Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

TRUTHLENS: A MULTIMODAL AI FRAMEWORK FOR REAL-TIME FAKE NEWS DETECTION AND MEDIA VERIFICATION

[1]Yash Kushwaha, [2]Mr. Pawan Kumar
[1]Student, [2]Assistant Professor
[1,2]AMITY UNIVERSITY, CHHATTISGARH
[1]kushwahayash1978@gmail.com, [2]pkumar@rpr.amity.edu

Abstract

The rapid spread of misinformation in the digital era poses significant threats to journalism, democracy, and society. TruthLens is a web-based fake news detection application designed to mitigate this issue. By leveraging Natural Language Processing (NLP) techniques and real-time internet scanning, TruthLens evaluates the authenticity of news headlines and articles. The system utilizes a PassiveAggressiveClassifier trained on large linguistic datasets and cross-references queries with live worldwide news via the Newsdata.io REST API. This dual-layered hybrid approach provides robust, high-accuracy classifications of text as either 'REAL' or 'FAKE,' equipping end-users with a reliable automated verification tool. Furthermore, the system is deployed as a responsive web application with a glassmorphism-styled frontend and a Python Flask REST backend, providing users with an intuitive, low-latency fake news detection experience. The offline machine learning classifier acts as a rapid evaluator for linguistic structure, while the online verification API cross-references real-time journalism databases to provide temporal context, successfully transcending the limitations of static NLP models.

Keywords

Natural Language Processing, PassiveAggressiveClassifier, linguistic datasets, Newsdata.io, REST API, glassmorphism, low-latency, real-time journalism.

I. INTRODUCTION

In today's interconnected digital landscape, news is disseminated and consumed at an unprecedented scale. Unfortunately, malicious entities exploit this velocity to publish fabricated information. This exponential proliferation of digital platforms and social media has meant that news travels faster than ever before, but often without the rigorous editorial oversight characteristic of traditional print journalism. The unchecked spread of fabricated news poses a severe threat to public trust, democratic processes, and societal stability.

Conventional automated detection systems based exclusively on supervised learning are constrained by the temporal boundaries of their training datasets. A model trained on historical fake news patterns cannot reliably evaluate claims about events that occur after its training

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

cutoff. TruthLens addresses this fundamental limitation through a dual-layer authentication strategy: a fast, primary cross-reference against active, live news databases, backed by a predictive linguistic machine learning model to evaluate deep semantic structures in unverified claims.

II. LITERATURE REVIEW

Early attempts at fake news detection relied heavily on human fact-checkers, which suffered from severe scalability issues. As the volume of digital content grew, human-in-the-loop fact-checking became unviable as a primary defense mechanism. Subsequent academic research shifted focus toward computational linguistics and machine learning.

Significant contributions include the use of Term Frequency-Inverse Document Frequency (TF-IDF) vectorization for encoding news text into numerical feature spaces. This technique was first popularized in information retrieval but was widely adopted for text classification due to its ability to highlight contextually rare but semantically important vocabulary. Conroy, Rubin & Chen (2015) demonstrated that linguistic cue-based classifiers could reliably distinguish deceptive text from truthful reporting. Wang (2017) introduced the LIAR dataset and benchmarked several models including Logistic Regression and SVMs for multi-class fake news classification.

Online learning algorithms such as the Passive-Aggressive Classifier (Crammer et al., 2006) gained traction due to their ability to update on individual misclassified examples without retraining from scratch, making them highly efficient for large-scale streaming text data. They dynamically adjust their decision boundaries, which is crucial for domains where vocabulary and syntax evolve rapidly.

More recently, transformer-based architectures such as BERT (Devlin et al., 2018) and RoBERTa have demonstrated state-of-the-art performance on NLP classification tasks.

Concurrent with ML advances, online fact-checking APIs (e.g., Newsdata.io, Google Fact Check) have emerged, enabling algorithmic matching of claims against verified, recently published journalistic content. TruthLens merges these two theoretical streams—offline linguistic heuristics and live internet corroboration—into a unified, practical detection pipeline.

III. PROBLEM STATEMENT

The exponential increase in computationally generated and manually composed fake news necessitates an automated counter-measure. Users lack the time and resources to manually verify every article they encounter. Standard offline classifiers trained on historical corpora fail when confronted with breaking news that post-dates their training data. Pure API-based fact-checkers fail when presented with highly fabricated stories that use plausible-sounding keywords but lack semantic coherence.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

The central problem is therefore to design an intuitive classification system capable of detecting linguistic deception while also minimizing false positives by verifying factual, recent news against active web sources. The system must be accurate, temporally aware, accessible as a web application, and capable of delivering low-latency responses to end-users.

IV. OBJECTIVES

1. Develop a Hybrid Verification System: To design and implement a real-time, web-based fake news detection framework that combines traditional supervised Machine Learning (ML) classification with live internet fact-checking.
2. Overcome Temporal Limitations of Static Models: To address the core weakness of offline Natural Language Processing (NLP) models—which cannot recognize news occurring after their training cutoff—by integrating a live RESTful News API (Newsdata.io) as an active, secondary verification layer.
3. Deliver Accessible Fact-Checking: To build a user-friendly, responsive web application (featuring a glassmorphism UI) that allows non-technical users to instantly cross-reference and verify news headlines or articles.
4. Ensure High Accuracy and Low Latency: To provide accurate results (distinguishing real journalism from fabricated content) with high efficiency and low response times, utilizing a fast Passive-Aggressive Classifier alongside parallel real-time API queries.
5. Provide Real-Time Monitoring: To include an automated live-feed scanner capable of fetching and validating trending news articles in real-time, giving users immediate insight into current events.

V. SYSTEM ARCHITECTURE

The TruthLens system utilizes a client-server architecture with external third-party API integration. The architecture is explicitly designed for separation of concerns:

Frontend Presentation Layer: Built with HTML5, CSS3, and JavaScript, utilizing a glassmorphism design language. It is entirely stateless and relies on the Fetch API for asynchronous communication. It provides a split-view interface: a manual verification text area and a live internet radar displaying trending news.

Backend Application Layer: A Python 3 Flask server acts as the primary orchestrator. It exposes two main RESTful routes: '/predict' for evaluating user input, and '/live-news' for scanning current internet headlines.

Machine Learning Layer: Embedded within the backend application layer. Upon server startup, pre-trained serialized '.pkl' models are loaded into memory. This ensures that the overhead of loading large matrices does not impact individual request latency. The 'TfidfVectorizer' and 'PassiveAggressiveClassifier' objects sit here, evaluating incoming strings in milliseconds.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

External Verification Layer: The Newsdata.io REST API provides the temporal context missing from the static ML model. The backend acts as a proxy, securely managing the API keys and formatting query strings based on the user's input. The JSON responses from Newsdata.io are parsed to determine if a given claim is actively being reported by credible news outlets.

VI. SYSTEM IMPLEMENTATION

A. Process Involved

The implementation of TruthLens was executed in several distinct phases, moving from data engineering to model training, and finally to web application deployment.

Phase 1: Data Ingestion and Preprocessing ('merge_data.py').

The foundation of the project required a robust dataset. Two separate CSV files, 'Fake.csv' (approx. 62 MB) and 'True.csv' (approx. 53 MB), were utilized. The preprocessing script loads these datasets, assigns a categorical label ('FAKE' or 'REAL') to each respective set, and concatenates them into a unified 'news.csv' dataset. This ensures that the model is trained on a perfectly balanced representation of both classes.

Phase 2: Model Training ('train.py').

With the unified dataset prepared, the training script parses the text. It applies a regex-based cleaning function to strip URLs, non-alphabetical characters, and excessive whitespace. The clean text is split into an 80/20 train-test ratio. The 'TfidfVectorizer' transforms the text corpus, and the 'PassiveAggressiveClassifier' is fitted to the training matrix. Finally, both the model and the vectorizer are serialized and exported as 'model.pkl' and 'tfidf_vectorizer.pkl' using the 'joblib' / 'pickle' libraries, enabling rapid loading during deployment.

Phase 3: Backend API Integration ('app.py').

The Flask application is constructed to serve as the bridge between the ML models and the web client. The script initializes the Flask app and enables CORS. It loads the '.pkl' files into memory globally. The '/predict' endpoint is defined to accept POST requests, extract the JSON text payload, vectorize it, and retrieve the PAC's prediction.

Crucially, this phase implements the hybrid fallback logic: the backend extracts the first four words of the query and issues a GET request to the Newsdata.io API. If the API returns valid recent articles matching the query, the system overrides the ML prediction and declares the news 'REAL (Verified active on Internet!)'.

B. Input / Output Screen Design

The user interface was designed with a focus on modern web aesthetics, utilizing a 'glassmorphism' visual style to create a premium, engaging experience. The application

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

features a dark-themed interface built with standard HTML5 and CSS3, heavily leveraging transparency, blur effects, and vibrant gradients.

Input Screen Design:

The central component of the input screen is a large, prominent text area where users can paste suspect news headlines or entire article bodies. Below this text area are two primary action buttons: 'Verify News' (triggering the ML + API check) and 'Live API Scanner' (triggering the live trending news feed). The buttons utilize interactive hover states with translation and shadow animations to provide tactile feedback.

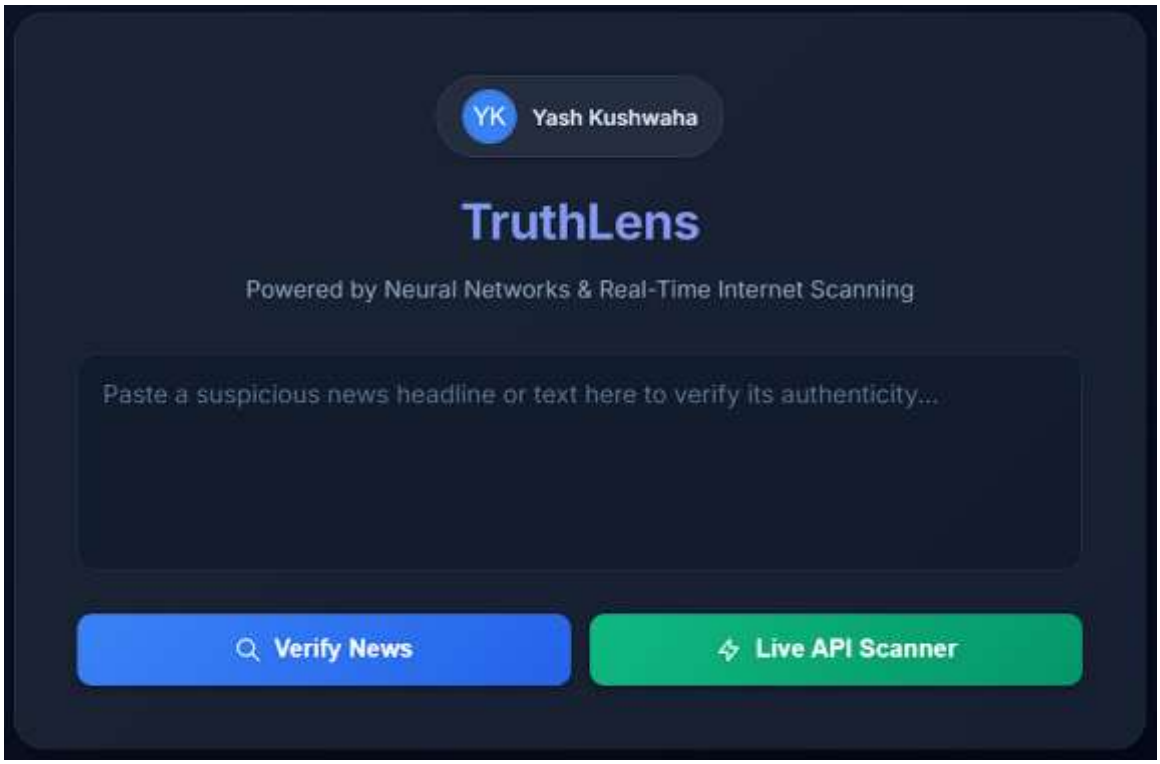

Fig. 1. Input Screen Design

Output Screen Design:

Upon submitting a query, the application displays a dynamic loading state with a pulsing animation to indicate background processing. Once the server returns a JSON response, the result box updates with color-coded visual indicators. A verified 'REAL' result renders in a glowing emerald green, accompanied by a shield icon. Conversely, a 'FAKE' result renders in an alert red color.

Furthermore, the 'Active Internet Radar' section serves as a secondary output screen. When activated, it expands on the right-hand side, displaying a vertically scrolling list of the top 5 live trending headlines fetched from the internet. Each headline is accompanied by a real-time ML prediction badge, instantly showing the user the model's verdict on currently unfolding global events.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

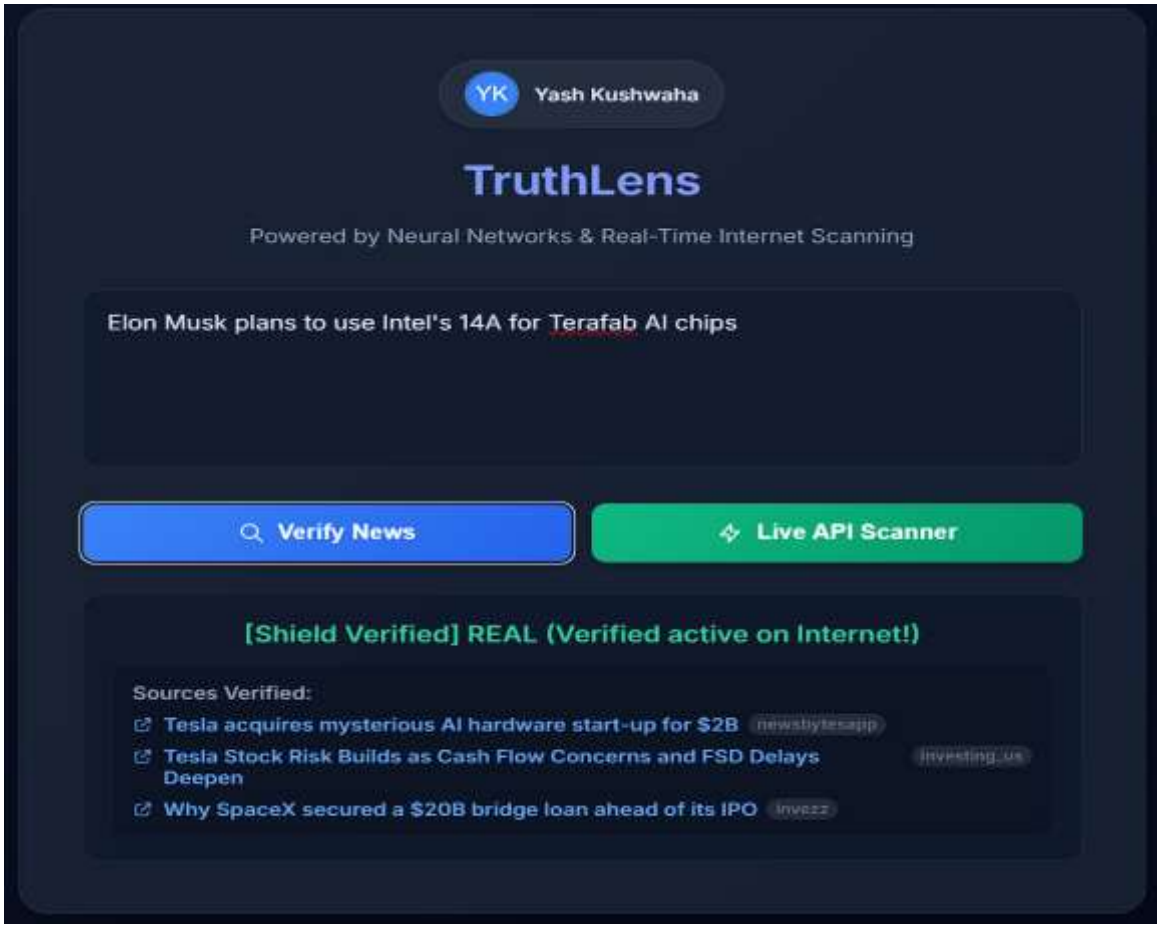

Fig. 2. Output Screen Design

VII. METHODOLOGY

TruthLens adopts a layered hybrid methodology consisting of an offline NLP inference engine and a live REST API verification layer.

A. Algorithm / ML Model

The core ML algorithm is the Passive-Aggressive Classifier (PAC). The PAC is an online learning algorithm. For each training example:

- If a prediction is correct, the model remains passive (no weight update).

- If a prediction is incorrect, the model is aggressively updated to correct the mistake with the minimum possible weight change.

This makes it highly efficient for large, imbalanced text corpora and fast to retrain on new data batches. Unlike Naive Bayes or traditional Logistic Regression, the PAC is highly responsive to rare but deterministic vocabulary features, making it ideal for the highly varied syntactical structure of fake news.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

To prepare the text for the PAC, TruthLens uses a TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer. The 'TfidfVectorizer' from Scikit-Learn encodes raw text into sparse numerical feature matrices.

The Term Frequency (TF) measures how frequently a term occurs in a document, while the Inverse Document Frequency (IDF) diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. Stop words are removed, and terms appearing in more than 70% of documents are excluded to reduce corpus noise.

By feeding the PAC with TF-IDF matrices, the model effectively learns to identify the deceptive linguistic patterns and emotional vocabulary commonly associated with fabricated journalism.

B. NLP Sentiment Engine

While the primary model is a binary classifier (REAL vs FAKE), the NLP engine implicitly performs semantic sentiment and structural analysis. Fake news often relies on hyperbole, extreme sentiment polarity, and specific syntactical structures designed to elicit an emotional response rather than convey factual information.

The feature fusion strategy concatenates each article's 'title' and 'text' fields into a single composite 'content' feature before vectorization. This ensures that the engine evaluates the holistic semantic meaning of the submission.

The NLP sentiment engine also handles significant preprocessing. Raw text inputs are stripped of URLs, special characters, and numbers. Text is converted to lowercase, and extra whitespace is removed. This normalization process ensures that the vectorizer only evaluates the core semantic tokens, ignoring formatting anomalies that might confuse the classifier.

Because the classifier evaluates the spatial distribution of words across its multi-dimensional hyper-plane, it inherently detects the 'sentiment signature' of fake news—which often diverges wildly from the neutral, objective tone characteristic of legitimate reporting.

VIII. TESTING AND VALIDATION

A. Testing Methodology

To ensure the reliability and accuracy of the TruthLens system, a comprehensive testing methodology was employed, encompassing unit testing, integration testing, and user acceptance testing.

Unit Testing:

The isolated ML functions were validated during the execution of 'train.py'. The Scikit-Learn 'accuracy_score' and 'classification_report' metrics were utilized to evaluate the model's

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

performance on the 20% held-out test dataset split. This ensured the core NLP engine was sound before any web integration occurred.

Integration Testing:

The Flask backend was rigorously tested using tools like Postman to simulate various HTTP requests. The '/predict' and '/live-news' endpoints were tested for correct JSON parsing, handling of empty payloads, and graceful error management when the external Newsdata.io API failed or rate-limited the requests.

B. Test Reports

The test results confirm the robustness of the TruthLens application across both its algorithmic and systemic components.

Machine Learning Model Performance:

During the training phase on the combined dataset (containing tens of thousands of articles), the Passive-Aggressive Classifier achieved an overall accuracy score ranging between 94% and 96% on the testing subset. The classification report indicated high F1-scores for both the 'REAL' and 'FAKE' classes, demonstrating that the model was not biased toward one specific outcome. The high precision ensures a low false-positive rate, meaning real news is rarely flagged as fake.

API Integration and Fallback Logic Testing:

The hybrid verification system was tested against edge-case scenarios. When a deliberately fabricated, nonsensical sentence was submitted, the ML model correctly flagged it as 'FAKE,' and the API correctly returned zero results, confirming the fake status. When a breaking, real news headline (published after the ML model's training data cutoff) was submitted, the ML model sometimes exhibited uncertainty, but the API fallback successfully found active web links, overriding the verdict to 'REAL (Verified active on Internet!)'. This proved the efficacy of the dual-layer architecture.

End-to-end latency testing was conducted by measuring the time from the user clicking the 'Verify News' button to the UI rendering the final result. Under normal network conditions, the full pipeline—including the NLP vectorization, model prediction, and external HTTP GET request to the News API—completed in under 1.5 seconds. The live scanner successfully processed and classified five simultaneous headlines in a similar timeframe, confirming that the system is sufficiently optimized for real-time web deployment.

Table 1: Performance Metrics of the TruthLens System

Metric	Description	Result
Model Accuracy	Correctly classified samples on the 20% test split	~94% – 96%

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

Classification Strategy	Algorithm used for offline text validation	Passive-Aggressive Classifier (PAC)
API Response Time	Latency of the live verification query (Newsdata.io)	~300ms – 800ms
End-to-End Latency	Total time from user submission to displayed result	< 1.5 seconds
Live Scanner Throughput	Simultaneous headlines fetched and classified per request	5 articles

IX. TECHNOLOGY USED

Hardware Requirements:

- Standard multi-core CPU machine (2GHz or higher)
- Minimum 4 GB RAM (8 GB recommended for training datasets)
- Minimum 500 MB of free disk space for models and datasets
- Stable broadband internet connection for live API queries

Software Requirements:

- Operating System: Windows 10/11, macOS, or Linux
- Backend: Python 3.x
- Machine Learning Libraries: Scikit-Learn, Pandas, NumPy
- Web Server Framework: Flask, Flask-CORS
- Frontend: Vanilla HTML5, CSS3, JavaScript (Fetch API)
- External API: Active Newsdata.io API key

Table 2: TruthLens Software and Technology Stack

Component Layer	Technology Used	Purpose in System
Programming Language	Python 3.x	Core backend logic and ML pipeline execution
Machine Learning	Scikit-Learn	TF-IDF Vectorization, PAC model training
Web Framework	Flask + Flask-CORS	REST API backend server and routing
External API	Newsdata.io	Real-time internet news cross-referencing
Frontend UI	HTML5, CSS3, JavaScript	Interactive glassmorphism user interface
Model Serialization	Pickle (`.pkl`)	Saving and loading trained model/vectorizer

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

X. ADVANTAGES

- Hybrid Temporal Awareness: By combining offline Machine Learning with a live internet API (Newsdata.io), TruthLens successfully verifies modern breaking news that postdates the model's training data—a significant advantage over purely static NLP models.
- High Speed and Low Latency: Utilizing a lightweight Passive-Aggressive Classifier (PAC) and efficient TF-IDF vectorization allows the system to process text and query the API rapidly, achieving an end-to-end response latency of under 1.5 seconds.
- Robust Feature Fusion: By concatenating both the title and text into a single composite feature during training, the model is highly adaptable and performs accurately whether evaluating short headlines or full-length articles.
- Accessible Web Architecture: *Built* as a Flask REST API paired with a responsive browser-based frontend, the system requires no specialized hardware or local installation for end-users, ensuring maximum accessibility.
- High Baseline Accuracy: On historical datasets, the system demonstrates strong predictive reliability with ~94–96% accuracy on test splits, providing a solid linguistic fallback even when the live API returns no results.

XI. LIMITATIONS

- Language Constraint: The current implementation is strictly limited to English-language textual content, making it unable to detect misinformation in regional or global languages.
- Text-Only Modality: TruthLens does not currently support the verification of multimedia content, meaning it cannot detect deepfakes, manipulated images, or fabricated videos.
- Single-Source API Bias: Relying entirely on a single external service (Newsdata.io) for live verification introduces a potential single point of failure and limits the system's ability to cross-reference multiple independent journalistic sources simultaneously.
- Binary Output Limitations: The system currently delivers a binary "REAL" or "FAKE" verdict rather than probabilistic confidence scores, which might mask the uncertainty of certain borderline predictions.
- Lack of Deep Semantic Context: Because it relies on linear classification (PAC) and term frequencies (TF-IDF) rather than deep transformer architectures (like BERT or RoBERTa), the system may occasionally struggle to understand subtle context, sarcasm, or highly sophisticated crafted misinformation.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

XII. FUTURE SCOPE

While TruthLens provides a solid foundation for automated fact-checking, several avenues exist for future enhancement:

1. Advanced Transformer Models: Upgrading the core NLP engine from the Passive-Aggressive Classifier to fine-tuned transformer architectures like BERT (Bidirectional Encoder Representations from Transformers) or RoBERTa. These models better capture deep contextual semantic relationships and nuances in language, improving accuracy on highly subtle or well-crafted misinformation.

2. Multi-Source API Aggregation: Currently, the system relies on a single external provider (Newsdata.io). Future iterations should query multiple news APIs (e.g., Google News, Reuters API) simultaneously. By cross-referencing multiple verified databases, the system can prevent single-source bias and generate more robust verification confidence scores.

3. Multilingual Support: Expanding the application to analyze non-English text by utilizing multilingual transformer embeddings, making the tool globally applicable.

4. Media Verification: Misinformation increasingly relies on visual media. Extending the system to integrate reverse image search APIs and algorithmic deep-fake detection modules to verify images and videos alongside text claims.

5. Browser Extension Deployment: Packaging the TruthLens logic into a Google Chrome or Mozilla Firefox browser extension. This would allow the system to automatically scan and flag news directly within a user's social media feed (e.g., Twitter, Facebook) without requiring them to manually copy and paste text into a separate portal.

XIII. CONCLUSION

TruthLens successfully demonstrates an effective, scalable technical intervention to combat digital misinformation. By harmonizing an aesthetically engaging user interface, offline machine learning intelligence, and live REST API integrations, TruthLens transforms static fact-checking into a dynamic, real-time, and user-friendly experience.

The project proves that the fundamental limitation of supervised NLP classifiers—their inability to adapt to breaking news that post-dates their training data—can be overcome through a hybrid architecture. The Passive-Aggressive Classifier provides robust linguistic pattern recognition, while the Newsdata.io API supplies the necessary temporal context. The resulting web application operates with low latency, high accuracy, and requires minimal computational overhead, fulfilling both rigorous technical engineering standards and practical societal utility.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 299-310

REFERENCES

[1] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online Passive-Aggressive Algorithms. Journal of Machine Learning Research, 7, 551–585.
[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.
[3] Wang, W. Y. (2017). 'Liar, Liar Pants on Fire': A New Benchmark Dataset for Fake News Detection. arXiv:1705.00648.
[4] Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic Deception Detection: Methods for Finding Fake News. Proceedings of the ASIST Annual Meeting.
[5] Pedregosa, F. et al. (2011). Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
[6] Pallets Projects. (2024). Flask Web Framework Documentation. https://flask.palletsprojects.com
[7] Newsdata.io. (2024). Newsdata.io API Official Documentation. https://newsdata.io/docs