IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

TRUTHLENS: AN EXPLAINABLE AI APPROACH FOR FAKE NEWS DETECTION

Niharika Saxena, Priyanka Yadav
Under the Guidance of: Er. Ratan Rajan Srivastava (Assistant Professor)
Coordinator: M.B Singh
Department of Computer Science & Engineering
Shri Ramswaroop Memorial College of Engineering and Management, Lucknow, India

Abstract: The rapid proliferation of misinformation and fake news across digital media platforms poses a serious threat to public discourse, democratic processes, and societal well-being. Existing automated detection systems predominantly operate as "black-box" models, offering classifications without human-understandable justification. This paper presents TruthLens, a Flask-based web application that integrates machine learning with Explainable Artificial Intelligence (XAI) techniques for transparent and interpretable fake news detection. The system employs a Logistic Regression classifier trained on TF-IDF features extracted from the ISOT Fake News Dataset, achieving an accuracy of 98.67%. Explainability is provided through LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations), which highlight the influential features driving each prediction. The pipeline further incorporates Named Entity Recognition (NER) using spaCy, real-time external fact-checking via the GNews API with cosine similarity scoring, and a hybrid decision fusion mechanism combining ML confidence scores with semantic similarity. Additional features include URL-based article analysis, trust scoring for news sources, user authentication, prediction history tracking, and downloadable PDF reports. The system is designed to be transparent, user-centric, and deployable in real-world misinformation detection scenarios.

Index Terms - Fake News Detection, Explainable AI, LIME, SHAP, TF-IDF, Logistic Regression, Named Entity Recognition, Fact Checking, Natural Language Processing, Flask.

I. INTRODUCTION

The digital information era has witnessed an unprecedented rise in fake news, disinformation, and propaganda, particularly amplified through social media platforms. Misinformation related to health crises, elections, and geopolitical events can cause measurable harm to individuals and communities. The COVID-19 pandemic exemplified how health-related false information can be as dangerous as the disease itself. Automated fake news detection systems have emerged as a promising solution, but their widespread adoption is hindered by a fundamental challenge: lack of interpretability.

Most state-of-the-art deep learning models for fake news detection are opaque in nature, providing classification results without any justification. This "black-box" behavior erodes user trust and limits practical deployment in high-stakes environments where accountability and transparency are essential. Explainable Artificial Intelligence (XAI) addresses this gap by enabling models to communicate the rationale behind their decisions in human-understandable terms.

TruthLens is a comprehensive, explainable fake news detection system that combines classical machine learning with XAI methodologies. Built as a web application using the Flask framework, TruthLens enables users to submit either raw news text or a URL. The system processes the input through a multi-

stage pipeline: text preprocessing, TF-IDF vectorization, Logistic Regression classification, NER-based entity extraction, external fact verification via GNews API, and hybrid confidence scoring. LIME and SHAP explainers then generate word-level justifications to clarify why the model predicted a given label. The result is a human-friendly, transparent, and actionable fake news verdict.

The primary contributions of this work are: (i) a high-accuracy fake news classifier (98.67%) using TF-IDF + Logistic Regression; (ii) integration of LIME and SHAP for local and global feature-level explainability; (iii) real-time external fact-checking using GNews API with semantic similarity; (iv) NER-based entity recognition to identify named entities in news articles; (v) hybrid decision fusion combining ML and fact-check scores; and (vi) a deployable web interface with user management, history tracking, and PDF report generation.

II. RELATED WORK

Automatic fake news detection has been studied extensively in recent years. Ahmed et al. [1] proposed a machine learning approach using N-gram language models and TF-IDF features with Support Vector Machines (SVM) and Naive Bayes classifiers, achieving up to 92% accuracy on the ISOT dataset. However, their work lacked any mechanism for explaining model decisions to end users.

Rashkin et al. [2] explored LSTM-based models for credibility analysis using the LIAR dataset, finding that linguistic cues such as hedging language and exaggeration patterns are indicative of fake content. Despite their linguistic depth, these models operated without transparency. Reis et al. [3] experimented with ensemble methods including Random Forests and Gradient Boosting on social-context features, noting the importance of metadata but again without interpretability provisions.

Popat et al. [4] introduced DeClarE, which combined LSTM networks with attention mechanisms and external evidence from the web for claim verification. Their attention-based interpretability was a step forward, but limited compared to the comprehensive XAI frameworks offered by LIME and SHAP. Lundberg and Lee [5] formalized SHAP as a unified framework for feature attribution rooted in cooperative game theory, while Ribeiro et al. [6] introduced LIME as a model-agnostic approach to generate local interpretable approximations.

TruthLens builds on these foundations by combining the simplicity and efficiency of Logistic Regression with the post-hoc interpretability power of both LIME and SHAP, while also incorporating real-time fact-checking that is absent from most prior work. Table 1 summarizes the comparison with related works.

Table 1: Comparison with Related Works

Reference	Model Used	Dataset	Accuracy	XAI Method
Ahmed et al. [1]	SVM, NB	LIAR, ISOT	88.6%	None
Rashkin et al. [2]	LSTM	LIAR	27.7%	None
Reis et al. [3]	Random Forest	BuzzFeed	76.4%	None
Popat et al. [4]	DeClarE (LSTM+Attention)	Multi-source	85.2%	Attention
TruthLens (Proposed)	LR + TF-IDF	ISOT	98.67%	LIME + SHAP

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The TruthLens architecture consists of six core modules that work in a sequential pipeline. Figure 1 illustrates the overall system flow.

A. Text Preprocessing Module

Input text undergoes a standardized cleaning process implemented in preprocess.py. Using the NLTK library, the text is converted to lowercase, non-alphabetic characters are removed, English stopwords are filtered, and Porter Stemming is applied to normalize word forms. This preprocessing ensures that the TF-IDF vectorizer operates on a clean, noise-free token space. For URL-based inputs, the application uses the BeautifulSoup library to extract article text from the webpage before applying the same pipeline.

B. Feature Extraction and Classification

Text features are extracted using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. The TF-IDF model was trained on the ISOT Fake News Dataset, a widely-used benchmark comprising 21,417 real news articles and 23,481 fake news articles. The serialized TF-IDF vectorizer (tfidf_vectorizer.pkl, approximately 52 MB) captures vocabulary from both classes. Three classifiers were trained and evaluated: Logistic Regression (LR), Support Vector Machine (SVM), and Passive Aggressive Classifier (PAC).

The Logistic Regression model demonstrated the best performance with 98.67% accuracy and was selected as the primary classifier for the production system. The classifier outputs both a binary prediction (REAL/FAKE) and class probability scores, which are used downstream for hybrid decision scoring.

C. Explainability Layer (LIME + SHAP)

The explainability module (explain.py and shap_explain.py) generates post-hoc explanations using two complementary XAI frameworks. LIME (Local Interpretable Model-agnostic Explanations) creates a locally faithful linear approximation around the input instance by perturbing the input text and observing how the model's predictions change. The top 6 most influential words are extracted along with their directional impact (toward REAL or FAKE classification).

SHAP (SHapley Additive exPlanations) computes feature attributions based on Shapley values from cooperative game theory, ensuring consistency and local accuracy. The LinearExplainer is used with a small background corpus representing general news topics. The top 8 features with non-zero SHAP values are presented to the user with directional labels. Together, LIME and SHAP provide complementary perspectives: LIME highlights locally important words while SHAP provides a global feature attribution baseline.

D. Named Entity Recognition (NER)

The ner_utils.py module employs spaCy's en_core_web_sm model to extract named entities (persons, organizations, locations, dates, etc.) from the input text. This serves a dual purpose: enriching the user's understanding of the news article and building a smarter query for the fact-checking module. Instead of using the full input text as a search query, TruthLens constructs a targeted query using only the extracted entity names, significantly improving the relevance of retrieved articles from the GNews API.

E. Real-time Fact Verification

The fact_checker.py module implements an automated fact-checking pipeline using the GNews API. The NER-derived entity query is used to fetch up to 10 recent news articles from trusted sources. A cosine similarity score is computed between the TF-IDF representation of the input text and those of the retrieved articles. A similarity score above 0.5 indicates substantial corroboration from external sources, contributing to a "verified real" classification.

The module handles three status outcomes: (i) success — articles retrieved and similarity computed; (ii) no_data — no relevant articles found, indicating unverifiable or novel content; and (iii) api_failed — network errors or API rate limits, triggering fallback to ML-only prediction.

F. Hybrid Decision Fusion

TruthLens employs a weighted hybrid decision mechanism to produce the final prediction. When the fact-checking module returns a success status, the final score is computed as:

Final Score = (ML Confidence × 0.6) + (Fact Similarity × 0.4)

If the final score exceeds 0.5, the article is classified as REAL NEWS (Verified); otherwise, it is classified as FAKE NEWS (Low verification). When fact-checking is unavailable, the system falls back to the ML-only prediction. An additional domain-level trust score is assigned to URL inputs based on keyword matching against known reliable sources (BBC, CNN, Reuters, NDTV, etc.).

IV. IMPLEMENTATION

A. Technology Stack

TruthLens is implemented using Python 3.x with Flask as the web framework. The key libraries used include: scikit-learn for ML model training and TF-IDF vectorization; NLTK for text preprocessing; spaCy (en_core_web_sm) for NER; LIME and SHAP for explainability; Matplotlib for generating bar and pie charts; ReportLab for PDF report generation; SQLite3 for user and prediction history storage; BeautifulSoup4 for web scraping; and the GNews API for live news retrieval. The application is deployed using Gunicorn and configured for hosting on Render (cloud platform) via a Procfile.

B. Dataset

The system was trained on the ISOT Fake News Dataset [7], collected from the Reuters news agency (real news) and various identified fake news websites. The dataset contains 44,898 articles split into two categories. The data was preprocessed and vectorized using TF-IDF with a vocabulary learned across both classes. An 80:20 train-test split was used for model evaluation.

C. User Interface and Features

The web interface provides six main pages: Home (index), Register, Login, Predict, History, and Admin. On the Predict page, users can enter either plain text or a URL. After submission, the system displays: the raw model prediction with confidence percentage; the final hybrid prediction; LIME word-level explanations; SHAP feature attributions; NER-extracted entities; fact-check results with similarity score; matched external news articles; a human-readable conclusion; and the website trust score for URL inputs. Users can download their prediction history as a PDF report. The Admin dashboard displays all user predictions with real vs. fake news distribution charts.

D. Authentication and Security

User authentication is handled via SQLite3 with bcrypt-hashed passwords. Password strength validation enforces a minimum of 8 characters including uppercase, lowercase, digits, and special characters. Session management is handled by Flask's server-side sessions with a secret key loaded from environment variables via the python-dotenv library. An admin account with privileged dashboard access is automatically created on first launch.

V. RESULTS AND DISCUSSION

The three trained classifiers were evaluated on a held-out test set from the ISOT dataset. Table 2 presents the performance comparison.

Table 2: Model Performance Comparison

Model	Accuracy	Precision	Recall / F1
Logistic Regression (TF-IDF)	98.67%	0.99 (Real), 0.98 (Fake)	0.99 / 0.99
SVM (TF-IDF)	98.23%	0.98 (Real), 0.98 (Fake)	0.98 / 0.98
PAC (TF-IDF)	97.85%	0.97 (Real), 0.98 (Fake)	0.98 / 0.97

Logistic Regression with TF-IDF achieves the highest accuracy of 98.67%, outperforming the SVM (98.23%) and PAC (97.85%) classifiers. The near-perfect precision and recall values indicate that the model generalizes extremely well on the ISOT dataset with minimal false positives and false negatives. This result is consistent with prior literature that demonstrates the effectiveness of TF-IDF features for stylistic fake news classification.

The LIME explanations demonstrate that words with high positive weights such as "trump", "reuters", "official", and "government" tend to push predictions toward REAL, while words like "hoax", "viral", "shocking", and certain sensationalist language patterns increase the FAKE probability. SHAP values confirm these patterns and additionally reveal feature interactions that LIME alone may miss.

The real-time fact-checking module further enhanced the system's reliability. In live testing with current news URLs from NDTV, Reuters, and BBC, the GNews-based similarity mechanism correctly corroborated genuine articles (similarity > 0.7) while assigning low similarity scores (< 0.3) to fabricated test inputs. The hybrid fusion mechanism thus produced a nuanced, multi-signal verdict rather than relying solely on the ML model's learned statistical patterns.

The NER component correctly identified named entities including persons, organizations, and locations in all test cases. This significantly improved the quality of GNews search queries by providing focused, entity-centric search terms rather than noisy full-text queries.

VI. CONCLUSION

This paper presented TruthLens, a transparent and explainable fake news detection system that addresses the critical need for interpretability in AI-powered misinformation detection. By combining a high-accuracy TF-IDF + Logistic Regression classifier (98.67%) with LIME and SHAP explainability, real-time GNews fact-checking, spaCy-based NER, and a hybrid decision fusion mechanism, TruthLens provides users with not just a classification verdict but a comprehensive, multi-perspective rationale.

The system demonstrates that classical machine learning, when paired with modern XAI techniques and real-time external verification, can match or exceed the performance of more complex deep learning alternatives while remaining transparent, computationally efficient, and deployable on standard web infrastructure. Future work will explore fine-tuned transformer models (e.g., BERT) as the base classifier, multilingual support, integration with additional fact-checking APIs, and deeper social context analysis including author credibility and publishing domain reputation.

ACKNOWLEDGMENT

The authors would like to thank the institution for providing computational resources and guidance. Special thanks to the open-source communities behind scikit-learn, LIME, SHAP, spaCy, and Flask that made this work possible.

REFERENCES

[1] Ahmed, H., Traore, I., & Saad, S. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, pp. 127-138. Springer, Cham.

[2] Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017). Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2931-2937.

[3] Reis, J. C. S., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. (2019). Supervised Learning for Fake News Detection. IEEE Intelligent Systems, 34(2), 76-81.

[4] Popat, K., Mukherjee, S., Yates, A., & Weikum, G. (2018). DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In Proceedings of EMNLP 2018, pp. 22-32.

[5] Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems (NeurIPS), 30, pp. 4765-4774.

[6] 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, pp. 1135-1144.

[7] Ahmed, H., Traore, I., & Saad, S. (2018). Detecting Opinion Spams and Fake News Using Text Classification. Security and Privacy, 1(1), e9. (ISOT Fake News Dataset).

[8] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake News Detection on Social Media: A Data Mining Perspective. ACM SIGKDD Explorations Newsletter, 19(1), 22-36.

[9] Thorne, J., & Vlachos, A. (2018). Automated Fact Checking: Task Formulations, Methods and Future Directions. In Proceedings of the 27th International Conference on Computational Linguistics (COLING), pp. 3346-3359.

[10] Honnibal, M., & Montani, I. (2017). spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing. To appear.