International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

Deploying Production AI Agents in Enterprise Workflows: Observability, Grounding, and Measured Business Outcomes

Rahul Mittal[1]
[1]*Illinois Institute of Technology, Chicago, US*
Email id: *mittalrahul001@gmail.com*[1]

Abstract
LLM agents have evolved from demonstrations to enterprise applications in customer support, software engineering, document processing, and more. But more than benchmark accuracy is required to determine their production value. An agent could choose to use tools, access evidence, alter records, and/or repeat actions, and failures could be due to stale sources, weak authorization, non-convergent planning, duplicated writes, incomplete audit trails, or misleading measurement of outcomes. The paper suggests a five-layered architecture consisting of the following: Workflow plane, Bounded agent runtime, Grounding and knowledge layer, Observability and governance plane, Outcome-measurement control loop. It also creates a seven-level taxonomy of telemetry: sessions, planning steps, tool calls, retrieval, outputs, human interventions and population drift. Grounding is considered as an admission control: material assertions are required to be backed up by legitimate and up-to-date evidence or abstention and escalation. The business value is measured using controlled comparison, task stratification, quality review, intervention accounting and joint reporting of performance/cost. Field experiments show that the use of generative AI has the potential to boost speed, quality, and throughput, however, depending on the experience of the workers and the location of the task within the system's capability range. These principles are then broken down into operational controls and measurement definitions in two tables and integrated with the architecture and feedback loop in one figure. Production AI agents need to be deployed as bounded, observable, evidence-linked systems with permissions, sources, and versions that can be tested continually, and with claims of outcomes.

Keywords: *AI agents; large language models; enterprise automation; observability; retrieval-augmented generation; grounding; hallucination; AI governance; productivity; human oversight*

1. Introduction
The focus of the central enterprise question has shifted from whether LLM's can do useful work, to whether agentic systems can do bounded work reliably, explainably, and economically. An AI agent is defined here as a compound system that includes a model that understands a goal, selects actions, sequences them, invokes external tools, sees the results and may revise its plan. Examples of reasoning-action interleaving, learned tool use, and feedback-based self-correction are ReAct [1], Toolformer [2] and Reflexion [3]. Agents are now described in terms of a number of recurring components: planning, memory, tool use, action and environmental feedback [4]. This ability alters the deployment risk. A conversational model can produce a false answer, an enterprise agent can produce a false record, send an unauthorized message, approve an exception not supported by the system, or repeatedly invoke a expensive service. While conventional model evaluation is still required, it is not enough as failures can come from retrieval, prompts, permissions, tool schemas, orchestration, data freshness and human review. There are three operational concerns that frequently are handled separately. The first question in grounding is whether the claims are backed by valid and up-to-date evidence. Retrieval-augmented generation (RAG) can help access external knowledge [5] but retrieval does not ensure faithful use of evidence. Models might fail

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

to capture relevant content, particularly if such content is not at the beginning or end of long contexts [6]. Second, observability is the question of whether a decision can be reproduced, that is, what model and prompt were used, what evidence was found, what tools were used, and what has changed. Third, outcome measurement asks: Was there a change in cycle time, quality, throughput or cost due to deployment versus a credible baseline? These concerns comprise one control system. Grounding limits assertions, observability makes actions auditable, and measurement will make sure that controlled behavior creates value. Field evidence is favourable with conditions. Controlled studies report increased productivity for professional writing [17] and for customer-support with especially large gains for less experienced workers [18] as well as productivity effects of coding assistants [19]. However, the performance can become poor when users try to use AI beyond the system's limits of capability [20]. Task boundaries and abstention rules are thus necessary for production programs, as well as more capable models. This review has three contributions. It suggests a five-layer structure for enterprise agents, introduces a classification of observability based on failures and introduces an outcome-based structure that combines technical telemetry with business measurement. The aim is not to develop a new model benchmark, but a synthesis that can be implemented.

2. Review Scope and Method

The review includes peer-reviewed papers, working papers that describe field experiments, and authoritative governance or technical standards from 2020 to 2025. The searches included large-language-model agents, the use of tools, RAG, hallucination, long-context behavior, RAG evaluation, LLM-as-judge approaches, AI productivity, agent evaluation, documentation of datasets and models, AI risk management, management systems, tracing, and security guidance. Studies were chosen if they provided a mechanism, evaluation method, empirical effect or governance requirement directly relevant to production deployment. The synthesis has been broken down into four evidence streams. Planning and tool interaction is explained in the agent-systems literature [1-4]. The literature on grounding covers retrieval, evidence use, hallucination and evaluation [5-10]. Field experiments provide estimates of business outcomes [17-20]. Governance and documentation frameworks establish accountability, risk, logging and oversight requirements [11-16,21-25]. The proposed architecture is not necessarily the best, as stated in the review. It is based on controls from known failure modes and correlates them to recent evidence. Effect sizes are provided as a way to suggest reasonable magnitudes, not to predict a particular deployment.

3. Production Risks in Agentic Workflows

3.1.Compound-system failure

The output of an agent is produced by an interacting system and not just by a model. The final result is influenced by model selection, prompt and policy versions, memory, retrieved context, tool schemas, authorization, external-system behavior, retry logic and human intervention. Production regression could be due to a model benchmark staying the same, while a corpus update, permission change or tool contract is causing regression. Planning loop adds more failure modes. An agent may be able to alternate between actions, use up a step budget, try a step that is not idempotent or that has an ambiguous response from the tool and consider it successful. Some tasks can be improved with self-correction [3] but unconstrained reflection also leads to an increase in latency, cost and potential for divergence. Production runtimes should thus have finite step limits, deadlines, typed tool arguments, explicit terminal states, rollback or compensation for mutating actions.

3.2.Failure of evidence and grounding

RAG helps to link the generation to external corpora, and allows for knowledge updates without the need to retrain the entire model [5]. But, retrieval scope, source quality, version control, ranking, context assembly and claim-level usage are all crucial for system reliability. Relevant information may be underused based on its position as evidenced in long context research [6]. Continuing challenges also identified by RAG surveys are retrieval quality, integration, evaluation

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

and efficiency [7]. Hallucination studies have identified four categories: unsupported material, factual errors, source conflict and attribution failures [8,9]. In enterprise environments, in addition to factuality, currency and authorization are important. A source may be correct, but outdated, or it may be up-to-date but beyond the user's authority. Grounding therefore needs to check the provenance of the evidence, the version of the evidence, access entitlement and relevance to the response before releasing the response. Automated RAG evaluation offers context relevance, faithfulness, and answer relevance metrics [10]. The model-based judges are able to scale the evaluation, but they may not like it or agree with the human assessment [11]. Automated scoring should be calibrated using human labeled samples specific to the task and not regarded as fact.

3.3.Failure of humans and organisations

Not all human review safeguards are created equal. The reviewers could rubber-stamp reasonable results, or underestimate automation error, or rework that is not measured. Bender et al. highlight the dangers of fluent language without an underpinning understanding [12]. Documentation approaches like datasheets [13], foundation-model reporting [14] and internal algorithmic auditing [15] can be used to explain data, limitations, intended uses and organisational responsibility. Security threats are timely injection, over-agency, disclosure of sensitive data, supply chain vulnerabilities and insecure handling of output. These risks are then turned into concrete threat categories by the OWASP guidance for LLM applications [16]. Security controls need to be at the tool and data boundaries and not just in natural-language instructions.

4. Five-Layer Reference Architecture

Figure 1 organises production control into five connected layers. Each boundary is both a validation point and a telemetry point. As shown in Figure 1 Five-layer reference architecture connecting bounded workflow execution, grounding, observability, governance, and outcome measurement.

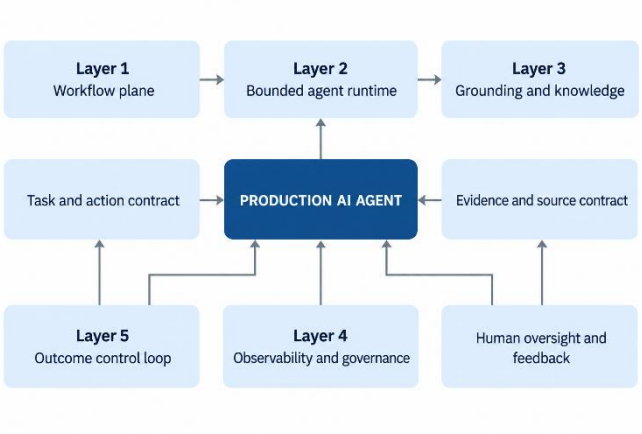


Figure 1 Five-layer reference architecture connecting bounded workflow execution, grounding, observability, governance, and outcome measurement

4.1.Workflow plane

The business task is specified by the workflow plane. The main artifact of the same is a task contract, which defines the allowed inputs, the desired output schema, the systems to which it is allowed to be granted, the things it can't do, the service levels, escalation conditions, and ownership. Task contracts are a way of making the validated capability envelope executable and not descriptive, which helps to avoid scope creep. There should be a separation between read and write actions. Retrieval can be automatic, depending on the access controls granted, while writes to systems of record can be previewed, confirmed or even approved by a human. Where possible, high impact actions should be reversible, and should have a correlation identifier [16].

4.2.Bounded agent runtime

The runtime plans, acts and observes within limits. Controls are such things as maximum steps, time and cost budgets, typed schemas, least-privilege credentials, idempotency keys, deterministic validators, safe retry rules and explicit terminal states. Tool outcomes should differentiate successful completion, partial completion, rejection and unknown. Policy decisions should be coded or controlled configuration in the runtime. A message from the agent to not use a restricted tool is less effective than an authorization layer that prevents the agent from using a restricted tool.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

Likewise, a "don't duplicate write" is less stringent than an idempotency key enforced by the destination service [17].

4.3.Layer of grounding and knowledge

The ingestion, access controlled indexing, retrieval, reranking, versioning, claim attribution and source currency is handled by the grounding layer. Corpus objects must have stable identifiers, version hashes, owners, effective dates, access metadata and retirement status. If there are conflicting versions, they should not be silently equivalent [18]. Response-level scoring is not recommended as one "unsupported" response may be critical to the claim. The system should demand the submission of evidence which is authorised, up to date and specific enough for regulated or high impact assertions. Failure should result in abstention or escalation, rather than an "embarrassment of riches" synthesis with inadequate evidence [19].

4.4.Observability and governance plane

The observability plane captures causal traces from user request to planning steps to retrieval to tools to validation to response to human review. It should record version identifiers for model, prompt, tool schema, policies, evaluators and corpus snapshot. This allows to carry out a regression analysis and audit [20]. The same evidence is used for governance. The AI Risk Management Framework from NIST breaks down activities into four categories: govern, map, measure, and manage [21]. The EU AI Act introduces provisions for risk management, logging, documentation, data governance, human oversight, and post-market monitoring of relevant AI systems [23] and ISO/IEC 42001 introduces a structure for an AI management system [22]. Operational telemetry should be engineered to meet engineering and governance requirements.

4.5.The control loop for measuring the outcome

The fifth layer correlates system traces to business results. It provides a comparison to pre-registered baselines or holdouts, stratifies results by task and user group, assesses quality, considers human review and provides an estimate of cost per accepted outcome. Governance then determines if the deployment should be scaled, revised, constrained or retired.

5. Observability Taxonomy

Agent observability must explain what happened, why it happened, what evidence supported it, and whether the outcome was acceptable. As shown in Table 1 organizes telemetry by unit of capture.

Table 1 Failure-driven observability taxonomy for production AI agents

Level	Unit of capture	Required signals	Failure detected	Operational response
L1 Session	One business task end to end	Session ID, task class, actor, terminal state, duration, total cost	Silent non-completion, excessive latency, unattributed spend	Alert on missing terminal state; reconcile with system of record
L2 Planning step	One plan-act-observe iteration	Step number, selected action, budget remaining, replans, state transition	Looping, oscillation, budget exhaustion	Stop, escalate, or switch to safe fallback
L3 Tool invocation	One external call	Tool, typed arguments, authorization,	Invalid arguments, unauthorized	Block request, compensate transaction,

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

		status, latency, retry, idempotency key	scope, duplicate writes	review permissions
L4 Retrieval and evidence	One evidence set per claim	Query, document IDs and versions, scores, access decision, citation link	Stale, unauthorized, irrelevant, or unused evidence	Re-retrieve, abstain, escalate, flag corpus owner
L5 Output	One returned artifact	Schema result, policy result, groundedness, uncertainty, abstention	Contract violation, unsupported assertion, unsafe output	Suppress, regenerate within limits, route to review
L6 Human interaction	One review event	Accept, edit, reject, review time, edit distance, override reason	Automation bias, hidden rework, systematic correction	Increase audit sampling; revise task boundary or prompts
L7 Population and version	Aggregated distribution over time	Task mix, refusal, retrieval hit rate, cost, quality, model and prompt versions	Drift, silent regression, workload shift	Roll back, recalibrate thresholds, revalidate deployment

Four implementation principles make these signals useful. First, correlation must be structural. A single trace identifier should propagate across orchestration, retrieval, tools, human review, and the relevant business record. Vendor-neutral trace context and semantic conventions can support this design [24]. Second, evidence should usually be retained by reference. Storing all retrieved passages can duplicate personal or confidential information. Stable document identifiers, versions, access decisions, scores, and content hashes may support reconstruction while limiting unnecessary retention. Retention rules should reflect legal and business requirements. Third, online evaluators require calibration. RAGAS and related methods can continuously assess faithfulness and relevance [10], but judge-based metrics should be checked against blinded human review [11]. Thresholds should be task-specific, and changes to evaluator prompts or models should be versioned like production code. Fourth, human interaction is primary telemetry. Accept rates alone are misleading because an accepted output may have required extensive editing. Review time, edit distance, escalation, and rejection reason reveal hidden work. A decline in review time accompanied by rising audit errors may indicate automation bias rather than system improvement.

6. Grounding as an Admission Control

Grounding should be used as a release condition, NOT as an optional quality enhancement. The policy should specify what types of claims must be backed up, what classes of sources are allowed, what it means by "currency," how to deal with conflicting sources and when the agent must sit down. The system should break down an output into verifiable claims, and link each material claim to evidence. Evidence must be relevant, available

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

to the user that initiated the evidence, valid for the date or jurisdiction, and available for version recovery. But the presence of a citation is not enough: a citation may be irrelevant, partially supportive or related to a claim that the source does not actually make [15]. Thus, Corpus governance is included in agent governance. Every collection should have an owner, an inclusion rule, a refresh schedule, a deprecation process, an access-control source and a quality review. Corpus owners should receive signals from retrieval failures that lead to action. Increases in refusal may signal decreasing coverage and decreases in refusal with stable coverage may signal decreasing thresholds [16]. Grounding also helps to reduce data leakage. Access control should be implemented at the time of retrieval, not after generation, as filtering after generation is not an effective way of removing information a model has already processed. The same should apply to the results of tools and memory stores.

7. Measuring Business Outcomes

Technical success does not create business value. It's possible to have low latency and high groundedness in a deployment and also add review burden or automate a low value step. Prior to rollout, the unit of work, comparison group, time window, quality rubric and cost boundary should be defined for the outcome evaluation. Field experiments show that there is heterogeneity. Noy and Zhang discovered that generative AI saves time and enhances quality in professional writing activities [17]. Brynjolfsson et al. found that there was greater productivity in customer support, and that the productivity improvements were greater for less experienced employees [18]. Coding-assistance experiments revealed significant speed effects, but the results differ in the amount of the speed-up depending on the task and environment [19]. Dell'Acqua et al. demonstrated that consultants were more likely to perform well within an AI capability frontier, and less well outside the frontier [20]. These results suggest that controlled adoption is better than universal automation. As shown in Table 2. Business outcome framework for enterprise agent deployments.

Table 2 Business outcome framework for enterprise agent deployments

Outcome	Operational definition	Data source	Analysis requirement	Interpretation risk
Cycle time	Median time from valid trigger to accepted artifact for matched task classes	Session traces and system-of-record timestamps	Compare baseline and exposed groups; report distribution, not mean alone	Faster generation can hide slower review
Throughput	Accepted units per paid worker-hour within a fixed staffing boundary	Queue, workforce, and acceptance records	Stratify by worker experience and task complexity	Task mix may change after rollout
Quality	Blinded expert score or verified defect rate against a fixed rubric	Sampled outputs, downstream errors, audits	Pre-register rubric and sample; retain blinded review	Automated judges may share model biases
Groundedness	Share of material claims supported	Claim-evidence records and	Report sample size and judge-	Citation presence may be mistaken

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

	by authorized, current evidence	human audit	human agreement	for support
Intervention load	Review, edit, escalation, and rejection time per accepted output	Human-interaction telemetry	Include rework and exception handling	Hidden labor can absorb apparent savings
Capability-boundary risk	Share of tasks outside the validated task contract	Task classifier, abstentions, audit outcomes	Compare errors inside and outside approved scope	Scope creep may not change system version
Unit economics	Fully loaded cost per accepted output	Model/tool cost, infrastructure, labor, failure cost	Report cost jointly with quality and completion	Cheap but low-quality automation appears efficient

There are five commitments that are required to make a credible evaluation. Before deployment use the same definition of metrics and task strata to determine the baseline. Second, keep a control or phased-rollout group until learning, novelty and task-mix effects have been captured. Third, stratify according to task complexity and worker experience as average effects can mask benefit and harm. Fourth, don't forget to include displaced work, such as reviewing, escalating, correcting, monitoring and incident response. Fifth, report both quality and cost. When evaluations of agents do not consider cost, reproducibility and realistic baselines, they can become misleading, as Kapoor et al. contend [25]. If operationally feasible, randomized rollout is preferred. When randomization is not possible, difference-in-differences or matched comparisons can be used, with appropriate assumptions being stated. Uncertainty intervals, exclusions, failed sessions and negative outcomes should be reported in outcome reports. If there is not a causal design in the dashboard, it's not evidence of impact, it's monitoring.

8. Governance, security and change control

Ownership is the first step in governance. There must be a business owner, technical owner, corpus owner, risk owner and accountable approval authority for each deployment. Document the task contract, validated boundaries, model and prompt versions, connected tools, evaluation results, known limitations and human-oversight design. The risks should include harmful content, privacy, IP, discrimination, cyber security, disruption of operation and wrong action. There are multiple approaches to risk: NIST AI RMF is flexible [21]; ISO/IEC 42001 is a management-system approach [22]; and the EU AI Act may have legal requirements, depending on provider, deployer, system classification, and use [23]. The frameworks overlap but can't be substituted. Any content retrieved and user input should be considered to be adversarial. Prompt injection can be used to try to override instructions or to exfiltrate secrets. Tools should be scoped, have allowlists, typed arguments, output validation, and transaction limits. Any action that could have a high impact should be subject to further authorization. OWASP's LLM threat classification provides a useful threat checklist [16]. Model, prompt, policy, tool schema, evaluator, memory design and corpus snapshot should all be considered production configuration in the context of change control. Affected task classes should be identified by the change assessment and the need for regression testing, targeted revalidation or rollback should be identified. Performance of the population before and after deployment by version should be

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

compared using population telemetry.

9. Deployment Sequence

One bounded workflow is the starting point for a disciplined rollout. The organization establishes task and action contracts, records the baseline and failure costs. It then adds session and tool traces prior to adding automation. With Abstention and Escalation, Grounding and Claim Attribution are added. Shadow deployment does not permit writes, and tests recommendations. Human review data are gathered and applied to improve boundaries. A comparison group, explicit stopping rules and incident handling should be considered in a limited production pilot. Scaling should come after the agent has proven to be acceptable in quality, grounding, intervention load, security and unit economics. Growth to neighbouring task classes should be considered as a new validation decision and not ordinary growth.

Conclusion

Production AI agents are not autonomous model capabilities, but are bounded socio-technical systems. To be trustworthy, it is necessary to have an explicit task contract, to limit the runtime, to control the evidence, to have telemetry of human interaction, to have a measurement loop that allows to stop or revise deployment. The five layer architecture has controls at the points where information and authority intersect. The observability taxonomy allows for reconstructable sessions, decisions, tools, evidence, outputs, review and drift. The outcome framework will ensure that an increase in generation does not get confused with an increase in productivity if the quality, review or failure costs have been shifted elsewhere. However, recent evidence indicates that generative AI can yield material benefits, particularly for certain workers and tasks that are a good fit. It also indicates a conditional performance. The sustainable enterprise value therefore does not rely on autonomy as much as it does on establishing boundaries for autonomy, demonstrating the value of each autonomy action, and assessing if the system enhances the entire process.

References

[1]. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. International Conference on Learning Representations.

[2]. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36.

[3]. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 36.

[4]. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J. (2024). A survey on large language model based autonomous agents. Frontiers of Computer Science, 18(6), 186345. https://doi.org/10.1007/s11704-024-40231-1

[5]. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

[6]. Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, 12, 157–173. https://doi.org/10.1162/tacl_a_00638

[7]. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. arXiv. https://doi.org/10.48550/arXiv.2312.10997

[8]. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. ACM

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

Computing Surveys, 55(12), 248. https://doi.org/10.1145/3571730

[9]. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems, 43(2), 42. https://doi.org/10.1145/3703155

[10]. Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAS: Automated evaluation of retrieval augmented generation. Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, 150–158.

[11]. Zheng, L., Chiang, W. L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. Advances in Neural Information Processing Systems, 36.

[12]. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 610–623. https://doi.org/10.1145/3442188.3445922

[13]. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. Communications of the ACM, 64(12), 86–92. https://doi.org/10.1145/3458723

[14]. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. arXiv. https://doi.org/10.48550/arXiv.2108.07258

[15]. Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., & Barnes, P. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 33–44. https://doi.org/10.1145/3351095.3372873

[16]. OWASP Foundation. (2025). OWASP Top 10 for LLM applications 2025. Open Worldwide Application Security Project.

[17]. Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. Science, 381(6654), 187–192. https://doi.org/10.1126/science.adh2586

[18]. Brynjolfsson, E., Li, D., & Raymond, L. R. (2025). Generative AI at work. Quarterly Journal of Economics, 140(2), 889–942. https://doi.org/10.1093/qje/qjae044

[19]. Cui, K. Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., & Salz, T. (2024). The productivity effects of generative AI: Evidence from a field experiment with GitHub Copilot. Microsoft Research.

[20]. Dell'Acqua, F., McFowland, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K. C., Rajendran, S., Krayer, L., Candelon, F., & Lakhani, K. R. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. Harvard Business School Working Paper 24-013.

[21]. National Institute of Standards and Technology. (2023). Artificial intelligence risk management framework AI RMF 1.0. NIST AI 100-1. https://doi.org/10.6028/NIST.AI.100-1

[22]. International Organization for Standardization & International Electrotechnical Commission. (2023). ISO/IEC 42001:2023 Information technology—Artificial intelligence—Management system. ISO.

[23]. European Parliament and Council of the European Union. (2024). Regulation EU 2024/1689 laying down harmonised rules on artificial intelligence. Official Journal of

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5828 - 5837
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0755

the European Union.

[24]. OpenTelemetry Authors. (2025). OpenTelemetry specification. Cloud Native Computing Foundation.

[25]. Kapoor, S., Stroebl, B., Siegel, Z. S., Nadgir, N., & Narayanan, A. (2024). AI agents that matter. arXiv. https://doi.org/10.48550/arXiv.2407.01502