International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 5, September – October 2025 ||

DOI: 10.15680/IJCTECE.2025.0805010

Generative Agents at Scale: A Practical Guide to Migrating from Dialog Trees to LLM Frameworks

Dr. Rashmiranjan Pradhan

AI, Gen AI, Agentic AI Innovation leader at IBM, Bangalore, Karnataka, India

rashmiranjan.pradhan@gmail.com

ABSTRACT: The rapid advancements in Large Language Models (LLMs) are fundamentally transforming conversational AI, enabling the development of more flexible, context-aware, and human-like Digital Virtual Agents (DVAs). While traditional dialog tree-based systems have served as the backbone for many enterprise chatbots, their inherent rigidity and maintenance overhead limit scalability and user experience in complex scenarios. Migrating from these rule-based systems to LLM-driven frameworks presents significant technical and operational challenges, particularly at scale and within highly regulated industries like healthcare, finance, and telecommunications. This paper provides a practical guide for organizations undertaking this migration. We analyze the limitations of dialog trees, the capabilities introduced by LLMs, and propose a structured migration methodology encompassing assessment, planning, phased implementation, testing, and ongoing governance. The paper details practical strategies for leveraging LLM frameworks while maintaining control, ensuring accuracy, managing security and privacy, and integrating with existing enterprise systems. We discuss industry-specific considerations, drawing insights from real-world challenges and widely adopted approaches in healthcare, finance, and telecom. The aim is to equip practitioners with a clear understanding of the steps, considerations, and technical approaches necessary to successfully transition to scalable, intelligent, and effective generative agent architectures.

KEYWORDS: “Generative AI,” "Large Language Models," "Digital Virtual Agents," "Conversational AI," "Migration Strategy," "Dialog Management," "Enterprise AI," "Healthcare IT," "FinTech," "Telecommunications," "AI Deployment," "Practical Guide," " IEEE Standards."

I. INTRODUCTION

Digital Virtual Agents (DVAs) have become ubiquitous across industries, serving as the first point of contact for customer inquiries, internal support, and various transactional tasks. For years, the dominant paradigm for building these agents was based on dialog trees or finite state machines. These systems rely on predefined paths, explicit rules for intent recognition, and templated responses. While predictable and relatively easy to control in simple, narrow domains, dialog trees become exponentially complex to build, manage, and scale as the number of intents, variations, and conversation paths grows. Maintaining these intricate structures is labor-intensive and often results in rigid, unnatural, and frustrating user experiences when conversations deviate from expected flows.

The advent of Large Language Models (LLMs) has introduced a paradigm shift in conversational AI. LLMs possess remarkable capabilities in understanding natural language nuances, maintaining context over longer turns, and generating fluent, coherent, and even creative text. This enables the development of DVAs that can handle more open-ended conversations, understand implicit meaning, and provide more natural and personalized interactions. These "generative agents" promise a leap forward in user satisfaction and the ability to automate more complex tasks.

However, the transition from deterministic dialog trees to probabilistic LLM frameworks is not without significant hurdles. Organizations face challenges related to model controllability, ensuring factual accuracy, managing security and privacy risks associated with generative models, integrating with legacy systems, and establishing robust governance and monitoring frameworks. Migrating existing, often large and complex, dialog tree-based systems to LLM architectures requires careful planning and execution, especially in industries handling sensitive data and operating under strict regulations.

This paper serves as a practical guide for organizations contemplating or undertaking the migration from traditional dialog tree-based DVA architectures to frameworks leveraging LLMs. We provide a structured methodology and discuss practical strategies, technical considerations, and potential pitfalls based on observed trends and challenges in

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 5, September – October 2025 ||

DOI: 10.15680/IJCTECE.2025.0805010

real-world enterprise deployments across key sectors. The goal is to demystify the migration process and provide actionable insights for building scalable, reliable, and effective generative agents.

The key contributions of this paper include:

1. An analysis of the limitations of dialog trees and the capabilities of LLM frameworks in the context of enterprise DVAs.
2. A proposed structured methodology for migrating from dialog tree architectures to LLM-based systems.
3. Practical strategies for integrating LLMs while maintaining control, accuracy, security, and compliance.
4. Industry-specific considerations for healthcare, finance, and telecommunications, highlighting unique challenges and approaches.

II. DIALOG TREES VS. LLM FRAMEWORKS

Understanding the fundamental differences between traditional dialog tree systems and LLM-based frameworks is crucial for planning a migration.

A. Dialog Tree Systems

- **Architecture:** Based on predefined states and transitions. User input is matched against specific rules or patterns to determine the next state and trigger a predefined response or action.
- **Control:** Highly deterministic and predictable. Developers have explicit control over every possible conversation path and response.
- **Development:** Requires manual mapping of intents, entities, and conversation flows. Scales linearly or worse with complexity.
- **Flexibility:** Limited to predefined paths. Struggles with variations in user phrasing, out-of-scope queries, and context switching.
- **Maintenance:** Can become cumbersome to update and manage as the tree grows. Changes in one part can have unintended consequences elsewhere.
- **Strengths:** Predictability, ease of initial implementation for simple tasks, clear control over responses, suitability for structured data collection.
- **Weaknesses:** Rigidity, poor handling of natural language variation, lack of context across turns, difficult to scale for complex domains, unnatural user experience.

B. LLM Frameworks for DVAs

- **Architecture:** Leverages large pre-trained language models for understanding and generation. Often uses orchestration layers (e.g., prompt engineering, retrieval-augmented generation (RAG), function calling) to guide the LLM's behavior and integrate with external systems.
- **Control:** Less deterministic; relies on the LLM's learned patterns. Requires guardrails, prompt engineering, and external logic to constrain behavior and ensure accuracy.
- **Development:** Shifts from mapping explicit paths to defining goals, providing context, and engineering prompts/orchestration logic. Can handle broader domains with less explicit rule definition.
- **Flexibility:** Inherently more flexible in understanding natural language and handling variations. Can maintain context over longer conversations.
- **Maintenance:** Focus shifts to managing knowledge sources, prompt strategies, and guardrail configurations rather than intricate flow charts.
- **Strengths:** Superior natural language understanding and generation, better context handling, more natural and engaging conversations, potential for handling broader or novel queries.
- **Weaknesses:** Lack of inherent control and predictability, potential for generating inaccurate or nonsensical information (hallucinations), security risks (prompt injection), computational cost, need for robust guardrails and orchestration.

The migration is essentially a shift from a rigid, rule-based system to a more flexible, learning-based system that requires different control mechanisms and a new approach to development and governance.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 5, September – October 2025 ||

DOI: 10.15680/IJCTECE.2025.0805010

III. PROPOSED MIGRATION METHODOLOGY

A successful migration from dialog trees to LLM frameworks requires a structured, phased approach. We propose the following methodology:

Phase 1: Assessment and Planning

1. **Assess Existing DVA Landscape:**
   - Inventory current dialog tree-based DVAs, their functions, complexity, and performance metrics (e.g., containment rate, escalation rate, user satisfaction).
   - Analyze conversation logs to identify user pain points with the current system (e.g., frequent escalations due to unhandled intents, repetitive phrasing, inability to handle context).
   - Document integrations with backend systems.
2. **Define Migration Goals and Scope:**
   - Clearly articulate what success looks like (e.g., improved containment, higher user satisfaction, reduced maintenance effort, ability to handle new query types).
   - Determine which existing DVAs or specific functionalities are candidates for migration. Prioritize based on potential impact and feasibility.
   - Define the scope of the initial migration phase (e.g., migrate a specific, well-defined use case first).
3. **Select LLM Framework and Architecture:**
   - Evaluate available LLM models (commercial APIs, open-source models) based on capabilities, cost, security features, and deployment options (cloud, on-premises).
   - Design the target architecture, including orchestration layers (RAG, function calling), guardrail mechanisms, and integration strategy.
4. **Identify Required Data and Resources:**
   - Determine what data is needed for fine-tuning (if applicable), knowledge base population (for RAG), and testing.
   - Assess required technical expertise (prompt engineering, MLOps, security).

Phase 2: Phased Implementation

1. **Start with a Pilot Use Case:**
   - Migrate a single, well-defined dialog tree or create a new LLM-based DVA for a specific function identified in Phase 1.
   - Implement core LLM orchestration logic (e.g., RAG for accessing a knowledge base, function calling for simple actions).
   - Establish initial guardrails for safety and basic control.
2. **Develop Integration Strategy:**
   - Build connectors to necessary backend systems, adapting existing integration points or creating new APIs.
   - Design how the LLM framework will utilize these integrations (e.g., through function calling).
3. **Implement Core Guardrails:**
   - Develop input sanitization (e.g., removing sensitive data, blocking malicious prompts).
   - Implement output filtering (e.g., detecting and blocking harmful, inaccurate, or sensitive content).
   - Configure safety policies and moderation layers.
4. **Build Knowledge Base (if using RAG):**
   - Extract relevant information from existing documentation, FAQs, and potentially dialog tree content.
   - Structure and index the knowledge base for efficient retrieval.

Phase 3: Testing and Validation

1. **Develop Comprehensive Test Cases:**
   - Create test scripts covering common user intents, edge cases, out-of-scope queries, and security test cases (e.g., prompt injection attempts).
   - Include test cases specifically designed to compare LLM behavior against expected outcomes from the original dialog tree (where applicable).
2. **Conduct Automated and Manual Testing:**
   - Use automated testing frameworks to run a large volume of test cases.
   - Perform manual testing with human evaluators to assess conversational quality, empathy, and handling of complex scenarios.
   - Conduct user acceptance testing (UAT) with a representative group of end-users.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 5, September – October 2025 ||

DOI: 10.15680/IJCTECE.2025.0805010

3. **Evaluate Against Success Metrics:**
   - Measure key performance indicators (KPIs) defined in Phase 1 for the pilot DVA (e.g., containment rate, task completion rate, user satisfaction scores from testing).
   - Analyze conversation logs for errors, escalations, and unhandled queries.

4. **Refine and Iterate:**
   - Based on testing results, refine the LLM prompts, orchestration logic, guardrails, and knowledge base.
   - Iterate on the implementation and testing until performance meets defined criteria.

Phase 4: Deployment and Ongoing Governance

1. **Phased Rollout:**
   - Deploy the migrated DVA to a limited user group (e.g., internal users, a small segment of customers).
   - Monitor performance closely and gather feedback.
   - Gradually expand the rollout based on successful monitoring.
2. **Establish Continuous Monitoring:**
   - Implement real-time monitoring of DVA performance, including latency, error rates, escalation rates, and security events.
   - Monitor conversation logs for emerging patterns, unhandled intents, or potential misuse.
3. **Implement Ongoing Governance and Maintenance:**
   - Define processes for updating the LLM model, knowledge base, prompts, and guardrails.
   - Establish a feedback loop from monitoring and user feedback to drive continuous improvement.
   - Regularly review and update security and privacy guardrails in response to new threats or regulations.
   - Plan for retraining or fine-tuning the model as needed based on performance drift.

IV. PRACTICAL IMPLEMENTATION STRATEGIES

Migrating to LLM frameworks involves specific technical strategies to ensure control and reliability.

A. Orchestration Patterns

- **Prompt Engineering:** Crafting clear, specific instructions for the LLM to guide its behavior, define its persona, and constrain its responses. This is the most fundamental control mechanism.
- **Retrieval-Augmented Generation (RAG):** Integrating the LLM with an external, up-to-date knowledge base. The system retrieves relevant information from the knowledge base based on the user query and provides it to the LLM as context for generating a response. This is crucial for factual accuracy and reducing hallucinations.
- **Function Calling (Tool Use):** Enabling the LLM to identify when a user request requires an action (e.g., checking an account balance, placing an order) and generating a structured call to an external API or function to perform that action. This allows the DVA to interact with backend systems and perform tasks beyond generating text.
- **Sequential/Parallel Calling:** Orchestrating multiple LLM calls, or tool uses in a sequence or in parallel to handle complex, multi-step user requests.

B. Guardrail Implementation

- **Input Sanitization:** Using rule-based filters, NER, or smaller, specialized ML models to detect and remove sensitive information or malicious prompts *before* they reach the main LLM.
- **Output Filtering:** Applying similar techniques to the LLM's generated response to detect and redact sensitive data, block harmful content, or correct factual inaccuracies by comparing against trusted sources.
- **Moderation Models:** Using dedicated models (often provided by LLM vendors or third parties) to classify input prompts and generated outputs for safety risks (e.g., hate speech, self-harm, illegal content).
- **Rule-Based Overrides:** Implementing explicit rules that override LLM generation for specific critical scenarios (e.g., always provide the official contact number for emergencies, always use a predefined response for legal disclaimers).
- **Confidence Scoring:** Assessing the confidence of the LLM's response or the NLU component's intent prediction. Low confidence can trigger clarification prompts or escalation to a human.

C. Integration with Enterprise Systems

- **API Connectors:** Building secure and robust APIs to allow the LLM orchestration layer to interact with CRM, databases, knowledge bases, transaction systems, etc.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 5, September – October 2025 ||

DOI: 10.15680/IJCTECE.2025.0805010

- **Middleware/Orchestration Layer:** Developing a layer of software between the LLM and backend systems to manage API calls, data transformation, and state management, ensuring the LLM doesn't directly access sensitive systems without control.

Data Mapping and Transformation: Ensuring data retrieved from backend systems is in a format the LLM can effectively use, and that data sent to backend systems from the DVA is correctly formatted and validated.

V. INDUSTRY-SPECIFIC CONSIDERATIONS

Migrating to LLM-based DVAs has unique implications and requirements across different industries.

A. Healthcare

- **Challenges:** Handling Protected Health Information (PHI) under HIPAA, ensuring clinical accuracy, managing sensitive patient conversations, regulatory compliance, integrating with Electronic Health Records (EHRs).
- **Strategies:**
  - **Strict PHI Redaction:** Implementing advanced NER and rule-based redaction *before* data enters the LLM and on *all* LLM outputs. Using specialized healthcare NLP models.
  - **RAG on Verified Clinical Knowledge:** Using RAG to ground LLM responses in trusted medical databases and internal clinical guidelines, preventing inaccurate medical advice (hallucinations).
  - **Limited LLM Access:** Restricting LLM access to sensitive backend systems (like EHRs) via secure, audited APIs and function calls, with strict access controls.
  - **Escalation for Medical Advice:** Implementing clear guardrails to escalate any requests for medical diagnosis or treatment to qualified healthcare professionals.
  - **Compliance Monitoring:** Continuous auditing of conversations and data access to ensure HIPAA compliance.

B. Finance

- **Challenges:** Handling Nonpublic Personal Information (NPI) and Payment Card Information (PCI) under regulations like GLBA and PCI DSS, ensuring financial accuracy, preventing fraud, managing sensitive transactions.
- **Strategies:**
  - **Tokenization and Redaction:** Implementing PCI-certified tokenization for payment card details and robust redaction for NPI in conversation logs and data passed to the LLM.
  - **Secure Function Calling for Transactions:** Using strictly controlled and audited function calls to interact with banking systems for balance checks, transfers, etc., ensuring the LLM does not directly handle sensitive credentials or transaction details.
  - **RAG on Financial Regulations and Policies:** Grounding responses in internal policies and external financial regulations to ensure compliance and accuracy in providing information.
  - **Fraud Detection Integration:** Integrating DVA interactions into existing fraud monitoring systems.
  - **Clear Disclaimers:** Implementing guardrails to ensure the DVA provides necessary legal and financial disclaimers.

C. Telecommunications

- **Challenges:** Handling customer PI and account details, managing complex billing and service inquiries, integrating with diverse backend systems (billing, network, CRM), high volume of interactions.
- **Strategies:**
  - **PI Redaction and Masking:** Implementing robust redaction and masking of customer names, addresses, phone numbers, and account identifiers in conversation data.
  - **Secure API Integrations:** Using function calling to securely interact with billing systems, service provisioning, and network status APIs, limiting the LLM's direct data access.
  - **RAG on Service Plans and Troubleshooting Guides:** Grounding responses in detailed knowledge bases about service offerings, pricing, and common troubleshooting steps.
  - **Managing Complex Intents:** Utilizing LLMs' advanced NLU to better understand complex, multi-part customer requests related to billing or service issues.
  - **High Availability and Scalability:** Ensuring the LLM framework and its integrations can handle the high volume and peak loads typical in telecom customer service.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 5, September – October 2025 ||

DOI: 10.15680/IJCTECE.2025.0805010

In all these industries, a phased migration starting with lower-risk, well-defined use cases is crucial before tackling more complex or highly sensitive interactions.

VI. CHALLENGES AND SOLUTIONS

Migrating to and operating LLM-based DVAs introduces several challenges:

- **Hallucinations and Factual Accuracy:** LLMs can generate convincing but incorrect information.
  - *Solution:* Implement RAG on verified knowledge bases, use output filtering to validate facts against trusted sources, and use rule-based overrides for critical information.
- **Controllability and Predictability:** LLMs are less deterministic than dialog trees, making it harder to guarantee specific outputs.
  - *Solution:* Extensive prompt engineering, fine-tuning on domain-specific data, using structured output formats (e.g., JSON), and implementing strong guardrails and overrides.
- **Security Risks (Prompt Injection):** Malicious prompts can try to manipulate the LLM or extract sensitive information.
  - *Solution:* Implement robust input sanitization, use dedicated moderation models, apply least privilege principles to LLM access to tools/data, and continuously monitor for suspicious input patterns.
- **Data Privacy and Sensitive Information Handling:** LLMs might inadvertently reveal sensitive data from training or context.
  - *Solution:* Rigorous data redaction/tokenization *before* data reaches the LLM, output filtering, secure logging, and strict access controls.
- **Integration Complexity:** Connecting LLMs to diverse and often legacy enterprise systems.
  - *Solution:* Develop a robust middleware or orchestration layer, use standardized API connectors, and plan for data transformation.
- **Performance and Latency:** LLM inference can be computationally expensive and introduce latency.
  - *Solution:* Optimize model serving infrastructure, use smaller models where appropriate, implement caching, and optimize backend API calls.
- **Evaluation and Monitoring:** Defining metrics and systems to continuously evaluate the performance and safety of a less deterministic system.
  - *Solution:* Focus on outcome-based metrics (task success, user satisfaction), analyze conversation logs for errors and safety failures, and implement continuous monitoring dashboards.
- **Bias in LLMs:** LLMs can inherit biases from their training data.
  - *Solution: Use debiased training data where possible, implement output filtering to detect and mitigate biased language, and conduct fairness audits.*

VII. EVALUATION AND GOVERNANCE

Effective evaluation and robust governance are critical for the success and safety of LLM-based DVAs post-migration.

A. Evaluation Metrics

Beyond traditional DVA metrics like containment and escalation rates, LLM-based agents require metrics that capture their generative nature and user experience:

- **Task Success Rate:** Can the DVA successfully help the user complete their goal? (Still relevant)
- **Containment Rate:** Percentage of conversations resolved by the DVA without human escalation. (Still relevant, but definition might evolve)
- **Accuracy/Factuality Score:** Percentage of DVA statements that are factually correct, verified against trusted sources. Crucial for RAG-based systems.
- **Response Quality Score:** Human evaluation or automated metrics assessing fluency, coherence, relevance, and tone.
- **User Satisfaction Score (CSAT/NPS):** Direct feedback from users.
- **Error Rate (Hallucinations, Irrelevant Responses):** Specific metrics tracking instances where the DVA generates incorrect or unhelpful information.
- **Guardrail Evasion Rate:** Rate at which test cases or user inputs successfully bypass security or safety guardrails (e.g., prompt injection).
- **Latency:** Time taken for the DVA to respond.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 5, September – October 2025 ||

DOI: 10.15680/IJCTECE.2025.0805010

B. Governance Framework

A comprehensive governance framework should include:

- **Policy Definition:** Clear policies on data handling, security, privacy, acceptable use, and brand persona for the DVA.
- **Model Management:** Processes for selecting, evaluating, deploying, and updating LLM models and associated components.
- **Prompt Management:** Version control and review processes for prompt templates and strategies.
- **Knowledge Base Management:** Processes for updating, verifying, and indexing the knowledge base used by RAG.
- **Guardrail Management:** Processes for updating, testing, and monitoring the effectiveness of security, safety, and quality guardrails.
- **Monitoring and Alerting:** Systems for real-time performance, security, and safety monitoring with automated alerts.
- **Human Oversight and Feedback Loops:** Mechanisms for human review of flagged conversations, user feedback analysis, and incorporating insights into DVA improvement.
- **Compliance and Audit Trails:** Logging all interactions, data access, and system events to demonstrate compliance with regulations.

VIII. CONCLUSION

Migrating from traditional dialog tree architectures to frameworks leveraging Large Language Models represents a significant opportunity to build more capable, flexible, and engaging Digital Virtual Agents. However, this transition is complex, requiring careful planning, technical expertise, and a strong focus on governance, particularly in regulated industries.

This paper has provided a practical guide to this migration, outlining a structured methodology from assessment and planning through phased implementation, testing, and ongoing governance. We have detailed practical strategies for orchestrating LLM behavior, implementing robust guardrails for security, privacy, and quality, and integrating with existing enterprise systems. Industry-specific considerations for healthcare, finance, and telecommunications highlight the critical need to address unique compliance and data handling requirements.

While challenges related to controllability, accuracy, and security persist, they can be effectively mitigated through the diligent application of techniques such as RAG, function calling, rigorous input/output filtering, and continuous monitoring. By adopting the structured approach and practical strategies outlined in this paper, organizations can navigate the complexities of this migration, unlock the potential of generative AI, and build scalable, reliable, and highly effective virtual agents that enhance user experience and drive business value while maintaining trust and compliance.

REFERENCES

1. Rawal, A., McCoy, J., Rawat, D.B., Sadler, B.M. and Amant, R.S., 2021. Recent advances in trustworthy explainable artificial intelligence: Status, challenges, and perspectives. IEEE Transactions on Artificial Intelligence, 3(6), pp.852-866.
2. Pradhan, Dr. Rashmiranjan. “Establishing Comprehensive Guardrails for Digital Virtual Agents: A Holistic Framework for Contextual Understanding, Response Quality, Adaptability, and Secure Engagement.” International Journal of Innovative Research in Computer and Communication Engineering, 2025. doi:10.15680/IJIRCCE.2025.1307013.
3. Pradhan, D. R. RAGEvalX: An Extended Framework for Measuring Core Accuracy, Context Integrity, Robustness, and Practical Statistics in RAG Pipelines. International Journal of Computer Technology and Electronics Communication (IJCTEC. https://doi.org/10.15680/IJCTECE.2025.0805001
4. Pradhan, D. R. (2025). RAG vs. Fine-Tuning vs. Prompt Engineering: A Comparative Analysis for Optimizing AI Models. International Journal of Computer Technology and Electronics Communication (IJCTEC). https://doi.org/10.15680/IJCTECE.2025.0805004
5. Pradhan, Rashmiranjan, and Geeta Tomar. "AN ANALYSIS OF SMART HEALTHCARE MANAGEMENT USING ARTIFICIAL INTELLIGENCE AND INTERNET OF THINGS.". Volume 54, Issue 5, 2022 (ISSN:

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 5, September – October 2025 ||

DOI: 10.15680/IJCTECE.2025.0805010

0367-6234). Article history: Received 19 November 2022, Revised 08 December 2022, Accepted 22 December 2022. Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology.

6. Pradhan, Rashmiranjan. "AI Guardian- Security, Observability & Risk in Multi-Agent Systems." International Journal of Innovative Research in Computer and Communication Engineering, 2025. doi:10.15680/IJIRCCE.2025.1305043.
7. Pradhan, D. R. (no date) "RAGEvalX: An Extended Framework for Measuring Core Accuracy, Context Integrity, Robustness, and Practical Statistics in RAG Pipelines," International Journal of Computer Technology and Electronics Communication (IJCTEC. doi: 10.15680/IJCTECE.2025.0805001.
8. Rashmiranjan, Pradhan Dr. "Empirical analysis of agentic ai design patterns in real-world applications." (2025).
9. Pradhan, Rashmiranjan, and Geeta Tomar. "IOT BASED HEALTHCARE MODEL USING ARTIFICIAL INTELLIGENT ALGORITHM FOR PATIENT CARE." NeuroQuantology 20.11 (2022): 8699-8709.
10. Rashmiranjan, Pradhan. "Contextual Transparency: A Framework for Reporting AI, Genai, and Agentic System Deployments across Industries." (2025).
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Neural Information Processing Systems (NIPS).
12. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.
13. Lewis, P., Yih, W. T., Pihur, V., Lewis, K., Simig, D., Koren, N., ... & Riedel, S. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems (NeurIPS).
14. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems (NeurIPS).
15. Mosbah, S., & Driss, M. (2024). Comparative Analysis of RAG Fine-Tuning and Prompt Engineering in Chatbot Development. Available via platforms that index research, potentially including IEEE.
16. Gao, Y., Ma, X., Zhou, J., Yan, Y., Zhang, J., Liu, S., ... & Li, H. (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv preprint arXiv:2312.10997.
17. Lester, B., Al-Rfou, R., & Constant, N. (2021). The Power of Scale for Parameter-Efficient Fine-Tuning. arXiv preprint arXiv:2103.10385.
18. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685.
19. Kirchhoff, K., & Kordoni, A. (Eds.). (2017). Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE.