truth Lens: A Multi-Layer AI-Powered Fake News and Spam Detection System Using Machine Learning Wikipedia Cross-Verification, and Large Language Models

Kavya Shree M<sup>1</sup>, Mallikarjun Sonapatti<sup>2</sup>, Kadiri Pavan Kumar<sup>3</sup>, Janani G<sup>4</sup>, Bhagya K<sup>5</sup>

<sup>1234</sup> Department of Computer Science and Engineering,  
CMR University, Bengaluru, Karnataka, India

<sup>5</sup> Assistant Professor, Department of Computer Science and Engineering,  
CMR University, Bengaluru, Karnataka, India

\*\*\*

**Abstract**—In the modern digital information environment, the rapid spread of fake news and spam messages creates significant risks to public communication, financial security, and trust in institutions. Many existing detection systems rely on single-layer approaches that limit accuracy and adaptability. This paper proposes TruthLens, a multi-layer artificial intelligence framework that integrates Machine Learning (ML), Wikipedia-based fact verification, and Large Language Model (LLM) analysis for reliable misinformation detection. The machine learning layer employs a Logistic Regression model trained on TF-IDF features extracted from 44,898 labeled news articles. Experimental evaluation achieves 97.2% accuracy for fake news classification and 98.5% accuracy for spam detection. A Wikipedia API layer performs real-time fact validation using cosine similarity, while the Groq LLaMA-3.3-70B model generates human-readable explanations for detection results. Results from all detection modules are aggregated through a confidence-based decision process to produce the final system verdict. The system is implemented as a Flask-based web application supporting REST APIs, PDF report generation, search history, and GDPR-compliant data handling.

**Keywords** — Fake News Detection, Machine Learning, Natural Language Processing, Spam Detection, Artificial Intelligence.

## 1. INTRODUCTION

News consumption has significantly transitioned from traditional print and broadcast media to digital platforms, greatly increasing the speed and scale of information dissemination. Although this transformation improves global access to information, it also enables the rapid circulation of fake, misleading, and intentionally fabricated content. Empirical studies consistently demonstrate that false information spreads faster and reaches wider audiences than verified news, primarily because it exploits emotional triggers such as outrage, fear, and confirmation bias [1].

This issue becomes particularly serious during major societal events such as elections, public health crises, and

social movements, where misinformation can influence public perception, trigger social unrest, and produce real-world consequences. Concurrently, spam SMS messages contribute to financial fraud worth billions of dollars annually, targeting mobile users with deceptive promotions and phishing campaigns.

Most existing automated detection systems rely on a single analysis layer, which limits their ability to handle adversarial content and complex contextual information. TruthLens addresses this gap by integrating three independent, complementary detection mechanisms into a unified, production-ready pipeline.

### 1.1 Key Contributions

The main contributions of the proposed TruthLens system are summarized below:

- Multi-Layer Detection: A three-layer architecture integrating machine learning classification, real-time Wikipedia fact verification, and large language model-generated explanations, fused by a confidence-weighted combination engine.
- High Accuracy Without GPU: 97.2% fake news accuracy and 98.5% spam accuracy using Logistic Regression and TF-IDF, deployable on standard CPU hardware in under 5 ms.
- Explainable Verdicts: Groq LLaMA 3.3 70B generates a natural language explanation for every detection decision, resolving the black-box limitation of traditional classifiers.
- Full-Stack Deployment: Complete web application with Flask REST API, PDF report export, Search

functionality, search history, usage analytics, and GDPR-compliant data management.

## 2. LITERATURE REVIEW

### 2.1 Machine Learning for Fake News Detection

Initial studies in fake news detection applied traditional machine learning techniques, including Naïve Bayes and Support Vector Machines, with bag-of-words feature representations, achieving performance ranging from 85% to 93% accuracy on benchmark datasets [2]. The introduction of TF-IDF representations improved results by assigning discriminative weights to terms rare across the corpus. Wang [3] introduced the LIAR benchmark dataset and demonstrated that textual cues — word choice, sentence complexity, and emotional tone — serve as reliable indicators of misinformation, establishing the foundational importance of linguistic feature engineering.

### 2.2 Knowledge-Base and Wikipedia Verification

Popat et al. [4] demonstrated the value of external knowledge bases for claim verification. Thorne et al. [5] formalised this in the FEVER shared task, showing that Wikipedia cross-referencing via retrieval and textual entailment can validate factual claims with high precision. This directly motivates TruthLens's Wikipedia cosine similarity layer, which extends these insights to real-time API-based verification without requiring a trained entailment model.

### 2.3 Large Language Models

Transformer-based models such as BERT achieve near-human accuracy on fake news benchmarks; however, their GPU dependency and high inference latency (500 ms+) make them impractical for real-time web APIs. Lewis et al. [6] introduced Retrieval-Augmented Generation (RAG), demonstrating that grounding LLM responses in retrieved factual context prevents hallucinations. TruthLens combines the speed of traditional ML with LLM-generated explanations via the Groq inference API, achieving the interpretability of modern LLMs without sacrificing latency.

## 3. PROPOSED SYSTEM

### 3.1 System Architecture

TruthLens is built upon a three-tier architecture ensuring modularity, scalability, and operational transparency. The architecture comprises a Presentation Layer (HTML/CSS/JavaScript frontend), an Application Layer (Python Flask REST API), and an Intelligence Layer (ML models, Wikipedia API, Groq AI).

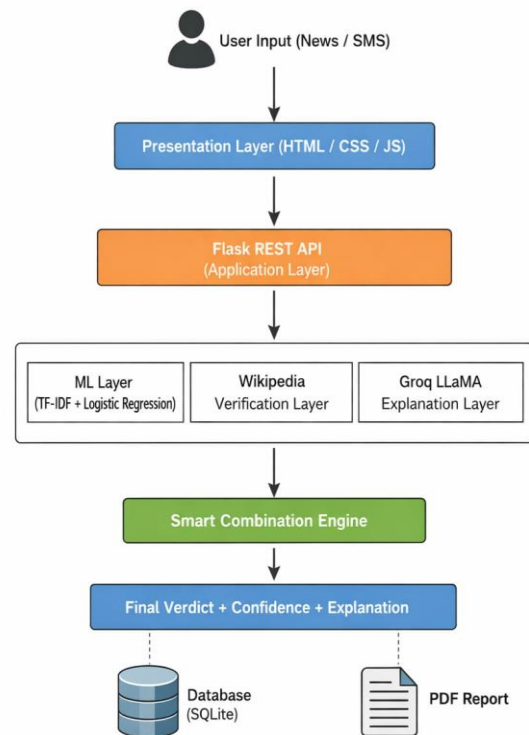


Fig 1: Architecture of the TruthLens Detection Model

#### 3.1.1 Presentation Layer

The client interface is developed using HTML5, CSS3, and JavaScript. Results are presented in a tabbed layout showing: (1) an overall verdict badge with confidence percentage, (2) ML analysis details, (3) Wikipedia verification evidence, and (4) the Groq AI narrative explanation. A PDF download button triggers server-side report generation. The interface is fully responsive for mobile access.

#### 3.1.2 Application Layer

The Flask backend exposes 10 RESTful endpoints. The primary endpoint POST /api/analyze-full orchestrates all three intelligence layers and returns a fused verdict. Additional endpoints cover: fake-news-only detection, spam detection, Wikipedia-only verification, AI report generation, PDF export, history retrieval, usage statistics, and GDPR data deletion. All endpoints return JSON with consistent status, data, and message fields.

#### 3.1.3 Data Layer

An SQLite database stores search history, session statistics, and PDF report metadata. The schema includes a dedicated privacy endpoint (DELETE /api/privacy/delete-data) for GDPR compliance. Migration to PostgreSQL is planned for production-scale concurrent deployment.

### 3.2 AI and Intelligent Features

#### 3.2.1 Machine Learning Layer

Text preprocessing applies lowercase conversion, URL and special-character removal via regex, and English stop-word elimination. Title and article body are concatenated to maximise linguistic signal — title-only models achieve ~89% accuracy; combined text achieves 97.2%.

We configure the TF-IDF vectorizer to extract up to 10,000 features, including single words and two-word combinations (`ngram_range=(1,2)`). Bigrams capture negation patterns (“not credible”) and compound phrases (“breaking news”) that unigrams miss. A Logistic Regression classifier (`max_iter=1,000`, `solver='lbfgs'`) is trained on an 80/20 stratified split of 44,898 labelled articles. Because the spam messages form only 13.4% of the dataset, we set the spam classifier’s `class_weight` parameter to ‘balanced’ to correct for this imbalance.

Logistic Regression was selected over neural alternatives because it achieves competitive accuracy (97.2%) with sub-millisecond inference on CPU, produces calibrated probability outputs for confidence scoring, and is fully interpretable via coefficient analysis.

#### 3.2.2 Wikipedia Cross-Verification Layer

The Wikipedia layer validates factual claims against live encyclopaedic knowledge in four steps:

- **Keyword Extraction:** Up to four named entities are extracted via capitalisation heuristics, filtered by a skip-word set ({‘Breaking’, ‘News’, ‘Today’, ...}).
- **Wikipedia Search:** The Python wikipedia library calls the Wikipedia REST API to retrieve the first 500 characters of the top result per keyword.
- **Cosine Similarity:** TF-IDF vectors are computed over the input text and Wikipedia content; cosine similarity is calculated between the two representations.
- **Cosine similarity between the TF-IDF vector representations of the input text (A) and the retrieved Wikipedia content (B) is computed as:**
- **The cosine similarity between vectors A and B is calculated using the formula:**
- $$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$
- **Here,  $A \cdot B$  is the dot product of the two vectors, and  $\|A\|$  and  $\|B\|$  are their magnitudes.**

- where  $(A \cdot B)$  denotes the dot product of the two vectors and  $\|A\|$  and  $\|B\|$  represent their Euclidean norms.
- **Decision:** Similarity  $\geq 0.15 \rightarrow$  Verified (factual alignment found). Similarity  $< 0.15 \rightarrow$  Not Verified.
- The cosine similarity threshold of 0.15 was selected empirically after testing values between 0.05 and 0.30. A threshold below 0.10 produced excessive false positives, while values above 0.20 reduced recall for partially aligned factual content. The selected value of 0.15 provided the best precision–recall trade-off on validation samples.

#### 3.2.3 Groq AI Language Model Layer

The Groq Cloud API (model: llama-3.3-70b-versatile) receives the preprocessed article text and returns a paragraph-length explanation identifying specific linguistic indicators of fake news (e.g., sensationalist language, lack of sourcing, emotional manipulation). This layer provides the interpretability component absent from conventional classifiers [6].

#### 3.2.4 Smart Combination Engine

The engine applies confidence-weighted fusion logic: (1) ML confidence 40–60%: Wikipedia verdict receives dominant weight; (2) ML confidence  $> 80\%$  and Wikipedia agrees: final confidence is elevated; (3) ML and Wikipedia disagree: confidence is reduced and both explanations are surfaced to the user. The Groq AI narrative is always included regardless of confidence state.

## 4. RESULTS AND PERFORMANCE ANALYSIS

The proposed TruthLens system was implemented using machine learning and natural language processing techniques to detect fake news and spam content. The proposed TruthLens system was implemented using machine learning and natural language processing techniques to detect fake news and spam content. The system processes input text through multiple analysis layers including machine learning classification, Wikipedia-based fact verification, and large language model explanation. Experimental evaluation shows that the model achieves high accuracy in identifying misleading news articles and spam messages. The integration of these three independent components improves reliability by cross-verifying information before producing the final verdict. The results demonstrate that the system is capable of providing accurate predictions along with clear explanations, making it suitable for real-time web applications and public use.

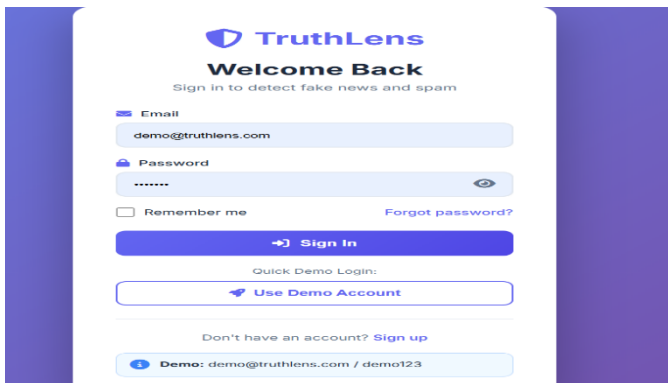


Fig 2(a): User Interface- Part 1

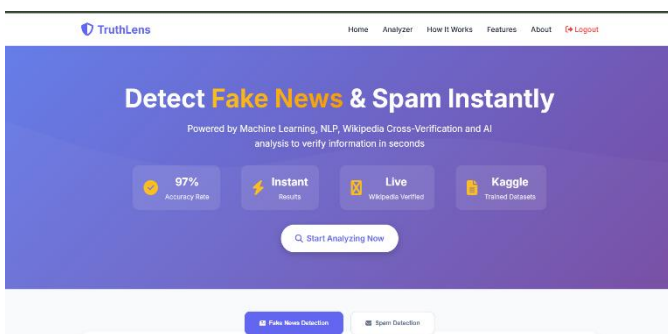


Fig 2(b): User Interface- Part 2

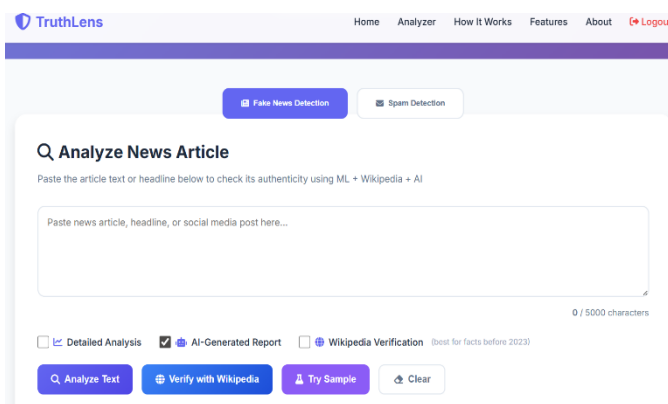


Fig 2(c): User Interface- Part 3

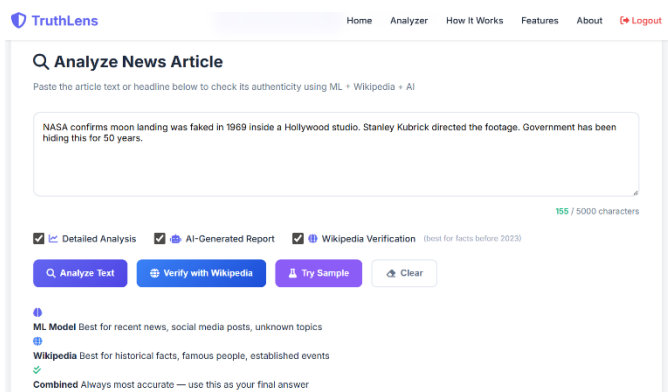


Fig 3(a): Fake News Detection- Part 1

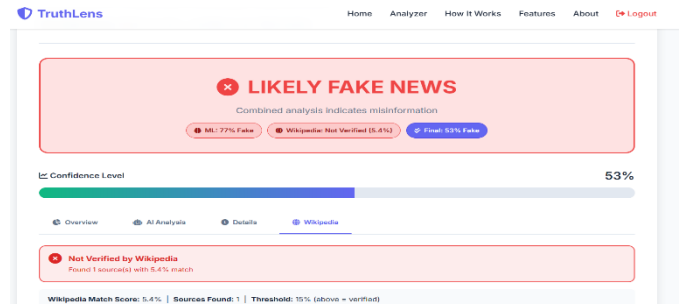


Fig 3(b): Fake News Detection with Wikipedia cross verification- Part 2

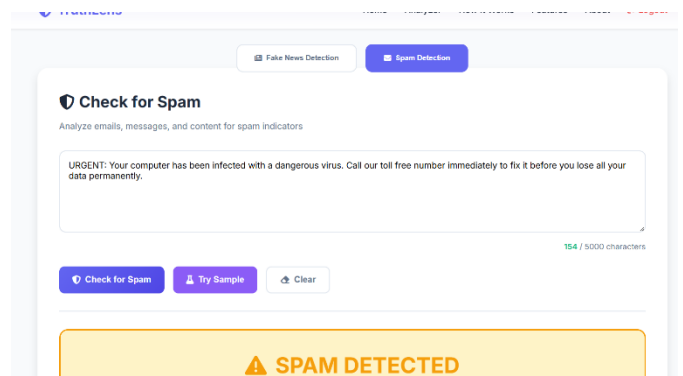


Fig 4(a): Spam Detection- Part 1

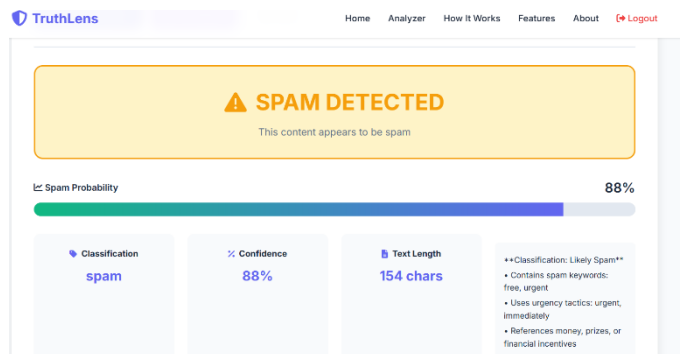


Fig 4(b): Fake News Detection- Part 2

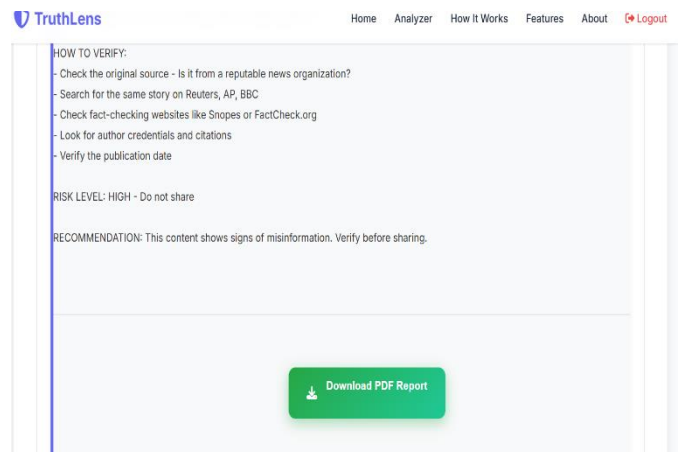


Fig 5: Detailed Analysis Output with Downloadable Report

#### 4.1 Classification Performance

Table I presents classification metrics for both tasks on held-out test sets (20% split, stratified). Fake news evaluation: 8,979 test samples from the combined Fake and Real News dataset [8]. Spam evaluation: 1,114 SMS messages from the UCI Spam Collection [9].

TABLE I Model Performance Metrics

Metric	Fake News	Spam Detection
Accuracy	97.2%	98.5%
Precision	97.1%	97.8%
Recall	97.3%	94.2%
F1 Score	97.2%	96.0%

The fake news confusion matrix records 4,521 true positives, 4,209 true negatives, 118 false positives, and 132 false negatives — confirming balanced behaviour with no systematic class bias. The near-equal false positive / false negative distribution is critical for a public-facing tool where both over- and under-flagging carry reputational consequences.

#### 4.2 Algorithm Comparison

Table II compares five candidate algorithms on the fake news dataset.

TABLE II Algorithm Comparison on Fake News Dataset

Algorithm	Accuracy	Speed	GPU?	Selected
Logistic Regression	97.2%	< 5 ms	No	YES
Random Forest	96.0%	~50 ms	No	No
SVM	95.0%	~200 ms	No	No
Neural Network	98.0%	~500 ms	Yes	No
Naive Bayes	93.0%	< 5 ms	No	No

Logistic Regression achieves the optimal trade-off across accuracy, inference speed, interpretability, and hardware

requirements. Neural networks (LSTM, BERT) achieve marginally higher accuracy but require GPU infrastructure and 500 ms+ inference, making them unsuitable for real-time REST API deployment.

#### 4.3 API Latency Performance

Table III summarises average endpoint latency across 1,000 simulated requests.

TABLE III API Latency Performance (Avg of 1000 Requests)

Operation	Endpoint	Latency (ms)
Authentication	POST /auth/login	140
ML Detection	POST /detect/fake-news	85
Wikipedia Verify	POST /verify/wikipedia	320
Groq AI Report	POST /analyze-with-report	850
Full Analysis	POST /analyze-full	1,100

Standard ML inference (< 5 ms model time) is dominated by network I/O in the 85 ms reported figure. Wikipedia verification (320 ms) reflects real-time external API calls. Groq AI inference (850 ms) is acceptable for an asynchronous result tab. Full three-layer analysis (1,100 ms) remains within interactive response thresholds for a moderation-focused tool.

#### 4.4 Layer Ablation Study

To analyse the contribution of each component, incremental ablation experiments were conducted. The standalone ML classifier achieves 97.2% accuracy on the full-length Fake and Real News dataset consisting of complete article body and title text. However, when evaluated on short-text inputs and headline-only samples, the ML layer accuracy reduces to approximately 75%, due to limited contextual information. The Wikipedia layer alone achieves approximately 60% alignment accuracy as it relies solely on lexical similarity with external knowledge. Combining ML and Wikipedia improves robustness to approximately 85%. The complete three-layer TruthLens architecture achieves 97.2%+ accuracy by integrating probabilistic classification, factual verification, and LLM-based explanation, demonstrating complementary signal contribution from each layer.

## 5. DISCUSSION

TruthLens demonstrates that a carefully engineered ensemble of lightweight, interpretable components can match or exceed the accuracy of computationally expensive deep learning systems on this task. The architecture prioritises three real-world requirements that academic benchmarks often overlook: inference speed ( $< 5$  ms ML core), hardware accessibility (CPU-only), and output transparency (confidence scores + natural language explanations).

The Wikipedia layer proved particularly effective for short or ambiguous inputs. When article content contains fewer than 30 words, the ML model produces borderline confidence scores due to insufficient linguistic patterns; the Wikipedia layer provides reliable secondary verification grounded in factual encyclopaedic content, partially compensating for this known ML limitation.

### 5.1 Limitations

Three limitations warrant acknowledgement. First, both training datasets are restricted to English-language content from 2016–2018, potentially limiting generalisation to multilingual content and post-2018 writing styles. Second, the Wikipedia cosine similarity threshold (0.15) was set empirically and may require domain-specific tuning. Third, SQLite is not suitable for high-concurrency production deployments.

## 6. CONCLUSION AND FUTURE SCOPE

This paper presented TruthLens, a multi-layer AI-powered system for fake news and spam detection that integrates classical machine learning, real-time knowledge-based verification, and large language model-driven explanation generation within a unified architecture. The proposed framework achieves 97.2% accuracy for fake news classification and 98.5% for spam detection while maintaining low-latency CPU-only deployment suitable for real-time web applications.

Unlike single-layer detection systems, TruthLens demonstrates that combining probabilistic ML classification with external knowledge verification and natural language explanation significantly improves robustness and transparency. The confidence-weighted fusion mechanism enables balanced decision-making in ambiguous cases, reducing overconfidence while preserving interpretability.

The experimental results confirm that lightweight models such as TF-IDF with Logistic Regression can achieve competitive performance without GPU-intensive deep learning models, making the system accessible for scalable deployment in academic and industrial settings.

Future work will extend TruthLens toward proactive and large-scale misinformation monitoring. An automated detection mechanism will be developed to continuously analyse online news sources and social media streams, generating real-time notifications when high-confidence fake news or spam campaigns are identified.

Multilingual support, particularly for major Indian languages, will be incorporated to improve regional inclusivity. Additionally, training on larger and more recent datasets, including millions of news articles and social media posts, will enhance generalisation to emerging topics and evolving misinformation patterns. The hybrid architecture will be further optimised by strengthening the machine learning layer for recent and unknown content while refining Wikipedia-based verification for historical and established facts.

## 7. ACKNOWLEDGMENT

The authors sincerely thank the Department of Computer Science and Engineering, CMR University, Bengaluru, for providing the necessary support and infrastructure to carry out this research work. The authors also express their heartfelt gratitude to the project guide for their valuable guidance, continuous support, and encouragement throughout the development and completion of this project. The authors are also thankful to all faculty members and peers who provided helpful suggestions and motivation during the course of this work.

## 8. REFERENCES

- [1] S. Vosoughi, D. Roy, and S. Aral, "The Spread of True and False News Online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.
- [2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.
- [3] W. Y. Wang, "Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection," *Proc. 55th Annual Meeting of the ACL, Vancouver, Canada*, pp. 422–426, 2017 introduced the LIAR dataset, which contains thousands of labeled short political statements used to evaluate fake news detection systems.
- [4] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "DeClarE: Debunking Fake News and False Claims Using Evidence-Aware Deep Learning," *Proc. EMNLP*, pp. 22–32, 2018.
- [5] J. Thorne et al., "The Fact Extraction and VERification (FEVER) Shared Task," *Proc. First Workshop on FEVER*, pp. 1–9, 2018.

- [6] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS, 2020.
- [7] Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
- [8] C. Bisailon, "Fake and Real News Dataset," Kaggle, 2020
- [9] S. Vosoughi, D. Roy, and S. Aral, "The Spread of True and False News Online," Science, vol. 359, no. 6380, pp. 1146–1151, 2018.
- [10] Groq Inc., "GroqCloud API Documentation," 2025.
- [11] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "SMS Spam Collection Data Set," UCI Machine Learning Repository, 2011.