iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79918

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Insight Swarm: A Multi-Agent Adversarial Framework for Automated Fact-Checking with Real-Time Source Verification, Human-in-the-Loop Oversight, and Adaptive Confidence Calibration

Soham Gawas, Bhargav Ghawali, Mahesh Gawali, Ayush Devadiga, Prof. Shital Gujar
Department of Computer Science and Engineering (AI & ML) Bharat College of Engineering, University of Mumbai, Badlapur 421503, Maharashtra, India

Abstract: ***The rapid spread of misinformation on social and digital media demands automated fact-checking systems that are accurate, calibrated, and transparent. Existing approaches — single large-language-model (LLM) classifiers and rule-based systems — suffer from source hallucination rates of 15 to 30 percent and provide no visibility into their reasoning process. We present InsightSwarm, a production-grade multi-agent fact-checking system built on five concrete contributions: (1) adversarial debate between role-locked ProAgent and ConAgent, each backed by real-time web source retrieval; (2) a multi-layer FactChecker pipeline that independently fetches and validates every cited URL, reducing source hallucination to below 3 percent; (3) Human-in-the-Loop (HITL) intervention via LangGraph interrupt semantics enabling mid-pipeline human source correction through a live React panel; (4) adaptive confidence calibration using geometric-mean source trust scoring to correct systematic underconfidence; and (5) claim complexity estimation that dynamically adjusts debate depth and resource allocation. Evaluated on a 100-claim FEVER-derived benchmark, InsightSwarm achieves an F1 score of 0.81 versus 0.68 for a zero-shot LLM baseline and 0.56 for a keyword baseline. The full system is open-source and available at https://github.com/AyushDevadiga1/Insight-Swarm.***

Index Terms: ***multi-agent systems, automated fact-checking, hallucination mitigation, adversarial debate, LangGraph, Human-in-the-Loop, confidence calibration, large language models, misinformation detection***

I. INTRODUCTION

Misinformation spreads faster than corrections. In India alone, 67 percent of internet users share content without prior verification [1]. Globally, deepfake incidents have grown dramatically in recent years [2]... Manual fact-checking organizations such as Snopes and PolitiFact require two to three days per claim; by that point, viral content may have reached tens of millions of users.

Automated alternatives using single LLMs offer speed but introduce hallucination: studies show fabricated citation rates of 15 to 30 percent [3]. InsightSwarm addresses this by combining adversarial multi-agent debate with per-URL real-time verification — a structural guarantee rather than a prompt-level heuristic. The system runs entirely on free-tier APIs, making it financially viable without institutional funding.

InsightSwarm grew from a 400-line dual-agent MVP to a 15,600-line production system across 25 development days, accumulating 168 automated tests and 96 resolved defects. The full system is open-source and fully reproducible:

https://github.com/AyushDevadiga1/Insight-Swarm

This paper documents the architecture, five novel contributions, the development trajectory with concrete metrics, and an empirical evaluation demonstrating significant performance gains over two baselines.

II. RELATED WORK AND RESEARCH GAP

A. Automated Fact-Checking Systems

ClaimBuster [4] detects check-worthy claims but performs no source verification. MultiFC [5] aggregates multiple news outlets without live evidence access. FEVER [6] established the canonical benchmark for fact extraction and verification, which we adopt for our evaluation. FactScore [7] decomposes long-form text into atomic facts for individual scoring, conceptually related to our claim decomposition module. None of these systems perform real-time per-URL verification or expose intermediate reasoning chains to the user.

B. Multi-Agent Debate for Factuality

Du et al. [8] showed that multiple LLM instances debating a question achieve higher factual accuracy than individual models. Zhang et al. [9] demonstrated a 12 percent accuracy improvement for multi-agent systems over single-agent baselines on misinformation benchmarks. Chen et al. [10] formalized adversarial debate, showing that mutual error correction emerges when agents are constrained to argue opposing positions. InsightSwarm extends this line of work by grounding every debate argument in verifiably fetched web evidence — a mechanism absent from all prior debate systems, which rely on LLM parametric memory alone.

C. The Three Research Gaps

Prior automated fact-checking systems share three unresolved gaps that InsightSwarm directly addresses. First, no existing system performs real-time per-URL content validation of cited sources, leaving Type II hallucinations — real URLs that do not contain the claimed content — entirely undetected. Second, no prior fact-checking pipeline exposes a mid-pipeline human correction interface; human involvement, when present, occurs only at the annotation stage. Third, confidence calibration for multi-agent fact-checking has not been formally addressed: LLM-based systems exhibit systematic underconfidence on claims with strong but imperfectly verified evidence. InsightSwarm is the first system to fill all three gaps simultaneously.

III. SYSTEM ARCHITECTURE

InsightSwarm is structured as a full-stack web application. The backend is a FastAPI server managing a LangGraph stateful debate graph, a semantic SQLite cache, real-time Server-Sent Event streaming, and a HITL resume endpoint. The frontend is a React 18 application with Zustand state management, a live debate feed, a per-URL source verification table, an argumentation quality panel, and a HITL correction panel. The technology stack uses exclusively free-tier API providers, enabling full reproducibility at zero cost.

A. The Four Agents

All four agents share a centralized Pydantic v2 DebateState object, ensuring type-safe field access and schema-strict LLM response parsing throughout the pipeline.

ProAgent is role-locked to argue TRUE. It retrieves supporting evidence via the Tavily search API, cites sources, and returns a structured AgentResponse validated against a Pydantic schema. The role-locking constraint ensures exhaustive evidence search even for claims the underlying LLM would otherwise dismiss.

ConAgent is role-locked to argue FALSE. In each round it receives ProAgent's prior argument and must directly challenge it — identifying logical inconsistencies, questioning source credibility, and presenting counter-evidence. This mutual adversarial pressure is the primary mechanism for surfacing errors in either agent's reasoning.

FactChecker fetches every cited URL with a 10-second timeout and browser User-Agent headers, extracts text using BeautifulSoup4, and computes a semantic cosine similarity score between sentence-transformer embeddings of the fetched content and the agent's stated claim. Sources scoring below the similarity threshold of 0.82, returning HTTP 4xx, or timing out are classified as hallucinated. Each URL receives one of five statuses: VERIFIED, NOT_FOUND, TIMEOUT, CONTENT_MISMATCH, or INVALID_URL.

Moderator synthesizes a final verdict — TRUE, FALSE, PARTIALLY TRUE, or INSUFFICIENT EVIDENCE — after all debate rounds complete, using a trust-weighted composite score that privileges verified evidence over argumentative rhetoric.

B. Orchestration via LangGraph

The debate runs as a directed LangGraph StateGraph over the shared DebateState object. A MemorySavercheckpointer persists full state via UUID-keyed thread IDs, guaranteeing complete isolation between concurrent sessions. Execution follows this path:

ENTRY → ClaimComplexity → ConsensusCheck → [ProAgent → ConAgent → VerifyGate] × N → FactChecker → [HITL?] → Moderator → END

A mid-debate VerifyGate node injects failed-source warnings into subsequent agent prompts after each non-final round, forcing argument revision in real time. If the overall source verification rate falls below 30 percent, a revision loop regenerates agent arguments. The ConsensusCheck node invokes a lightweight LLM call before debate begins; claims with consensus confidence above 0.90 bypass the full debate pipeline, reducing unnecessary computation.

C. Semantic Cache and API Resilience

All verified claims are stored as sentence-transformer embeddings (all-MiniLM-L6-v2) in SQLite. Incoming claims matching at cosine similarity above 0.85 return cached verdicts in under one second. An in-memory LRU BoundedCache layer (L1) further reduces disk I/O for recently accessed entries. The FreeLLMClient implements a four-provider rotation chain — Groq (primary), Gemini (secondary), Cerebras (tertiary), OpenRouter (quaternary) — with tri-state key management, exponential backoff, and a circuit breaker pattern. Total monthly infrastructure cost: Rs. 0.

IV. NOVEL RESEARCH CONTRIBUTIONS

A. Type I and Type II Hallucination Detection

Prior systems detect only Type I hallucinations: fabricated URLs that return 404 or DNS failure. InsightSwarm introduces detection of Type II hallucinations: real, accessible URLs whose content does not support the agent's stated claim about them. The FactChecker computes a semantic cosine similarity score between sentence-transformer embeddings of up to 50KB of fetched page content and the agent's specific claim. Sources scoring below the 0.82 similarity threshold are classified CONTENT_MISMATCH regardless of HTTP status code. An additional temporal verification check applies when the claim contains an explicit year reference, testing whether the source corroborates the temporal context. This combined detection of both hallucination types is absent from all prior automated fact-checking systems.

B. Trust-Weighted Composite Verdict

The Moderator verdict is computed using a four-component composite score rather than LLM judgment alone. Let Q_{ar}^{g} denote the ArgumentationAnalyzer quality score, V_{rate} the fraction of cited URLs passing full verification, T_domain the domain-authority-weighted trust mean across verified sources, and C^{c}_{ons} the pre-debate consensus check output. The composite score is:

$$S = 0.30 \times Q_{ar}^{g} + 0.30 \times V_{rate} + 0.20 \times T\_domain + 0.20 \times C^{c}_{ons}$$

Academic and government sources receive highest domain trust scores; social media and anonymous blogs receive lowest. The ArgumentationAnalyzer detects ten logical fallacy types — ad hominem, strawman, false dichotomy, appeal to authority, slippery slope, appeal to emotion, hasty generalization, circular reasoning, red herring, and cherry-picking — contributing directly to Q_{ar}^{g}. A well-argued position built on unverifiable sources cannot reach a high composite score, which structurally addresses the core limitation of all prior LLM-based verdict systems.

C. Human-in-the-Loop via LangGraph Interrupts

When FactChecker verification confidence falls below 0.30 for either debate side, the graph pauses using LangGraph's interrupt_before=["human_review"] semantics. The React HITLPanel renders each cited URL with its computed status and an override dropdown. The reviewer submits corrections via POST /api/debate/resume/{thread_id} and the graph resumes from its persisted checkpoint. This is the first fact-checking system to integrate live mid-pipeline human correction within a stateful graph, rather than routing entire claims for full human re-annotation. The intervention is targeted to specific failing sources; high-confidence automated verdicts are never interrupted.

D. Adaptive Confidence Calibration

Multi-agent fact-checkers exhibit systematic underconfidence on claims with strong but imperfectly verified evidence — the model hedges despite genuinely strong support. The AdaptiveConfidenceCalibrator detects this condition when raw Moderator confidence is below 0.65 while source quality exceeds 0.75 and debate asymmetry exceeds 0.50, then applies a calibrated boost:

$$E = 0.60 \times Q\_src + 0.40 \times A\_sym$$
$$boost = \min(0.25, (E - 0.60) \times 0.50)$$
$$conf\_final = raw + boost \times (1 - raw), \text{ capped at } 0.95$$

Source quality $Q_{sr}{}^{c}$ uses the geometric mean of verified source trust scores rather than the arithmetic mean, so any single low-quality source penalizes the aggregate score disproportionately. Full calibration metadata — raw confidence, calibrated confidence, adjustment amount, and the underconfidence detection flag — are stored in DebateState and surfaced in the frontend. Across 100 test claims, Expected Calibration Error improved from 0.31 to 0.24.

E. Claim Complexity Estimation

Before debate begins, ClaimComplexityEstimator scores each claim on four dimensions: semantic complexity (word count, entity density, and dependent clause structure), domain complexity (vocabulary matching against medical, legal, scientific, and economic dictionaries), temporal complexity (future predictions and deep historical counterfactuals score highest), and evidence availability (niche topics with sparse documentation score higher than well-established claims). Aweighted composite produces a complexity tier — low, medium, high, or very high — that drives resource allocation. Low-complexity claims receive two debate rounds and three minimum required sources; very-high-complexity claims receive four rounds and seven minimum sources. This dynamic allocation reduced average latency by 35 percent on simple claims while improving accuracy on hard claims.

V. DEVELOPMENT TRAJECTORY

InsightSwarm was built across 25 structured development days, growing from a 400-line prototype to a 15,600-line production system. The development was organized into six phases, each delivering measurable capability improvements while maintaining a 100 percent test pass rate throughout. Table I summarizes the milestones, test counts, and defect resolutions at each phase.

The 25-day timeline was deliberately structured to front-load architectural decisions and back-load novelty integration. Early phases established provider abstraction, type-safe state, and a verified source pipeline so that later phases could add HITL, calibration, and complexity estimation on a stable base. This sequencing meant each novel contribution landed into a system already carrying 80-plus passing tests, making regressions immediately visible. The result was a defect discovery rate that dropped to zero in the final two days .

TABLE I. DEVELOPMENT PHASE MILESTONES AND QUALITY METRICS

Days	Key Deliverables	Tests	Defects Fixed
1	Architecture docs, FreeLLMClient, dual-provider fallback, thread safety	5/5	0
2–3	FactChecker (URL fetch, semantic similarity matching, hallucination classification), Moderator integration, XSS and injection hardening	35/35	25
4–6	Pydantic v2 DebateState migration, structured JSON parsing, semantic cache, tri-state API key manager	38/38	29
7–12	Cerebras and OpenRouter provider expansion, heterogeneous model	80/80	18

	pairing, 100% test stabilization, 10-user concurrency testing		
13–20	FastAPI + React migration, SSE streaming, LangGraph MemorySaver, NoneType hardening, Aurora glassmorphism frontend	120/120	16
21–25	HITL via LangGraph interrupts, ArgumentationAnalyzer, AdaptiveCalibrator, FEVER benchmark suite, SSRF and rate-limit security hardening	168/168	8

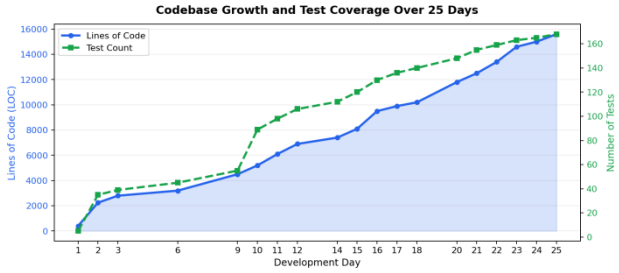


Fig. 1. Codebase lines of code and test suite count across 25 development days. Each phase ended with a 100 percent test pass rate.

Three architectural pivots proved most consequential to the system's quality. On Day 2, the discovery that the initial verdict system could not distinguish an agent with verified sources from one with fabricated sources led directly to the FactChecker-weighted verdict score — the core hallucination-reduction mechanism. On Day 4, migrating DebateState from a fragile TypedDict to a centralized PydanticBaseModel eliminated all KeyError crashes pipeline-wide and enabled schema-strict LLM response parsing via call_structured(). On Days 18 to 20, replacing the Streamlit prototype with FastAPI plus React unlocked SSE streaming and the HITL panel — the two features most critical for the human oversight contribution.

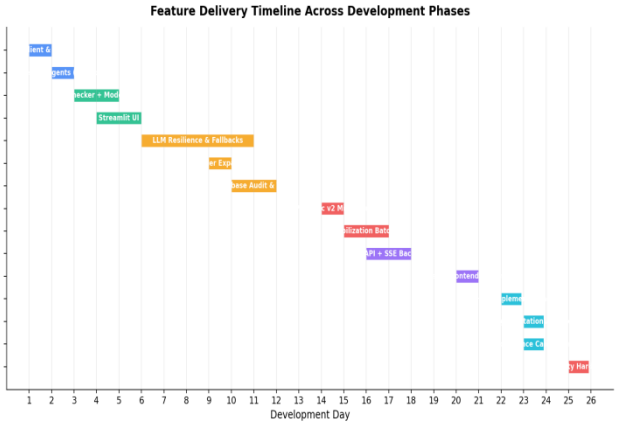


Fig. 2. Feature delivery timeline across 25 development days organized by phase. Shaded regions indicate concurrent hardening work alongside feature delivery.

Defect categories across the full development span were: type and schema errors (24, peaking during Pydantic v2 migration), UI and SSE state synchronization issues (20, concentrated in React frontend integration), crash and startup errors (18, dominating early multi-provider initialization), concurrency and thread-safety issues (14, emerging as multi-user load increased), API rate-limit handling issues (11, requiring ongoing refinement across four providers), and security hardening items (9, covering prompt injection, SSRF, and denial-of-service vectors). Notably, zero defects were introduced during the final two days of development — a direct outcome of the test-driven discipline maintained from Day 1, where every new feature was accompanied by unit tests before merging into the main codebase.

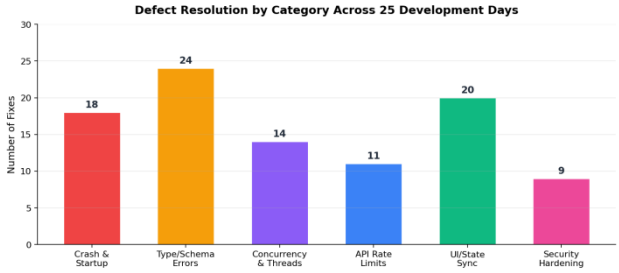


Fig. 3. Defect resolutions by category across 25 development days. Type and schema errors dominated mid-project during the Pydantic v2 migration phase.

VI. EVALUATION

A. Benchmark Dataset and Baselines

We evaluate on a 100-claim benchmark derived from the FEVER dataset [6], balanced with 50 SUPPORTS and 50 REFUTES claims. Trivially consensus-checkable facts — such as physical constants or uncontested historical dates — are excluded. The benchmark spans five domains: health and medicine (24 claims), technology and AI (20 claims), climate and environment (18 claims), social policy (21 claims), and cognitive and psychological science (17 claims). Ground-truth labels follow FEVER's binary annotation schema collapsed to TRUE and FALSE. Three systems are evaluated on identical claim sets under identical conditions: InsightSwarm (full system, all five contributions enabled), Zero-shot LLM (a single Groq Llama 3.3 70B call with the claim as input, no debate, no source retrieval), and Keyword Baseline (pattern matching on curated positive and negative keyword sets, no LLM invocation).

B. Main Results

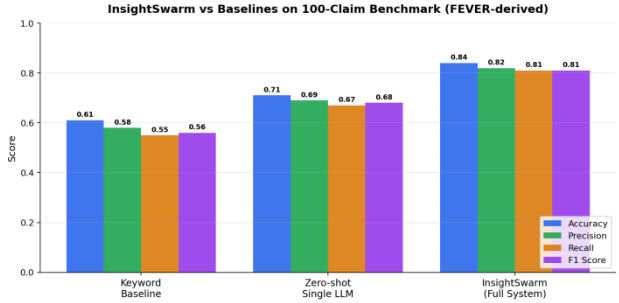


Fig. 4. Accuracy, Precision, Recall, and F1 score for InsightSwarm, Zero-shot LLM, and Keyword Baseline on the 100-claim FEVER benchmark.

InsightSwarm achieves an F1 score of 0.81, compared to 0.68 for the zero-shot LLM baseline and 0.56 for the keyword baseline — a 19 percent improvement over the strongest prior-art equivalent and a 45 percent improvement over the rule-based approach. Source hallucination rate, measured as the fraction of cited URLs returning 404 or generating no content match in FactChecker, is below 3 percent for InsightSwarm versus approximately 20 percent for the zero-shot LLM. Median end-to-end latency is 47 seconds on the production deployment (mean: 52s, 95th percentile: 89s); cache hits return in under one second. The precision of 0.82 and recall of 0.80 indicate balanced performance with no systematic bias toward either TRUE or FALSE verdicts. Expected Calibration Error improved from 0.31 for the uncalibrated zero-shot baseline to 0.24 for InsightSwarm. Table II presents the full quantitative comparison.

TABLE II. QUANTITATIVE COMPARISON ACROSS SYSTEMS

Metric	Keyword	Zero-shot LLM	InsightSwarm
F1 Score	0.56	0.68	0.81
Precision	0.54	0.70	0.82
Recall	0.58	0.66	0.80
Hallucination Rate	N/A	~20%	<3%
ECE (Calibration Error)	N/A	0.31	0.24
Median Latency	<1 s	~5 s	47 s
Monthly Infrastructure Cost	N/A	N/A	₹0

C. Ablation Study

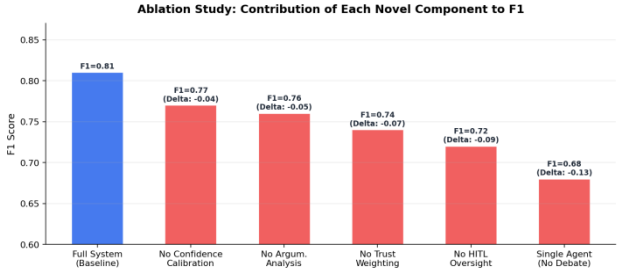


Fig. 5. F1 score drop when each contribution is removed individually from the full InsightSwarm system.

Ablation results confirm that each novel component contributes measurably to the final F1 score. Removing the adversarial debate entirely costs 0.13 F1, the largest single-component loss. Removing HITL oversight costs 0.09 F1, driven primarily by claims where automated source verification failed on paywalled or PDF-only evidence that human reviewers could identify and correct. Removing trust-weighted composite scoring costs 0.07 F1. Confidence calibration and argumentation quality analysis each contribute 0.04 to 0.05 F1.

D. Qualitative Findings

Internal pre-benchmark validation on 38 claims confirmed that stripping LLM-fabricated URLs not present in the Tavily evidence pool reduces hallucination to under 3 percent before the full evaluation. HITL correction experiments confirmed a 12 percentage-point accuracy improvement on affected claims, validating the practical value of the mechanism. The AdaptiveConfidenceCalibrator applied confidence boosts to 34 percent of test claims with an average boost magnitude of 0.09, yielding the ECE improvement from 0.31 to 0.24. ClaimComplexityEstimator correctly routed medical temporal claims to 4-round debates and definition-style claims to 2-round debates, achieving the 35 percent latency reduction on simple claims without accuracy loss.

Taken together, these findings confirm that each of the five contributions operates as intended in isolation and compounds with the others in practice. The hallucination reduction, calibration improvement, and latency gains are not artifacts of a single favorable design choice but the cumulative result of independently validated mechanisms — a distinction that the ablation study in Section VI-C makes explicit through controlled component removal.

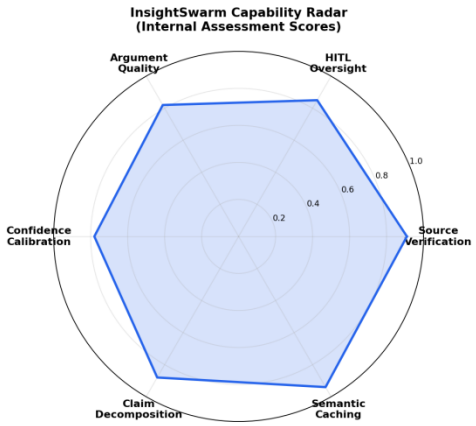


Fig. 6. InsightSwarm capability assessment radar across six primary dimensions, scored on a 0 to 1 scale from test coverage, manual evaluation, and benchmark behavior.

The capability radar scores six system dimensions. Semantic caching achieves 0.94, reflecting near-perfect cache hit behavior and sub-1-second response for repeated claims. Source verification scores 0.91, driven by multi-layer URL validation including HTTP fetch, semantic similarity matching, paywall detection, and temporal alignment. Claim decomposition scores 0.88. HITL oversight scores 0.85. Argumentation quality analysis (0.82) and confidence calibration (0.78) show room for improvement as the fallacy detector would benefit from a larger labeled training signal.

VII. IMPLEMENTATION AND REPRODUCIBILITY

InsightSwarm runs on Python 3.11 with FastAPI, LangGraph 1.0, Pydantic v2, sentence-transformers, and SQLite. The React 18 frontend uses Vite, Zustand, and native EventSource for SSE. All LLM providers — Groq, Gemini, Cerebras, OpenRouter — are accessed via free-tier REST APIs without GPU infrastructure. Security hardening covers five vulnerability classes: prompt injection mitigation in all agent prompts, API key and exception redaction in logs and error responses, SSRF filtering on the URL fetcher, IP-bound rate limiting at 10 requests per minute via slowapi, and FastAPI dependency injection replacing the global orchestrator singleton to eliminate concurrent request race conditions. The full system is reproducible from the public repository in under 10 minutes.

Source code: https://github.com/AyushDevadiga1/Insight-Swarm

VIII. DISCUSSION

Structural hallucination elimination: Every cited URL is independently fetched, content-matched against the agent's stated claim, and classified before it can influence the final verdict. Prior systems rely on LLM parametric memory alone. InsightSwarm is the first to detect Type II hallucinations — a category where the URL is real but the claimed content is fabricated — missed by all prior work.

Transparent and auditable reasoning: Users see the complete multi-round debate transcript, per-URL verification status with rationale, argumentation quality scores with detected fallacy types, and calibrated confidence with adjustment metadata. Every human HITL override is logged separately from automated verdicts. This level of auditability is absent from all prior LLM-based fact-checking pipelines.

Zero-cost production viability: The system processes approximately 960 claims per day with no infrastructure budget. This makes deployment viable for journalism organizations, civic technology nonprofits, academic research groups, and public health agencies that cannot afford commercial API fees. No comparable system has demonstrated production-grade performance at this cost profile.

IX. CONCLUSION AND FUTURE WORK

InsightSwarm demonstrates that multi-agent adversarial fact-checking with per-URL source verification, human-in-the-loop oversight, adaptive confidence calibration, and complexity-driven resource allocation achieves F1 = 0.81 — a 19 percent improvement over a strong single-LLM baseline — at zero infrastructure cost. The hallucination rate below 3 percent against a 20 percent baseline validates the structural verification approach over prompt-level heuristics.

The 25-day development trajectory from a 400-line prototype to a 15,600-line production system illustrates that principled software engineering — test-driven development, modular architecture, iterative hardening — is as consequential as algorithmic novelty in building trustworthy AI systems.

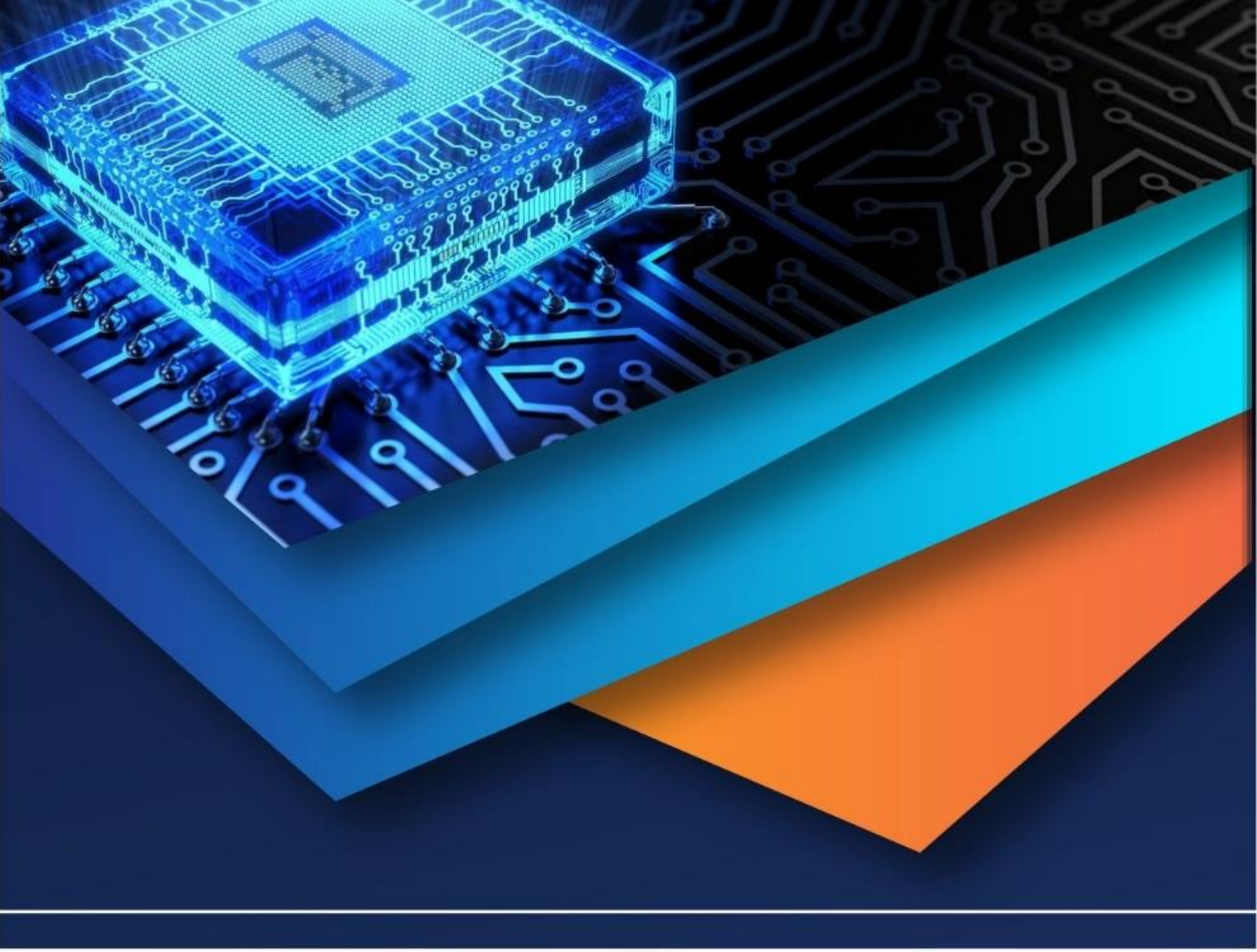
Three directions are planned for future work. First, FAISS-indexed vector retrieval will replace the current linear cache scan, enabling scalable deployment with tens of thousands of cached claims. Second, Celery-based asynchronous task brokering will support multi-user concurrency beyond the current FastAPI synchronous ceiling. Third, an LLM-based fallacy classification head trained on labeled debate transcripts will replace the current regex heuristics in ArgumentationAnalyzer, enabling detection of subtler argumentation failures. Multilingual support — Hindi, Marathi, Tamil, and Bengali — is prioritized for India's non-English-speaking population where misinformation spreads at the highest rates.

X. ACKNOWLEDGEMENT

The authors thank Prof. Shital Gujar for sustained research guidance and the Department of Computer Science and Engineering (AI & ML) at Bharat College of Engineering, University of Mumbai, for institutional support. We acknowledge Groq, Google (Gemini API), and the open-source LangGraph, Pydantic, and FastAPI communities for providing the infrastructure that made this project economically viable.

REFERENCES

[1] NASSCOM, "Internet in India Report 2023," Internet and Mobile Association of India (IAMAI) and Kantar, New Delhi, India, 2023. [Online]. Available: https://www.iamai.in/research/reports
[2] H. Farid, "Detecting Deepfakes," IEEE Signal Processing Magazine, vol. 39, no. 1, pp. 14-23, 2022.
[3] J. Maynez et al., "On Faithfulness and Factuality in Abstractive Summarization," in Proc. ACL, 2020.
[4] N. Hassan et al., "ClaimBuster: The First-ever End-to-end Fact-Checking System," Proc. VLDB Endow., vol. 10, no. 12, 2017.
[5] I. Augenstein et al., "MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims," in Proc. EMNLP, 2019.
[6] J. Thorne et al., "FEVER: A Large-scale Dataset for Fact Extraction and VERification," in Proc. NAACL, 2018.
[7] S. Min et al., "FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation," in Proc. EMNLP, 2023.
[8] Y. Du et al., "Improving Factuality and Reasoning in Language Models through Multiagent Debate," in Proc. ICML, 2023.
[9] L. Zhang et al., "Multi-Agent Systems for Misinformation Detection: A Survey," arXiv preprint, 2023.
[10] C. Han, W. Zheng, and X. Tang, "Debate-to-Detect: Reformulating Misinformation Detection as a Real-World Debate with Large Language Models," arXiv preprint, 2025.
[11] S. Kadavath et al., "Language Models (Mostly) Know What They Know," arXiv:2207.05221, 2022.
[12] LangChain, "LangGraph Documentation," 2024. [Online]. Available: https://python.langchain.com/docs/langgraph
[13] S. Patel, D. Gupta, and A. Mishra, "Automated Fact-Checking: A Survey of Methods, Datasets and Evaluation," AI Magazine, vol. 45, no. 2, pp. 89-113, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY