INTERNATIONAL JOURNAL IN ENGINEERING SCIENCES

Journal Homepage: https://www.ijesjournal.com

OPEN ACCESS

Research Paper

An Explainable Fake News Screening System Built on TF-IDF Features and Supervised Classifiers

Rushikesh Muthal[1]*, Aditya Thorat[1], Pralhad Rathod[1], Ganesh Ragade[2]

[1]Student, [2] Assistant Professor, Department of computer applications, Sandip Foundation's Sandip Institute of Engineering & Management, Nashik-422013, India

ARTICLE INFO

Published: 15 Nov. 2026

Keywords:
misinformation, fake news, text classification, TF-IDF, linear models, interpretability, Flask, WELFake.

DOI:
10.5281/zenodo.24248595

ABSTRACT

Misleading stories travel across the internet as quickly as accurate ones, and readers rarely have the time to verify each article they see. This paper presents a web-based screening tool that reads a headline together with its article body and assigns one of two labels, FAKE or REAL, while showing the user which words influenced the outcome. The system is written in Python, uses Flask for the interface and scikit-learn for learning. Raw text is cleaned and stemmed, converted into weighted unigram and bigram TF-IDF vectors, and passed to one of three candidate classifiers: Logistic Regression, Multinomial Naive Bayes, and a calibrated linear Support Vector Machine. The candidate with the best mean F1 score under three-fold cross-validation on the training split is retained and then scored once on a held-out 20% test split. Explanations are produced directly from the learned feature weights, without surrogate models. Two optional add-ons, a fact-check lookup and a trusted-outlet corroboration check, can be switched on through API configuration and are presented separately from the classifier's output. We describe the data audit, design, evaluation protocol and limitations, and we stress that the tool is a decision-support aid that cannot establish factual truth.

INTRODUCTION

Publishing online costs almost nothing, and sharing costs even less. This openness helps legitimate journalism reach people, but it also lets fabricated or distorted stories spread before anyone can check them. Professional fact-checkers do careful work, yet their capacity is small compared with the volume of content produced every day. Automated screening cannot replace that work, but it can help prioritise what deserves a closer look. In this paper, "fake news" has a deliberately narrow, operational meaning: an article is fake if the dataset used for training labels it that way. A classifier trained on such labels

***Corresponding Author:** Rushikesh Muthal

Address: *Department of computer applications, Sandip Foundation's Sandip Institute of Engineering & Management Nashik-422013, India*

Email : muthalrushikesh@gmail.com

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

learns the statistical differences between the two groups of examples. It does not verify claims, consult evidence or reason about the world. Keeping this distinction in view shapes every design decision described below, including how results are worded in the user interface.

1.1 Aims

- Offer a simple page where a person can paste a headline and article text and receive a prediction.
- Process training data and user input with exactly the same preparation steps so that the two stay consistent.
- Compare several well-understood classifiers fairly, selecting among them without touching the test data.
- Show a strength-of-prediction indicator and the terms that pushed the decision, so that users can judge the output rather than accept it blindly.
- Keep dataset upload, retraining, storage and analytics in the same application for convenience.

1.2 Scope and contribution

The system handles English-language article text only. It is a baseline, not a state-of-the-art detector. Its contribution is practical: a complete, reproducible pipeline that links data auditing, model selection, held-out evaluation, feature-weight explanations and optional external evidence inside one deployable program. Transformer-based models, multilingual support and social-context signals are outside the implemented scope and are discussed only as future directions.

Literature Review

Fake news detection has attracted significant attention from researchers in recent years. Numerous approaches have been proposed using machine learning, deep learning, and hybrid techniques.

Shu et al. presented a comprehensive survey of fake news detection on social media and highlighted the challenges associated with content-based and context-based detection methods. Their study emphasized the importance of feature engineering and user behavior analysis.

Ahmed et al. applied machine learning algorithms including Naive Bayes and Support Vector Machines for fake news classification. Their results demonstrated that textual features can effectively distinguish between authentic and fabricated news content.

Ruchansky et al. introduced a hybrid model combining textual content analysis with social context information. The proposed framework achieved higher classification accuracy compared to content-only approaches.

Devlin et al. proposed the BERT architecture, which revolutionized Natural Language Processing by introducing bidirectional contextual learning. BERT significantly improved performance across various NLP tasks including text classification and sentiment analysis. Go Liu et al. introduced RoBERTa, noan optimized version of BERT that achieved superior results through improved training strategies and larger datasets. RoBERTa has been successfully applied in fake news detection systems.

Recent studies have incorporated Explainable Artificial Intelligence techniques such as SHAP and LIME to interpret classification decisions.

These approaches improve transparency and user trust while maintaining high predictive performance.

Although existing studies have achieved promising results, challenges remain in handling multilingual content, evolving misinformation strategies, and ensuring explainability of automated decisions

Research Gap

The literature review reveals that significant progress has been made in fake news detection using Machine Learning and Deep Learning techniques. However, several challenges still exist.

1. Many traditional machine learning models rely heavily on manually engineered features and often fail to capture contextual meaning within news articles.

2. Existing systems primarily focus on English-language datasets and demonstrate limited effectiveness in multilingual environments.

3. Most deep learning models function as "black boxes," making it difficult for users to understand why a particular article was classified as fake or real.

4. Several studies focus only on accuracy while neglecting interpretability, transparency, and trustworthiness.

5. Real-world fake news continuously evolves, requiring adaptable and scalable detection systems.

The identified research gap motivates the development of an Explainable Fake News Detection Framework that combines Machine Learning, Transformer-based architectures, and Explainable AI techniques.

Problem Statement

1. Background & Context

The rapid proliferation of digital journalism, algorithmic content recommendation, and decentralized online publishing platforms has intensified the dissemination of unverified, misleading, and fabricated news content. Automated text classification frameworks powered by Natural Language Processing (NLP) offer scalable avenues to identify linguistic and structural indicators characteristic of disreputable reporting.

However, conventional binary text classifiers operate as rigid "black boxes." They generate detached, overconfident categorical labels without explaining the underlying rationale or contextualizing system uncertainty. When applied to high-stakes information ecosystems, these uncalibrated systems risk misguiding end-users, mistaking algorithmic correlation for factual verification, and failing silently on anomalous, ambiguous, or out-of-domain inputs.

2. Core Problem Formulation

The fundamental objective is to design, develop, and benchmark an end-to-end, production-grade Natural Language Processing software system that ingests a news item represented by a pair (H, B)—where H denotes the news **Headline** and B denotes the news **Article Body**—and maps the input to a binary classification space:

$$\mathcal{Y} \in \{\text{FAKE}, \text{REAL}\}$$

The classification task is strictly defined relative to the ground-truth distribution and editorial labeling conventions established by the underlying training corpus. Crucially, the system must not function merely as a discrete predictive pipeline; it must treat prediction, confidence calibration, feature

attribution, and input validation as interdependent operational requirements.

3. Key Technical Challenges & Operational Requirements

- **Bi-Textual Interaction & Semantic Incongruence:**

Headlines and article bodies exhibit distinct structural characteristics. Headlines frequently utilize clickbait mechanics, emotional exaggerations, or syntactically condensed phrasing, whereas article bodies contain narrative explanations, quotations, and contextual prose. The system must process both text streams simultaneously, capturing lexical, syntactic, and semantic interactions, while identifying potential headline-body stance discrepancies (e.g., an alarmist headline paired with an incongruent or neutral body).

- **Uncertainty Quantification & Calibrated Confidence:**

Raw softmax probabilities produced by deep neural networks or parametric classifiers are notoriously uncalibrated, frequently exhibiting overconfidence on incorrect predictions. The system must implement robust uncertainty quantification—utilizing techniques such as probability calibration (Platt Scaling, Isotonic Regression), temperature scaling, or Bayesian approximations. The output must communicate the statistical strength and epistemic uncertainty of the prediction to the end-user rather than providing a detached binary score.

- **Local Interpretability & Feature Attribution:**

To foster user trust, facilitate human-in-the-loop auditing, and expose spurious dataset correlations, the system must expose the exact lexical signals and tokens that influenced the inference. It must integrate local explainability mechanisms (such as token-level gradient attribution, Integrated Gradients, attention weight extraction, or model-agnostic methods like LIME/SHAP) to highlight the specific phrases driving the classification.

- **Input Guardrails & Out-of-Distribution (OOD) Detection:**

Real-world systems receive anomalous, truncated, or domain-shifted text. The pipeline must enforce strict input-validation guardrails that detect:

  - Insufficient input length (e.g., empty bodies, single-word headlines);
  - Non-informative text (e.g., pure boilerplate, non-textual syntax, gibberish);
  - Out-of-vocabulary or semantic distribution shifts.

When an input fails these validity criteria, the system must gracefully reject inference or issue explicit warning banners indicating that the input is insufficient for reliable classification.

- **Epistemic Humility & Anti-Definitive Framing (Ethical AI Alignment):**

Automated text models analyze *stylistic, linguistic, and statistical patterns* learned from labeled datasets—they do not verify real-world facts or conduct investigative journalism. Consequently, the user-facing interface, documentation, and API contracts must deliberately avoid presenting predictions as authoritative adjudications of truth. The system's output must be explicitly framed as an *algorithmic assessment of stylistic similarity to known training classes*, functioning as an assistive decision-support tool rather than an automated arbiter of truth.

- **Reproducibility, Modularity & Systematic Benchmarking:**

The underlying architecture must be designed for repeatable experimentation, modular data preprocessing, deterministic training pipelines (fixed random seeds, versioned data splits), and standardized evaluation across multi-class performance metrics (F_1-Score, Precision, Recall, Confusion Matrix, and Brier Score for probability calibration).

Proposed Methodology

1 Data preparation

The loader detects the title, text and label columns, standardises their names, converts accepted textual labels to numbers, removes duplicates, trims the publisher prefix and verifies that both classes are sufficiently represented. Training can draw from several sources in a fixed order of priority: a custom uploaded dataset, the full WELFake file, the WELFake sample, the ISOT fake and true files, and lastly a tiny built-in demonstration set.

2 Text preprocessing

Preprocessing is intentionally simple so that it is transparent and repeatable. It removes markup, web addresses and email addresses, extracts word and number tokens, lower-cases them, trims possessive endings, drops common English stop words and very short tokens, and applies a small suffix-stripping stemmer. Three kinds of style cue are kept as extra tokens: fully capitalised words, exclamation marks and numeric expressions. These cues can carry useful signal, but they may also push the model toward judging sensational formatting instead of content.

3 Feature extraction

Each document becomes a sparse TF-IDF vector over single words and word pairs. Terms must appear in at least two documents and in no more than 90% of them, and the vocabulary is capped at 60,000 entries. Sublinear scaling dampens the effect of a word repeated many times. The vectoriser fitted during training is reused unchanged at prediction time, so new text is always mapped into the original vocabulary.

4 Candidate models and selection

Candidate	Configuration	Rationale
Logistic Regression	max_iter = 1000, C = 5.0	Linear model with readable coefficients
Multinomial Naive Bayes	alpha = 0.1	Quick probabilistic baseline for sparse counts
Linear SVM	LinearSVC with C = 1.0, wrapped in three-fold probability calibration	Strong linear learner for high-dimensional text

The data are divided with a stratified 80/20 split and a fixed random seed of 42. Each candidate is scored by three-fold cross-validation on the 80% portion, using F1 with FAKE as the positive class. The candidate with the highest mean F1 is refitted on the full training portion and evaluated once on the 20% test portion. Because the test portion plays no part in choosing the model, the final estimate is not inflated by selection.

5 Scores, warnings and explanations

At prediction time the headline and body are concatenated, preprocessed and vectorised. The model returns a probability for the FAKE class, and the label is FAKE when that probability reaches 0.5. The reported strength is the larger of the probability and its complement, grouped into high, moderate and low bands. Warnings appear when no known vocabulary is found, when only a handful of features are active, or when the strength is low. Explanations multiply each active TF-IDF value by the corresponding model weight to estimate how much a term moved the decision toward either class, and the interface highlights the largest contributors. These explanations describe how the model behaves. They are not evidence about whether a statement is true, and the strength indicator should not be read as a calibrated probability of truth.

6 Optional external verification

Two modules can be enabled by supplying API credentials. The first queries a fact-checking service for existing reviews of claims similar to the submitted text. The second searches news sources for coverage by trusted outlets and compares candidate stories with the submission using sentence-embedding similarity. Both are reported next to, not inside, the classifier result. Finding no match proves nothing: the story may be new, regional or worded differently. The proposed methodology consists of multiple stages that transform raw news articles into meaningful predictions.

Related Work

Studies of misinformation detection generally divide into content-based and context-based families. Content-based methods look only at the article, including its vocabulary, structure and style. Context-based methods add information about who shared a story, how it propagated and how people reacted to it. Shu and colleagues surveyed both directions from a data-mining standpoint and highlighted the difficulty of collecting reliable labels [1]. Wang introduced a benchmark of short political statements with graded truthfulness labels [2], while Ahmed, Traore and Saad examined n-gram features with several classifiers for fake news and opinion spam [3]. Ruchansky and co-authors proposed a hybrid neural model that combines article text, user response and source characteristics [4].

For student and small-team projects, content-only models remain attractive because they need nothing beyond text and labels. TF-IDF weights a term higher when it is frequent in a document but uncommon across the collection, which makes it a dependable representation for sparse text. Logistic Regression yields coefficients that can be read as term influences. Multinomial Naive Bayes is very fast, although it assumes terms are independent given the class. Linear SVMs usually perform well in the high-dimensional spaces that text produces. Contextual language models such as BERT [5] and RoBERTa [6] capture word meaning in context and can outperform these baselines, but they demand far more computation and data. Model-agnostic explanation methods such as SHAP [9] and LIME [10] are widely used; our system instead reads weights straight from linear models, which is cheaper but limited to that model family. The gap this work addresses is not a new algorithm but the distance between a notebook experiment and a usable application. We combine a careful data-handling stage, disciplined model selection, a prediction interface with explanations, and clearly separated optional evidence checks.

Data and Audit

Two data assets were available. The first is a 2,000-row demonstration sample, documented as containing 1,000 fake and 1,000 real articles,

which ships with the application. The second is the full WELFake collection, which we inspected before use.

Audit item	Observed value
Records in full file	72,134
Columns	index field, title, text, label
Rows with label 1	37,106
Rows with label 0	35,028
Missing titles	558
Missing article bodies	39
Label convention used	1 = FAKE, 0 = REAL (to be confirmed per file)

The two classes are close in size but not identical. The index-like column is ignored. During loading, absent titles and bodies are replaced with empty strings, repeated title-and-text pairs are dropped, a recognised newswire-style prefix is removed, and records whose body has fewer than 30 characters are discarded. A later training step removes any record that leaves fewer than three tokens after preprocessing. Label meaning deserves special care. Different published copies of the same collection use opposite conventions, and a model trained with the wrong mapping can look accurate while telling users the reverse of the truth. The convention must therefore be verified against the exact file before each training run. More generally, a benchmark is a product of its own collection and annotation process; it can carry topic, publisher, date and style artefacts. Scores on a random split show how well the model reproduces the dataset's labelling, not how well it detects falsehood in the wild.

System Architecture

The application is organised as a sequence of stages.

Stage	Function
Input	Accepts a headline and body; rejects submissions that are too short or too long.
Preprocessing	Strips markup, links and email addresses; tokenises and lower-cases; removes stop words and very short tokens; applies a light suffix stemmer; appends style markers.
Feature extraction	TF-IDF over unigrams and bigrams with minimum document frequency 2, maximum document frequency 0.9, at most 60,000 features and sublinear term frequency.
Model selection	Cross-validated comparison of three classifiers using F1.
Final evaluation	Refit of the winner on the training split and a single evaluation on the held-out test split.
Prediction and explanation	Class, strength indicator, warnings and influential terms.
Web and admin layer	Flask pages and API, SQLite storage, dataset upload, retraining and analytics.
Optional verification	Fact-check lookup and trusted-outlet similarity search, inactive until configured.

The flow is: headline and body, then validation, preprocessing, vectorisation, classification, and finally a label with its explanation. If the optional services are enabled, their results are attached as separate evidence rather than merged into the classifier's score.

The implementation relies on Python with pandas and NumPy for data handling, scikit-learn for vectorisation, models and metrics, joblib for saving the fitted pipeline, JSON for the metrics file, Flask for serving pages and endpoints, and SQLite for records.

Experimental Protocol

The archive supplied with the source code did not include a trained model or a metrics file. For that reason this paper does not assert measured accuracy figures. The code writes a metrics file after each run, containing the chosen model, dataset details, split sizes, test metrics, the confusion matrix, the cross-validation comparison and the most influential terms. The values below should be copied from that file after a real training run on the chosen dataset.

1 Metrics

FAKE is the positive class. A true positive is a fake article predicted as fake, a true negative is a real article predicted as real, a false positive is a real article wrongly flagged as fake, and a false negative is a fake article that slipped through as real.

Metric	Formula	Meaning
Accuracy	(TP + TN) / (TP + TN + FP + FN)	Share of all predictions that are correct
Precision	TP / (TP + FP)	Of articles flagged FAKE, the share that truly are
Recall	TP / (TP + FN)	Of truly fake articles, the share that are caught
F1	2PR / (P + R)	Harmonic mean of precision and recall

2 Results (to be completed from the generated metrics file)

Model	Mean CV F1	Test accuracy	Test precision	Test recall	Test F1
Logistic Regression	[insert]	[insert]	[insert]	[insert]	[insert]
Multinomial Naive Bayes	[insert]	[insert]	[insert]	[insert]	[insert]
Calibrated Linear SVM	[insert]	[insert]	[insert]	[insert]	[insert]

Only the model chosen by cross-validation needs test-set figures; if test figures are shown for all three, the table should make clear that selection was made beforehand and independently of them. Include the confusion matrix and a short discussion of which kind of error is more frequent.

3 Reproducibility checklist

- Record the dataset file name, its size after cleaning, label counts and whether the sample or the full file was used.

- Verify the label convention before training.
- Note library versions and the training date.
- Keep the seed and split rule unchanged when comparing runs.
- Report test metrics from the held-out portion only.
- Examine the confusion matrix and per-class errors, not accuracy alone.
- Test on a second dataset or a time-ordered split before claiming real-world usefulness.

4 Illustrating explanations

After training, the most positive and most negative weights in the selected linear model can be tabulated to show which terms the model associates with each class. This list should be inspected critically. If it is dominated by publisher names, newswire phrases or image-credit boilerplate rather than claim content, that is a sign that the model has learned source artefacts, which is a limitation worth reporting honestly.

Discussion

1. Architectural Trade-offs: Interpretability and Operational Feasibility

Selecting linear architectures—such as Logistic Regression, Linear Support Vector Classifiers (Linear SVC), or Passive-Aggressive Classifiers—parameterized over Term Frequency-Inverse Document Frequency (TF-IDF) n-gram vectorizers reflects deliberate design trade-offs tailored for transparent, low-overhead deployments:

- **Computational Efficiency and Determinism:** High-capacity neural backbones (e.g., RoBERTa, DeBERTa) demand substantial compute resources, dedicated GPU acceleration, and prolonged inference cycles. In contrast, TF-IDF representations paired with convex linear objectives train in minutes on standard commodity hardware and execute inference with sub-millisecond latencies. This deterministic behavior guarantees reproducible parameter convergence and zero-latency execution in production serving environments.
- **Direct Intrinsic Interpretability:** Linear models expose monotonic mapping between feature weights and decision boundaries. Each learned coefficient corresponds directly to an n-gram token's positive or negative log-odds contribution. This intrinsic explainability eliminates the need for computationally heavy post-hoc approximation wrappers (such as perturbation-based LIME), allowing developers and auditors to inspect the exact lexical features driving any classification directly from the weight vector.
- **Rigorous Evaluation Discipline:** Optimizing hyperparameters using the macro-averaged F_1-score mitigates class-imbalance biases by penalizing asymmetric trade-offs between Type I errors (false alarms, labeling legitimate journalism as fake) and Type II errors (missed detections, classifying fabricated narratives as real). Maintaining a strict physical isolation between the cross-validation hyperparameter tuning loop and the final holdout test split prevents data leakage and hyperparameter over-adaptation, ensuring that reported performance metrics reflect unbiased generalization bounds.

2. Spurious Correlations and Dataset Exploitation (Shortcut Learning)

While linear baselines achieve competitive accuracy on closed-corpus benchmarks, these empirical gains often stem from "shortcut learning"—the exploitation of unintended statistical artifacts rather than true semantic deception indicators:

- **Publisher-Specific Lexical Signatures:** Public news classification datasets frequently aggregate articles from a small set of source domains. Models inadvertently latch onto publisher-specific idiosyncrasies—such as editorial capitalization rules, distinct punctuation patterns (e.g., hyper-sensationalist exclamation density, non-standard quote conventions), or repetitive layout boilerplate. While stripping explicit newswire prefixes (e.g., *"WASHINGTON (Reuters) - ..."*) removes obvious leakage, subtler syntactical and typographic artifacts persist throughout the document bodies.

- **Topical and Temporal Confounding:** Datasets collected during specific political or historical cycles exhibit pronounced topic-label correlations. If articles concerning a specific political figure or geopolitical event are disproportionately represented in the "Fake" partition of the training corpus, the model learns that the topical entity itself is inherently deceptive. The classifier subsequently penalizes legitimate coverage of that entity, substituting semantic fact-checking with topical profiling.

- **Leakage via Standard Random Splits:** Employing uniform, random cross-validation splits on web-scraped journalism introduces systemic evaluation bias. Syndicated stories, wire service reprints, and recurring viral claims often produce near-duplicate textual pairs. A random split inevitably distributes near-identical variants of the same core story across both the training and test partitions. This proximity inflates test scores, transforming the task from out-of-sample generalization into near-memorization.

- **Mitigation Protocols:** To establish authentic generalizability, evaluation frameworks must enforce:

1. **Temporal Splits:** Partitioning training and test sets chronologically (e.g., training on historical quarters and testing strictly on future quarters) to quantify temporal degradation and concept drift;

2. **Cross-Domain/Cross-Dataset Evaluation:** Validating models on fully independent corpora (e.g., benchmarking an ISOT-trained model against LIAR or WELFake) to expose dataset-specific overfitting.

3. Integrating External Verification: Mechanics, Boundaries, and Failure Modes

Augmenting local textual classification with automated external verification modules (e.g., search indexing, fact-checking API aggregators, or knowledge graph querying) provides external corroboration, yet introduces critical systems-level trade-offs:

- **Operational Latency and Infrastructure Dependencies:** Real-time retrieval pipelines introduce distributed external dependencies. Upstream API rate limits, dynamic IP throttling, unpredictable network round-trip time (RTT), and remote endpoint deprecation threaten pipeline uptime. A classifier meant to deliver real-time verdicts becomes vulnerable to network failures when tightly coupled to live web APIs.

- **The Perils of Semantic Thresholding:** Assessing similarity between a query claim

and retrieved fact-checks relies heavily on vector embedding distance (e.g., Cosine Similarity over dense bi-encoders). Setting high similarity thresholds causes false negatives (ignoring relevant debunks due to slight phrasing variations), whereas permissive thresholds yield false positives (retrieving articles that share named entities but address entirely distinct assertions).

- **Information Asymmetry and the Open-World Assumption:** The Internet is inherently asynchronous. The absence of an external debunk does not validate an article; it frequently indicates that the claim is hyper-local, recently fabricated (zero-day disinformation), or outside mainstream media scrutiny. The system must adhere to open-world semantics: a missing retrieval match constitutes **"unverified / absence of evidence"**, never **"corroboration of authenticity"**.

- **Architectural Decoupling:** External verification modules must serve solely as non-blocking, additive layers rather than hard decision overrides. The primary classifier must output its calibrated linguistic risk score independently, while retrieval agents attach complementary context banners when authoritative corroborations or direct refutations are discovered. Under no circumstances should the system synthesize these separate signals into a single opaque score, preserving algorithmic transparency and epistemic humility.

Ethics, Limitations and Risks

1 Limitations

- Two labels flatten a wide spectrum that includes satire, opinion, partially accurate claims, outdated reports and missing context.
- The system works for English only.
- Quality depends on the dataset's coverage, annotation policy and source mix.
- The strength indicator is not guaranteed to be calibrated.
- Weight-based explanations show associations, not causes and not evidence.
- External checks depend on third-party services and configuration.
- The bundled sample is meant for demonstration, not for final evaluation.

2 Responsible use

Results should be described as a screening signal. Users should be encouraged to read original sources, compare several credible outlets, check publication dates and consult professional fact-checkers. The interface should never accuse a person or organisation of deception on the basis of a FAKE label alone. For deployment, administrative credentials must be protected, default secrets changed, access to uploaded datasets restricted, and the storage of submitted text and history documented in a privacy notice.

Future Work

To evolve the proposed system from an academic prototype into a production-grade, clinically and operationally robust NLP pipeline, subsequent research will focus on the following eight technical extensions:

1. Full-Dataset Benchmarking and Repeated-Run Variance Quantification

Subsequent iterations will expand training across complete, un-sampled benchmark corpora (e.g., WELFake, ISOT, LIAR, and FakeNewsNet). To establish statistical rigor, future experiments will execute repeated k-fold cross-validation across multiple random seeds, reporting mean performance along with empirical confidence intervals (e.g., 95% bootstrap intervals) and standard deviations for Macro-F_1, Precision, and Recall. This will isolate the true algorithmic signal from stochastic initialization variance.

2. Cross-Corpus and Chronological Generalization Analysis

To combat overfitting to specific editorial patterns and temporal topics, evaluation protocols will introduce:

- **Out-of-Distribution (OOD) Cross-Corpus Evaluation:** Models trained on one distribution (e.g., ISOT) will be tested directly against independent corpora with disparate labeling policies and topical distributions (e.g., LIAR).
- **Chronologically Stratified Splits:** Data will be partitioned along a strict temporal boundary—training models on past reporting and evaluating exclusively on subsequent publication windows. This setup directly quantifies model resilience against concept drift, evolving linguistic trends, and emergent geopolitical vocabulary.

3. Probability Calibration and Reliability Diagnostics

Raw classification scores will be converted into trustworthy, calibrated probabilities by implementing post-hoc calibration algorithms, such as:

- **Platt Scaling (Logistic Calibration)** and **Isotonic Regression** for linear baselines;
- **Temperature Scaling** for neural representations.

Future validation will evaluate calibration performance quantitatively using Expected Calibration Error (ECE) and Brier Score, supplemented by Reliability Diagrams (calibration curves) to ensure that a confidence output of 80% corresponds to an 80% empirical likelihood of alignment with dataset labels.

4. Decoupled Claim-Level Evidence Retrieval

To enhance contextual decision support without corrupting model transparency, subsequent designs will implement an external, decoupled Retrieval-Augmented pipeline:

- **Claim Decomposition:** Deconstructing articles into standalone atomic factual claims via dependency parsing or text-to-text generation.
- **Open-Web/Fact-Check Evidence Fetching:** Querying external fact-checking APIs (e.g., Google Fact Check Tools API) and knowledge graphs to retrieve relevant refutations or corroborating reporting.
- **Architectural Decoupling:** Retrieved evidence will be rendered strictly as complementary metadata cards rather than merged into the classifier's internal feature vector. This design ensures that retrieval API failures or missing debunks never silently bias or override the primary stylistic classification.

5. Systematic Transformer Experiments Under Explicit Compute Budgets

While linear models provide an interpretable baseline, future investigations will benchmark contextual Transformer representations (e.g., RoBERTa, DeBERTa-v3, and long-context variants like Longformer) as a distinct, dedicated experimental track. Each experiment will explicitly catalog:

- Memory footprint, FLOPs, parameter count, and GPU training/inference runtimes;
- The operational trade-off between latency-critical throughput and incremental gains in F_1-score relative to optimized linear baselines.

6. Multilingual Adaptation and Indic Language Support

Misinformation frequently spreads across linguistically diverse, low-resource media ecosystems. Future work will expand linguistic coverage to Indian regional languages (e.g., Hindi, Marathi, Bengali, Tamil):

- Benchmarking multilingual encoders (such as IndicBERT, MuRIL, and XLM-RoBERTa);
- Handling script mixing, phonetic transliteration (e.g., Hinglish, Romanized Marathi), and dialectical nuances common in decentralized communication channels and regional digital news outlets.

7. Deduplication and Near-Duplicate Leakage Mitigation

To eliminate data contamination caused by syndicated wire stories and social media reposts spanning both training and testing sets, future pipelines will integrate near-duplicate filtering:

- Applying MinHash combined with Locality-Sensitive Hashing (LSH) or dense vector cosine clustering to identify and prune document pairs exhibiting high Jaccard similarity;
- Enforcing strict source-domain and cluster-level splitting, guaranteeing that near-identical variations of a single reporting thread are isolated entirely to either the train or test partition.

8. MLOps, Continuous Monitoring, and Model Version Tracking

To ensure operational integrity and compliance in production environments, the system infrastructure will incorporate standard MLOps best practices:

- **Metadata & Artifact Tracking:** Managing deterministic pipelines using tools like MLflow or DVC for versioning datasets, preprocessing pipelines, and serialized model checkpoints;
- **Drift & Degradation Monitoring:** Implementing continuous distribution monitoring (e.g., Population Stability Index, Kolmogorov-Smirnov tests on token distributions) to detect data drift, vocabulary obsolescence, and automated model staleness in real-time streaming ingestion.

Conclusion

This paper presented the design, implementation, and methodological formulation of an explainable fake news screening application intended for decision-support workflows. Addressing the limitations of opaque "black-box" classifiers, the proposed system establishes a reproducible, end-to-end natural language processing pipeline that emphasizes model transparency, confidence calibration, and contextual verification.

The underlying architecture enforces a systematic training and validation protocol:

- **Feature Representation & Selection:** Raw headline and body texts undergo standardized text normalization, following which TF-IDF unigram and bigram representations are constructed to balance linguistic coverage with computational efficiency.

- **Deterministic Model Selection:** Three candidate linear classifiers are tuned and evaluated strictly using cross-validated F_1-score on the training split, prioritizing symmetric precision-recall balance to guard against both false alarms and missed misinformation.

- **Leakage-Free Generalization:** The best-performing model configuration is evaluated exactly once on a strictly held-out test split, ensuring reported metrics remain free from hyperparameter adaptation bias.

The experimental setup leverages the **WELFake** corpus, an aggregated benchmark comprising 72,134 records (distributed between 37,106 label-1 and 35,028 label-0 rows prior to cleaning). However, this study deliberately underscores an essential methodological principle: *true system efficacy cannot be inferred solely from in-corpus cross-validation scores.* Authentic real-world generalizability requires validation across out-of-distribution corpora, temporal splits, and resilience tests against publisher-specific styling shortcuts, recurring political jargon, and near-duplicate leakage.

Rather than operating as an autonomous, unchallengeable arbiter of objective truth, the system is designed to provide calibrated epistemic humility. Its interface exposes actionable diagnostic signals—including continuous prediction strength indicators, input-validity warnings for truncated or out-of-domain text, token-level feature attributions, and optional evidence streams retrieved from external fact-checking and news APIs. By decoupling local stylistic predictions from external verification, the system functions as an assistive screening mechanism that empowers human readers and researchers to critically examine claims and navigate digital information ecosystems with heightened scrutiny.

REFERENCES

1. 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.
2. W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in Proc. ACL, 2017.
3. H. Ahmed, I. Traore and S. Saad, "Detecting opinion spams and fake news using text classification," Security and Privacy, vol. 1, no. 1, e9, 2018.
4. N. Ruchansky, S. Seo and Y. Liu, "CSI: A hybrid deep model for fake news detection," in Proc. CIKM, pp. 797–806, 2017.
5. J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019.
6. Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv:1907.11692, 2019.
7. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
8. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions,"

in Advances in Neural Information Processing Systems, 2017.

9. M. T. Ribeiro, S. Singh and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in Proc. KDD, pp. 1135–1144, 2016.
10. Project source archive and WELFake_Dataset.csv, supplied with the application and audited for this paper.

HOW TO CITE: Rushikesh Muthal*, Aditya Thorat, Pralhad Rathod, Ganesh Ragade, An Explainable Fake News Screening System Built on TF-IDF Features and Supervised Classifiers, Int. J. in Engi. Sci., 2026, Vol 3, Issue 11, 15-29. https://doi.org/10.5281/zenodo.24248595