iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78864

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 III Mar 2026- Available at www.ijraset.com

Large Language Models in Production: Evaluation Frameworks, Safety Engineering, and Multi-Objective Deployment Strategies

Chandra Chinthapanti
Independent Researcher

Abstract: ***The deployment of large language models (LLMs) in production systems has exposed a fundamental gap between the rapid advancement of model capabilities and the maturity of the engineering practices used to govern them. Unlike conventional software, LLMs are probabilistic, non-deterministic, and increasingly embedded in safety-critical, multi-objective, and multilingual environments where standard quality assurance techniques are insufficient.***

This paper surveys the key engineering challenges in production LLM deployment—covering evaluation methodology, lifecycle discipline, agent-level robustness, and linguistic coverage—and proposes a unified four-stage pipeline that integrates acceptance test-driven development, Pareto-based multi-objective evaluation, and rigorous language-specific benchmarking. We examine how these approaches address complementary failure modes: misalignment between model behaviour and business requirements, collapse of multi-dimensional performance to inadequate scalar metrics, and silent degradation in non-English deployment settings.

The proposed pipeline draws on engineering principles that are well-established in software development but largely absent from current LLM practice, and we demonstrate how their adoption yields more comprehensive safety coverage, more transparent trade-off analysis, and more reliable multilingual quality assurance than benchmark-centric approaches alone.

Keywords: large language models, LLM safety, test-driven development, acceptance testing, multi-objective optimisation, CPMAI, Pareto dominance, LLM evaluation, NLP tooling, trustworthy AI.

I. INTRODUCTION

Large language models have become foundational infrastructure in modern AI deployments. From customer-facing conversational assistants and internal knowledge retrieval systems to automated code review pipelines and regulatory compliance tooling, LLMs now occupy roles that demand reliability, auditability, and alignment with business-defined objectives. The pace of adoption, however, has substantially outrun the maturity of the engineering discipline that governs these systems.

Conventional software quality assurance—unit testing, integration testing, regression suites—does not transfer cleanly to probabilistic, generative models whose outputs are neither deterministic nor enumerable. Academic benchmarks such as GLUE, SuperGLUE, and BIG-Bench capture capability in isolation but do not integrate into iterative development cycles or reflect application-specific acceptance criteria [4]. Meanwhile, the multi-objective nature of real deployments—where a model must simultaneously maximise correctness, minimise latency, satisfy safety constraints, and respect resource budgets—is rarely addressed holistically by existing evaluation frameworks.

This paper argues that three recently published contributions offer a principled path forward. ATDLLMD [5] adapts acceptance test-driven development to the LLM lifecycle, embedding structured, business-aligned evaluation directly into model iteration. MO-PTSP [11] provides a rigorous, conflict-aware benchmarking environment in which AI agents must resolve genuine objective trade-offs in real time. CalamanCy [16] demonstrates how the same evaluation rigour must extend to linguistically diverse settings, using a carefully constructed Tagalog NER dataset as a case study. Together, these contributions cover the critical axes of trustworthy LLM deployment: lifecycle discipline, multi-objective robustness, and multilingual coverage.

The remainder of this paper is structured as follows. Section II surveys related work and situates each of the three contributions within it. Section III analyses the shared engineering principles across the frameworks. Section IV proposes a unified deployment pipeline. Section V discusses implications and limitations. Section VI concludes.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

II. RELATED WORK AND FRAMEWORK CONTEXT

A. LLM Evaluation and the Limits of Static Benchmarks

Standard LLM evaluation has evolved from perplexity-based intrinsic measures toward comprehensive task-specific benchmarks. BIG-Bench [2] and HELM [3] aggregate hundreds of tasks to assess model capability across a broad distribution of challenges. Instruction-following benchmarks such as MT-Bench [4] and AlpacaEval [18] focus on the alignment of model outputs with human preferences. Despite their breadth, these benchmarks share a fundamental limitation: they are static artefacts designed to measure model capability at a snapshot rather than system behaviour in context. They do not integrate into iterative development cycles, do not capture business-specific acceptance criteria, and do not model the multi-objective nature of production deployments.

Recent work has begun addressing some of these gaps. Constitutional AI [6] formalises alignment principles as testable propositions; RLHF-based methods [7] use human preference labels as implicit acceptance tests. LM-Eval [8] provides a modular harness for reproducible evaluation across tasks. However, none provides a structured engineering methodology that closes the loop between business stakeholders, development teams, and model evaluation artefacts. ATDLLMD [5] directly addresses this gap by adapting agile acceptance test-driven development to the LLM development lifecycle. Its core contribution is the integration of the Cognitive Process Management for AI (CPMAI) lifecycle with a bespoke evaluation harness—LM-Eval—that executes business-aligned acceptance test suites reproducibly across model versions. An iterative, user-centric feedback loop ensures that both the model and its test suite evolve as stakeholder requirements change, making it particularly suited to domains with shifting regulatory constraints or adversarial input distributions.

B. Test-Driven and Behaviour-Driven Development for Machine Learning

Test-Driven Development (TDD) and Behaviour-Driven Development (BDD) have been proposed as disciplinary frameworks for machine learning systems. ML-TDD [9] adapts the red-green-refactor cycle by defining data-dependent test cases prior to model training. Checklist [10] introduces a behavioural testing methodology for NLP models structured around capability types. These approaches establish the intellectual precedent for test-first LLM development but stop short of a full lifecycle methodology that encompasses deployment, monitoring, and business-aligned acceptance. ATDLLMD [5] builds directly on this lineage while closing those gaps through the CPMAI integration and its live feedback loop, distinguishing it from prior work that treats testing as a one-time pre-deployment activity rather than a continuous engineering practice.

C. Multi-Objective Optimisation and Agent Benchmarking

Multi-objective optimisation in AI agents has been studied extensively in the context of evolutionary algorithms, reinforcement learning, and planning. NSGA-II [12] and MOEA/D [13] provide foundational algorithms for Pareto-optimal solution search. In game-based AI evaluation, MCTS with multi-objective extensions yields competitive policies under objective conflict [14]. Standardised benchmarking environments for multi-objective agent evaluation, however, remain sparse—particularly for physics-based settings in which resource budgets and timing limits interact dynamically.

MO-PTSP [11] fills this gap by extending the Physical Travelling Salesman Problem with three simultaneous, genuinely conflicting objectives: minimising task completion time, fuel consumption, and structural damage. Unlike benchmark environments in which objectives are artificially separable, MO-PTSP imposes realistic trade-offs—minimising time increases fuel cost; avoiding structural damage requires detours that lengthen both. Agents are evaluated using Pareto dominance, and the study demonstrates that incorporating macro-actions within MCTS substantially improves agent performance across the Pareto frontier. The environment serves as a robust, near-realistic testbed for any AI system that must balance competing operational constraints—directly applicable to LLM-driven agents deployed under latency, cost, and safety budgets.

D. Multilingual and Low-Resource NLP Evaluation

The predominance of English in LLM pre-training corpora introduces systematic evaluation gaps for non-English and low-resource languages. Multilingual models such as mBERT [15] and XLM-R [17] achieve strong cross-lingual transfer for high-resource languages but degrade significantly on morphologically complex or data-scarce languages. The construction of language-specific, task-specific benchmarks with careful annotation methodology is a necessary precondition for meaningful LLM evaluation in these settings.

CalamanCy [16] exemplifies this discipline for Tagalog. Its tlunified dataset is manually annotated for three NER entity types—Person, Organisation, and Location—by native speakers under iteratively refined guidelines, with inter-annotator agreement (Cohen's kappa > 0.80) measured as a quality gate.

Benchmarks span the full model spectrum: spaCy baseline, fastText embeddings, mBERT (F1=78.0), XLM-R (F1=80.7), a monolingual Tagalog RoBERTa achieving top performance, and zero-shot LLM results that, while promising, remain below supervised baselines. This breadth enables principled cost-performance trade-off analysis and foregrounds the gap between English-centric benchmark performance and real-world deployment quality—a gap that any enterprise LLM deployment spanning multiple geographies must actively manage.

III. SHARED ENGINEERING PRINCIPLES

Despite operating at different levels of the LLM deployment stack—lifecycle methodology, agent benchmarking, and dataset construction—the three frameworks converge on a set of engineering principles that distinguish rigorous LLM deployment from ad hoc practice.

A. Specification Before Evaluation

ATDLLMD [5] requires that acceptance tests be specified before model development begins, ensuring that evaluation criteria are not retrofitted to match model capability. CalamanCy [16] applies an analogous discipline at the dataset level: annotation guidelines are iteratively refined and agreement is measured before labelling proceeds at scale. MO-PTSP [11] formalises the objective conflict structure of its environment before agent training, so that Pareto-frontier analysis reflects genuine trade-offs rather than post-hoc rationalisation. This shared principle of specification-before-evaluation is a hallmark of mature software engineering and is largely absent from mainstream LLM evaluation practice, where benchmarks are typically selected after model development to report the most favourable results.

B. Multi-Dimensional Performance Characterisation

All three frameworks resist the reduction of system quality to a single scalar metric. ATDLLMD maintains separate pass/fail records for each acceptance test category—safety, correctness, business alignment, regulatory compliance—enabling targeted iteration on specific failure modes. MO-PTSP reports full Pareto frontiers rather than a single aggregate score, preserving the trade-off information that a weighted sum would discard. CalamanCy reports per-entity-type F1 alongside aggregate macro-F1 and disaggregates results across model families, supporting cost-performance analysis that a single headline number cannot provide. Together, these approaches demonstrate that multi-dimensional reporting is not merely good practice but a functional requirement for deployments in which different stakeholders have different priorities.

C. Iterative Refinement as a Continuous Engineering Practice

Each framework is designed for iteration rather than one-time evaluation. ATDLLMD's user-centric feedback loop continuously refines both the model and its test suite, treating evaluation as a living artefact rather than a fixed checklist. MO-PTSP's parameterisable objective weights and physical constraints allow the environment to be updated as deployment conditions change. CalamanCy's annotation framework is extensible to new entity types and text domains. This orientation toward continuous evaluation mirrors the CI/CD discipline standard in software engineering and is essential for LLM systems operating in environments where adversarial inputs, regulatory requirements, and user behaviour evolve over time.

Table I Comparative Framework Properties

Property	ATDLLMD [5]	MO-PTSP [11]	CalamanCy [16]
Evaluation scope	System lifecycle	Agent behaviour	Model capability
Specification timing	Before development	Before agent training	Before annotation
Metric dimensionality	Per-category pass/fail	Pareto frontier	Per-type F1
Iteration model	Continuous (CI/CD)	Parameterisable env.	Extensible corpus
Primary stakeholder	Business owners	AI researchers	Native annotators

IV. A UNIFIED PIPELINE FOR TRUSTWORTHY LLM DEPLOYMENT

The shared principles identified above support the integration of all three frameworks into a four-stage deployment pipeline that addresses the full arc from initial specification to continuous post-deployment monitoring.

A. Stage 1 — Specification

Before any model training or fine-tuning begins, stakeholders collaboratively define acceptance test suites following the ATDLLMD methodology [5]. Tests are organised into four categories: (i) correctness tests verifying factual accuracy and task-specific quality; (ii) safety tests probing for harmful outputs, prompt injection vulnerabilities, and policy violations; (iii) business alignment tests verifying compliance with product requirements and regulatory constraints; and (iv) efficiency tests enforcing latency and cost budgets. For multilingual deployments, language-specific test suites are constructed following the annotation rigour principles demonstrated by CalamanCy [16], with inter-annotator agreement measurement as a mandatory quality gate before any labelling proceeds at scale.

B. Stage 2 — Development and Iteration

Model development proceeds against the acceptance test suite in a red-green-refactor cycle, with CPMAI lifecycle gates ensuring that each development phase is evaluated against the full suite before progressing. A failing test identifies a specific deficiency—a category of unsafe output, an unmet business requirement—and guides the next iteration of fine-tuning, RLHF, or prompt engineering. The iterative feedback loop from ATDLLMD [5] ensures the test suite itself evolves alongside the model as stakeholder understanding deepens, preventing the common failure mode in which acceptance criteria are frozen while deployment requirements continue to change.

C. Stage 3 — Multi-Objective Agent Evaluation

For deployments in which the LLM drives or augments an agent operating in a dynamic environment, the MO-PTSP benchmarking methodology [11] is applied to characterise agent behaviour under objective conflict. The agent is evaluated in a parameterised simulation whose objective weights are calibrated to the production deployment context. Pareto-dominance analysis identifies non-dominated strategies across all objectives, and the deployment policy is selected from this set based on stakeholder priority weights. MCTS with macro-actions, demonstrated to be effective in this setting [11], provides a practical starting point for policy search in environments where latency, cost, and safety constraints are simultaneously active.

D. Stage 4 — Continuous Monitoring and Re-evaluation

Post-deployment, the acceptance test suite is executed on a scheduled basis against live model outputs sampled from production traffic. Failures trigger automatic alerts and initiate a new development iteration. A multi-objective monitoring dashboard tracks the Pareto frontier of agent performance over time, flagging objective drift as the model or environment evolves. Language-specific evaluation panels, constructed following the CalamanCy methodology [16], provide disaggregated quality metrics for each deployment language, preventing the silent degradation of non-English performance that is common when model updates are validated only on English-language test sets.

Table II Pipeline Stage Summary

Stage	Activity	Framework	Key Output
1 — Specification	Define acceptance test suites & annotation guidelines	ATDLLMD [5], CalamanCy [16]	Validated test suite with measured inter-annotator agreement
2 — Development	Red-green-refactor iterations against acceptance suite	ATDLLMD [5]	Model passing full acceptance suite at each lifecycle gate
3 — Agent Evaluation	Pareto-dominance analysis under objective conflict	MO-PTSP [11]	Pareto-optimal deployment strategy calibrated to production
4 — Monitoring	Continuous test execution, drift detection, re-evaluation	ATDLLMD [5], MO-PTSP [11]	Automated alerts and per-language disaggregated metrics

V. DISCUSSION

A. Practical Implications

The unified pipeline has several practical implications for LLM practitioners. First, anchoring evaluation to acceptance tests defined by business stakeholders shifts the centre of gravity of LLM development from capability maximisation to requirements satisfaction—a shift that reduces the common failure mode in which benchmark improvements do not translate to deployment value [5]. Second, the Pareto-dominance framing provides a principled response to the enterprise challenge of optimising LLM deployments across conflicting KPIs—cost, latency, accuracy, and safety—without collapsing them to a weighted sum that obscures individual objective performance [11]. Third, the language-disaggregated monitoring panels address the silent degradation risk for non-English deployment languages, a risk that the CalamanCy benchmark [16] empirically documents through the gap between multilingual model performance on high-resource versus low-resource languages.

From an organisational standpoint, all three frameworks are designed to be adopted incrementally. An engineering team can introduce acceptance test-driven iteration for a single high-risk model capability, apply MO-PTSP evaluation to a single agent pipeline, or construct a CalamanCy-style benchmark for a single deployment language—without committing to the full pipeline from the outset. This modularity lowers the barrier to adoption in resource-constrained settings while preserving the option for full integration as organisational maturity grows.

B. Limitations

Several limitations warrant acknowledgement. The acceptance test-driven approach requires sustained stakeholder engagement to maintain and evolve test suites, which can be resource-intensive in organisations with fluid requirements. The MO-PTSP environment represents a stylised setting whose objective structure may not map directly to all production deployment contexts; calibrating its objective weights requires careful domain modelling.

Language-specific evaluation panels of the quality demonstrated by CalamanCy [16] require access to native-speaking annotators and iterative guideline development, which may be challenging for languages with small annotator pools. Finally, the pipeline as proposed assumes a relatively mature ML infrastructure; teams without existing CI/CD integration for model evaluation will require non-trivial tooling investment.

C. Future Directions

Several promising directions remain open. Automated adversarial test generation for ATDLLMD—using LLMs to synthesise acceptance test cases targeting known failure modes—could substantially reduce the manual burden of test suite maintenance. Extending MO-PTSP to environments with stochastic objective functions would better model the uncertainty inherent in production deployments. Scaling the CalamanCy annotation methodology to additional low-resource languages and NLP tasks beyond NER—relation extraction, coreference resolution, machine translation quality estimation—would broaden multilingual LLM evaluation coverage. Integrating all three frameworks into a unified evaluation platform with automated reporting and dashboard-level visibility could lower the adoption barrier for enterprise teams.

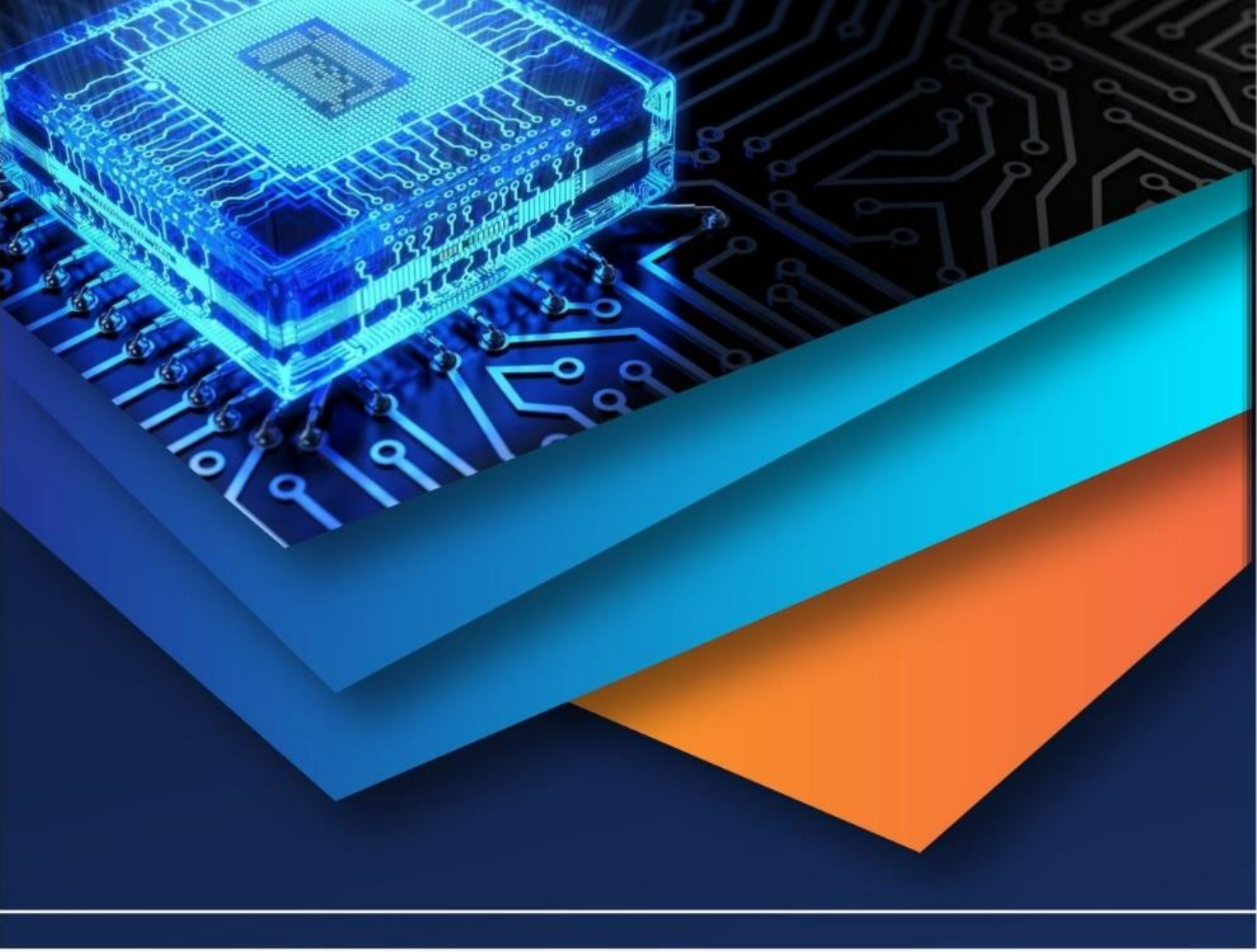
VI. CONCLUSION

This paper has examined three complementary contributions to the challenge of trustworthy LLM deployment. ATDLLMD [5] provides a lifecycle methodology that embeds business-aligned acceptance criteria into LLM development through the CPMAI framework and an iterative feedback loop. MO-PTSP [11] supplies a rigorous, physics-based benchmarking environment in which agents must navigate genuine objective conflicts, enabling Pareto-frontier analysis of deployment strategies. CalamanCy [16] demonstrates the annotation discipline and disaggregated reporting required to extend evaluation rigour to linguistically diverse deployment settings.

These frameworks share three foundational engineering principles—specification before evaluation, multi-dimensional performance characterisation, and continuous iterative refinement—that together define a more mature practice of LLM engineering than current benchmark-centric approaches afford. The four-stage unified pipeline synthesised from these contributions offers a practical roadmap for teams seeking to move from capability benchmarking to requirements-driven, Pareto-aware, and multilingual LLM deployment. As LLMs are deployed in increasingly high-stakes and diverse operational settings, such a roadmap represents an engineering necessity rather than a best-effort aspiration.

REFERENCES

[1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020.
[2] B. Srivastava et al., "Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models," arXiv:2206.04615, 2022.
[3] P. Liang et al., "HELM: Holistic Evaluation of Language Models," in Proc. NeurIPS Datasets and Benchmarks Track, 2022.
[4] L. Zheng et al., "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," in Proc. NeurIPS, 2023.
[5] V. R. Parupally, "ATDLLMD: A Test-Driven Framework for Safe, Reliable, and Business-Centric LLM Development," IET Conf. Proc., vol. 2025, no. 43, pp. 612–618, 2025, doi: 10.1049/icp.2025.4778.
[6] Y. Bai et al., "Constitutional AI: Harmlessness from AI Feedback," arXiv:2212.08073, 2022.
[7] P. F. Christiano et al., "Deep Reinforcement Learning from Human Preferences," in Proc. NeurIPS, 2017.
[8] L. Gao et al., "A Framework for Few-Shot Language Model Evaluation," Zenodo, 2021, doi: 10.5281/zenodo.5371628.
[9] K. Beck, Test-Driven Development: By Example. Addison-Wesley, 2003.
[10] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList," in Proc. ACL, 2020.
[11] V. R. Parupally, "A Multi-Objective Game Environment for Evaluating AI Agents," in Proc. 2nd Global AI Summit – Int. Conf. on Artificial Intelligence and Emerging Technology (AI Summit), Noida, India, 2025, pp. 692–697, doi: 10.1109/AISummit66170.2025.11410745.
[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, 2002.
[13] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," IEEE Trans. Evol. Comput., vol. 11, no. 6, pp. 712–731, 2007.
[14] T. Brys, A. Harutyunyan, P. Vrancx, A. Nowé, and M. Taylor, "Multi-Objectivization of Reinforcement Learning Problems by Reward Shaping," in Proc. IJCNN, 2014.
[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL, 2019.
[16] V. Parupally, "CalamanCy: A Tagalog Natural Language Processing Toolkit," in Proc. IEEE Int. Conf. on Industrial Technology & Computer Engineering (ICITCE), Penang, Malaysia, 2025, pp. 45–51, doi: 10.1109/ICITCE65255.2025.11210765.
[17] A. Conneau et al., "Unsupervised Cross-lingual Representation Learning at Scale," in Proc. ACL, 2020.
[18] Y. Li et al., "AlpacaEval: An Automatic Evaluator of Instruction-Following Models," GitHub, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY