

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 11 ISSUE : 07 Print / Issue Publication Date: November 2026

ISSN : 2455-2143

DOI : 10.33564/IJEAST.2026.v11i07.003

Indexed In

Google SCHOLAR OPEN ACCESS doi Crossref

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 18-22
Published Online November 2026 in IJEAST (http://www.ijeast.com)

VERITAS AI: FAKE NEWS DETECTION USING NLP

Mithilesh Meshram, Gyaneshwar Tiwari, Vishwajeet Sutar, Aditya Raut
Department of Computer Science and Engineering,
MIT College of Railway Engineering & Research,
Barshi, Maharashtra, India

Prof. Archana Bhosale
Professor, Department of Computer Science and Engineering,
MIT College of Railway Engineering & Research,
Barshi, Maharashtra, India

Abstract— **Online misinformation is difficult to assess at scale because a label produced by a text classifier does not, by itself, explain why a claim should be trusted or rejected. This paper presents VERITAS AI, a capstone prototype designed around an evidence-centred workflow for claim analysis. The system accepts article text, a URL, or a directly entered claim; normalizes text; identifies and decomposes checkable claims; generates both supportive and skeptical search queries; retrieves candidate news, web, and fact-check results; and presents source metadata and uncertainty in a web interface. Inspection of the supplied Phase 1 code indicates that claim detection/decomposition, query generation, evidence normalization and deduplication, provider abstractions, API endpoints, and a React-based interface are implemented. The repository also explicitly reports demo mode by default and marks the machine-learning classifier as not configured. Consequently, this paper reports an implementation review and test-design evidence, not a completed accuracy comparison between TF-IDF and DistilBERT. The contribution is a transparent workflow that separates claim extraction and evidence retrieval from the final judgement, labels demo data, preserves partial-failure states, and avoids presenting unavailable model predictions as fact. Future work includes configuring live providers securely, completing evidence entailment assessment, training and evaluating the proposed classifiers, and conducting multilingual and user-centred validation.**

Keywords—**Fake news detection, natural language processing, claim extraction, evidence retrieval, fact-checking, explainable AI, misinformation.**

I. INTRODUCTION

Digital news and social platforms enable rapid information sharing, but they also allow misleading claims to circulate before independent verification is available [1], [2]. Fake-news detection has therefore developed from feature-based classification toward approaches that combine contextual language representations, social context, external knowledge, and evidence-based verification [3]–[5]. A classifier can learn statistical patterns associated with labelled datasets, yet it may fail when topics, writing styles, or events differ from its training data. This motivates a design in which automated predictions are treated as signals and are presented alongside traceable evidence rather than as unquestionable facts.

VERITAS AI is a web-based capstone prototype intended to support this workflow. The supplied project synopsis proposes an English-language, text-first system that analyses articles or claims, retrieves relevant fact-checks and current news, and visualizes evidence. The current repository is more specific: it includes claim detection and decomposition services, query generation, evidence normalization and deduplication, provider interfaces, API routes, a React/TypeScript frontend, and explicit demo-mode indicators. The synopsis also lists TF-IDF and DistilBERT classification as objectives; however, the checked-in ML service returns MODEL_NOT_CONFIGURED, so these classifiers are not treated as operational in the present evaluation.

The paper makes three practical contributions: (1) it documents an evidence-first architecture for a misinformation verification prototype; (2) it describes implementation safeguards visible in the code, including demo labelling, explicit provider failure states, source-type metadata, and deduplication; and (3) it defines a reproducible evaluation plan without inventing performance numbers that have not been measured.

II. RELATED WORK

Lazer et al. frame fake news as a socio-technical problem and argue that understanding its spread requires more than isolated content classification [3]. Survey literature organizes detection methods into linguistic, propagation-based, knowledge-based, and hybrid approaches, while highlighting limitations caused by dataset bias, changing topics, and inconsistent definitions of misinformation [4]–[7]. These findings support the need for

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 18-22
Published Online November 2026 in IJEAST (http://www.ijeast.com)

VERITAS AI to retain evidence provenance and uncertainty instead of collapsing all signals into a single binary output.

Transformer models such as BERT [8] and its compact variants [9] can encode contextual relationships more effectively than sparse lexical features in many text classification tasks. Hybrid approaches combine transformer representations with conventional classifiers, and recent work explores knowledge-based or multimodal signals to improve detection [10]–[12]. However, strong benchmark results do not automatically transfer to a deployed application: dataset leakage, label quality, topic overlap, class imbalance, and distribution shift can inflate apparent performance. A fair comparison requires the same data splits, preprocessing policy, and metrics for all baselines.

Research on event-agnostic representations explicitly addresses the difficulty of detecting fake news about events not seen during training. Pu et al. propose DEAR, which disentangles authenticity-related and event-specific information, reporting improved performance in an unseen-event setting [13]. This line of work reinforces the importance of testing generalization beyond random splits. For an evidence-verification system, a related challenge is that a retrieved article may repeat the original claim without independently confirming it. Research on automated fact-checking, such as the FEVER dataset for claim verification against textual evidence [14] and the survey by Guo et al. [15], likewise treats claim detection, evidence retrieval and verdict prediction as distinct stages. Therefore, source diversity, publication dates, and claim–evidence relationships must be considered separately from raw search-result counts.

VERITAS AI differs from a standalone classifier by emphasizing a staged workflow: claim extraction, query generation, evidence retrieval, normalization, source classification, and review. The current phase does not yet establish that the full evidence-to-verdict loop is empirically accurate. It should be understood as an implementation foundation and a testable architecture rather than a validated automated fact-checker.

III. PROBLEM STATEMENT AND OBJECTIVES

Manual fact-checking is time-consuming, and style-only classifiers provide limited justification for their decisions. The project addresses the engineering problem of building a prototype that converts unstructured news text into checkable claims, searches for potentially relevant evidence, and presents the evidence with clear source and uncertainty information.

The synopsis identifies these objectives: (a) preprocess news text using NLP; (b) build a TF-IDF baseline [16] with conventional machine learning; (c) fine-tune DistilBERT [9] and compare performance; (d) retrieve fact-checks and current news, then assess semantic similarity [17] and natural-language inference [18]; (e) provide a web interface and persistent storage; and (f) containerize the system. In the supplied Phase 1 code, the implemented claim and retrieval pipeline is more mature than the classifier and final-verdict stages. Accordingly, the current work focuses on implementation status, safety-oriented behaviour, and a future evaluation protocol.

IV. SYSTEM ARCHITECTURE

The prototype follows a frontend–API–service architecture. The React/TypeScript interface provides Home, Analyze, Saved Claims, Evidence, Evidence Graph, Report, Source Intelligence, Claim History, Narratives, Live Monitoring, Alerts, Settings, and Responsible AI navigation. The backend is implemented in Python with FastAPI, typed schemas, service modules, and provider abstractions. PostgreSQL is configured as the persistence layer, and the README documents Docker Compose for the database and local development commands.

TABLE I. Major system layers and observed responsibilities

Layer	**Responsibility**
Frontend (React + TypeScript)	Input modes, claim/evidence views, source cards, saved claims, system-status and demo indicators.
FastAPI backend	API validation, analysis and evidence endpoints, service orchestration, system status.
Claim detection	Text normalization, sentence segmentation, claim extraction/decomposition, checkability and entity/attribute extraction.
Evidence retrieval	Supportive and skeptical query generation, web/news/fact-check provider calls, normalization and deduplication.
Provider layer	Mock and live provider adapters; explicit handling of missing configuration and provider errors.
Persistence / deployment	PostgreSQL models and migrations; Docker Compose-based local setup.
ML / NLI (planned or incomplete)	Classifier is explicitly not configured; final entailment-based verdict generation is not established by this Phase 1 review.

V. METHODOLOGY

A. Input and text normalization

The interface exposes separate modes for article text, URL, direct claim, and image. The supplied screenshots show the article and claim forms, including a headline field, article-text area, optional source URL, and a direct-claim input. In the reviewed code, text utilities normalize whitespace and HTML entities and segment sentences while preserving offsets and handling common abbreviations and decimal numbers. The presence of an image input in the interface should not be interpreted as proof that image-based misinformation detection is implemented end to end.

B. Claim extraction and decomposition

The claim-detection package contains normalization, segmentation, extraction, decomposition, checkability, lexicon,

and gazetteer components. The goal is to separate factual propositions from surrounding narrative so that retrieval can target a specific statement. Claims are represented with structured attributes such as claim type, canonical subject/object, entities, locations, dates, and numbers where detected. These fields support query generation and traceability; extraction confidence must not be confused with truth confidence.

C. Evidence query generation and retrieval

The evidence service generates multiple search queries, including SUPPORT and SKEPTIC modes and a separate fact-check query. For Indian government-related topics, the query logic can include official-domain searches. Search and fact-check integrations are abstracted behind provider interfaces. The repository includes a real search provider adapter and a Google Fact Check provider adapter, as well as mock providers. Real retrieval depends on valid environment configuration and external service availability; the README states DEMO_MODE=true by default.

D. Evidence normalization and source handling

Retrieved items are normalized into a common evidence representation. The reviewed tests cover URL canonicalization, removal of tracking parameters, rejection of unsafe URL schemes, source-domain categorization, publication-date provenance, and merging repeated results found by multiple queries. Near-duplicate syndicated stories can be flagged rather than silently removed. This is important because repeated copies of one report do not constitute independent corroboration. The implementation distinguishes primary, secondary, user-generated, and fact-checker source categories, but source category alone is not a guarantee of correctness.

E. Verdict and model policy

The classifier module explicitly returns MODEL_NOT_CONFIGURED with null probability and model metadata when no trained model is available. This behaviour is preferable to inventing a label or confidence score. The paper therefore does not report real/fake classification performance, nor does it claim that NLI-based support/contradiction assessment is complete. The interface's product principle—evidence before conclusions—is consistent with treating retrieved material as candidate evidence that requires assessment.

F. Data and privacy considerations

The project synopsis proposes public labelled datasets for training (such as LIAR [19] and FakeNewsNet [20]) and standard metrics for evaluation, but the supplied materials do not identify a finalized dataset version, sample count, class distribution, train/test split, or trained model artifact. These must be recorded before reporting quantitative model results. API keys should be supplied through local environment variables and must not be committed to source control. Retrieved content should be handled as untrusted input, and system outputs should retain source URLs, timestamps, and demo/live provenance.

VI. IMPLEMENTATION STATUS

The repository and screenshots support the following status assessment. This is a code-and-document review, not a claim that a full production deployment was executed during this paper's preparation.

- Implemented in the supplied code: claim text normalization and sentence segmentation; claim detection/decomposition modules; checkability and structured attribute extraction; support/skeptic/fact-check query generation; evidence normalization, URL canonicalization, source classification, date provenance and deduplication; search and fact-check provider abstractions; API endpoints and system-status reporting; frontend analysis, saved-claim, evidence, and responsible-AI views.
- Explicitly incomplete or configuration-dependent: the ML classifier returns MODEL_NOT_CONFIGURED; the system status tests expect the search provider to be NOT_CONFIGURED in the default configuration; fact-check retrieval may be not configured; live external search depends on API credentials and provider availability; the full claim-to-evidence-to-final-verdict workflow has not been validated as a complete truth-classification system.
- Visible UI evidence: the screenshots show a dark-themed dashboard with an evidence-first statement, Analyze modes, and a Saved Claims page listing 22 stored claims. These are interface observations only and are not a measure of classification accuracy, retrieval precision, or factual correctness.

VII. TESTING AND EVALUATION

The supplied repository includes backend tests for API claim handling, claim detection and decomposition, extraction, text normalization, query generation, provider parsing and error handling, evidence retrieval states, URL normalization/deduplication, system status, and the unconfigured ML behaviour. Test cases assert, among other behaviours, that total provider failure yields an explicit FAILED state without fabricated evidence; partial provider failure is surfaced as PARTIAL; quota errors stop further requests; and demo evidence is clearly marked. These are useful functional and failure-mode checks.

A test suite's presence is not equivalent to a measured model benchmark. No verified run log or held-out dataset metrics were supplied with the materials reviewed here. Therefore, accuracy, precision, recall, F1-score, confusion matrix, retrieval precision, and latency are not reported as achieved results. The saved-claims count shown in the screenshot (22) is a UI/database state, not a labelled evaluation set.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 18-22
Published Online November 2026 in IJEAST (http://www.ijeast.com)

TABLE II. Evaluation plan for the next phase

Evaluation target	**Method**	**Report**
TF-IDF baseline	Train/test split with fixed seed; fit vectorizer only on training data.	Accuracy, precision, recall, macro-F1, confusion matrix.
DistilBERT	Same split and labels; document tokenizer, epochs, batch size and seed.	Same metrics plus training/inference cost.
Generalization	Hold out topics/events or use time-based split where possible.	Per-topic performance and degradation on unseen events.
Evidence retrieval	Manually label a sample of retrieved items for relevance and stance.	Precision@k, recall@k where reference evidence exists, source diversity.
Evidence reasoning	Annotate claim–evidence pairs as supports, refutes, or insufficient information.	Per-class precision/recall/F1 and disagreement analysis.
Reliability	Simulate missing keys, timeouts, rate limits and malformed responses.	Correct status reporting, no fabricated evidence, no secret leakage.

Recommended evaluation protocol; no metric values are implied.

VIII. DISCUSSION

The strongest aspect of the current prototype is its separation of concerns. Claim extraction, query generation, provider access, evidence normalization, and presentation are separate components, allowing each stage to be tested and improved independently. Explicit statuses for demo, partial, failed, and unconfigured states also reduce the risk that an unavailable API or model will be mistaken for a successful verification.

The main limitation is that a robust pipeline is not the same as a validated verdict engine. Search engines retrieve relevant-looking pages, not necessarily authoritative or independent evidence. A claim may be time-sensitive, ambiguous, or too broad to verify from a snippet. Source reputation, publication date, independence, and whether the evidence actually entails or contradicts the claim must be assessed. The current default demo mode and unconfigured classifier make it especially important for the interface to preserve these distinctions.

The project synopsis lists multilingual support, image/video analysis, Redis/Celery background processing, and production-scale deployment as future extensions or beyond current scope. The paper therefore treats English-language text claims as the current scope and does not claim Hindi or Marathi support. Responsible deployment should include clear explanations, citations to source material, calibrated uncertainty, human review for high-impact claims, and monitoring for shifts in topics and source quality.

IX. LIMITATIONS AND FUTURE WORK

- Configure and validate live search and fact-check APIs; document provider quotas, coverage and date limits.
- Complete evidence-level semantic similarity and NLI/stance assessment, with an explicit insufficient-evidence outcome.
- Train TF-IDF and DistilBERT on documented datasets; prevent leakage and publish a reproducible split and configuration.
- Evaluate on unseen events and recent claims to measure temporal and topical generalization.
- Add human-labelled evidence-relevance and stance datasets; measure source diversity and repeated-syndication effects.
- Improve provenance and uncertainty displays, including source date, source type, query path, and whether data are demo or live.
- Only after English evaluation, extend to Hindi and Marathi with appropriate language-specific data, models, and expert review.
- Conduct usability, security, privacy, and deployment testing before any real-world use.

X. CONCLUSION

This paper documents VERITAS AI as an evidence-centred misinformation verification prototype. The supplied source code supports a staged architecture for claim extraction, query generation, evidence retrieval, normalization, source handling, and frontend review. It also contains important safeguards: demo-mode labelling, explicit provider failure states, safe URL handling, duplicate-source awareness, and an ML service that honestly reports when a model is not configured. At the same time, the available materials do not establish a completed TF-IDF/DistilBERT comparison or a validated automated verdict engine. The next phase should prioritize live-provider configuration, evidence stance assessment, reproducible model training, and rigorous evaluation before claiming real/fake detection performance.

XI. REFERENCES

[1]. Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online, Science, vol. 359, no. 6380, (pp. 1146–1151).

[2]. Allcott, H., and Gentzkow, M. (2017). Social media and fake news in the 2016 election, Journal of Economic Perspectives, vol. 31, no. 2, (pp. 211–236).

[3]. Lazer, D.M.J., et al. (2018). The science of fake news, Science, vol. 359, no. 6380, (pp. 1094–1096).

[4]. Ali, A.A., et al. (2022). Fake news detection techniques on social media: a survey, Wireless Communications and Mobile Computing, Article ID 6072084.

[5]. Alghamdi, J., Luo, S., and Lin, Y. (2024). A comprehensive survey on machine learning approaches for fake news detection, Multimedia Tools and Applications, vol. 83, (pp. 51009–51067).

[6]. Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: a data mining perspective, ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, (pp. 22–36).

[7]. Zhou, X., and Zafarani, R. (2020). A survey of fake news: fundamental theories, detection methods, and opportunities, ACM Computing Surveys, vol. 53, no. 5, Article 109.

[8]. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding, in Proc. NAACL-HLT 2019, (pp. 4171–4186).

[9]. Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv:1910.01108.

[10]. Essa, E., Omar, K., and Alqahtani, A. (2023). Fake news detection based on a hybrid BERT and LightGBM models, Complex & Intelligent Systems, vol. 9, no. 6, (pp. 6581–6592).

[11]. Nair, V., et al. (2024). A knowledge-based deep learning approach for automatic fake news detection using BERT on Twitter, Procedia Computer Science, vol. 235, (pp. 1870–1882).

[12]. Verma, A., et al. (2023). MCred: multi-modal message credibility for fake news detection using BERT and CNN, Journal of Ambient Intelligence and Humanized Computing.

[13]. Pu, X., Wu, H., Bi, X., Wu, Y., and Gao, X. (2025). DEAR: disentangled event-agnostic representation learning for early fake news detection, Transactions of the Association for Computational Linguistics, vol. 13, (pp. 343–356).

[14]. Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018). FEVER: a large-scale dataset for fact extraction and verification, in Proc. NAACL-HLT 2018, (pp. 809–819).

[15]. Guo, Z., Schlichtkrull, M., and Vlachos, A. (2022). A survey on automated fact-checking, Transactions of the Association for Computational Linguistics, vol. 10, (pp. 178–206).

[16]. Salton, G., and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, Information Processing & Management, vol. 24, no. 5, (pp. 513–523).

[17]. Reimers, N., and Gurevych, I. (2019). Sentence-BERT: sentence embeddings using Siamese BERT-networks, in Proc. EMNLP-IJCNLP 2019, (pp. 3982–3992).

[18]. Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference, in Proc. NAACL-HLT 2018, (pp. 1112–1122).

[19]. Wang, W.Y. (2017). “Liar, liar pants on fire”: a new benchmark dataset for fake news detection, in Proc. ACL 2017 (Short Papers), (pp. 422–426).

[20]. Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2020). FakeNewsNet: a data repository with news content, social context, and spatiotemporal information for studying fake news on social media, Big Data, vol. 8, no. 3, (pp. 171–188).

IJEAST

INTERNATIONAL JOURNAL OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY

AUTHORS

- Mithilesh Meshram
- Gyaneshwar Tiwari
- Vishwajeet Sutar
- Aditya Raut
- Prof. Archana Bhosale

ABOUT IJEAST

International journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high - quality research paper in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

JOURNAL METRICS

H-INDEX	i10-INDEX	IJEAST CITATIONS	PEER-REVIEWED & OPEN ACCESS
33	154	9,647 (IJEAST CITATIONS)	PUBLISHED MONTHLY

PEER REVIEWED

All submission are rigorously peer reviewed to ensure quality.

AUTHORS ARE OUR PRIORITY

We value our authors and recognize their contribution to the advancement of research and knowledge.

OPEN ACCESS

Free and unrestricted access to research for all.

GLOBAL REACH

Connecting researchers and professionals worldwide.

TIMELY PUBLICATION

We ensure a swift and efficient publication process.

For more information, visit our website
www.ijeast.com

editor@ijeast.com
www.ijeast.com
India

2455-2143