A Systematic Literature Review of Testing Tools for AI and Machine Learning Software

Goian Sergiu

Babes Bolyai University Cluj-Napoca, , Arad, , Romania

ARTICLE INFO

Keywords:
Testing, Validation, AI, Machine Learning, LLMs, Large Language Model, Evaluation, Robustness, Fairness, Explainability, Test Automation, Software Testing, Model Testing, Frameworks, Testing Tools, Systematic Literature Review, Quality Assurance, AI Models, ML Models, Evaluation Metrics, Performance Testing, SLR

ABSTRACT

1em This study presents a systematic literature review (SLR) of existing tools and frameworks for testing Artificial Intelligence (AI) and Machine Learning (ML) software systems. With the increasing adoption of AI and ML technologies, there is a critical need for reliable testing methods to ensure system robustness, fairness, and explainability. The primary aim of this review is to identify the tools currently used for testing AI and ML systems, explore the specific testing aspects they address, and evaluate their strengths and limitations. Through an analysis of 10 key research papers, this study identifies AI-powered test automation tools, scenario-based testing approaches, and real-world testing frameworks as prominent solutions. Additionally, the review highlights the tools' focus on various aspects, such as robustness, fairness, and explainability, and discusses the strengths of these tools, including increased efficiency and comprehensive testing. However, limitations such as the complexity of models, adoption barriers, and limited coverage of certain testing aspects are also identified. The results provide valuable insights for researchers and practitioners seeking to enhance the testing and validation of AI and ML systems.

1. Systematic literature review (SLR)

This review follows established guidelines for conducting systematic literature reviews in software engineering, as proposed in Kitchenham (2004); Kitchenham and Charters (2007). Additional methodological insights were drawn from Campeanu (2018) and Davila and Nunes (2021).

2. Study design

The structure of the study follows the approach of a Systematic Literature Review (SLR) and it details the need for the review, research questions, and the protocol used to ensure transparency and repeatability.

2.1. Review need identification

With the growing adoption of AI and ML systems in various domains, it is need to ensure their reliability through effective testing. However, the testing of such systems leads to challenges. While various tools and techniques have been proposed, there is a lack of consolidated knowledge about their coverage, capabilities, and limitations. This review aims to fill that gap.

2.2. Research questions definition

The study proposes to respond to the following research questions that guide this review: -What tools and frameworks currently exist for testing ML and AI software systems? -What testing aspects (robustness, fairness, explainability) do these tools target? -What are the strengths and limitations of the identified tools?

3. Conducting the SLR

The systematic review was conducted in several phases, following a transparent and repeatable process. This involved comprehensive searches in academic databases, refining the selection of studies, extracting relevant data and synthetizing findings. Each phase followed the format of a pre-established protocol to ensure consistency and reproducibility, ultimately ensuring that the conclusions drawn were based on sound evidence.

3.1. Search and selection process

The search and selection process followed a structured strategy to identify studies on AI and ML testing (including more recent features as LLMs). The primary focus was on articles related to software testing tools and methodologies, ensuring that the studies selected would provide comprehensive insights into current practices in testing complex AI/ML systems. The search was conducted in arXiv's database using a mix of keywords and each retrieved study was evaluated for alignment with the research questions. Rigorous filtering was applied at each step to ensure that only relevant, high-quality studies were included in the review.

3.1.1. Database search

The search process was carried out using arXiv's advanced search capabilities to ensure that only relevant articles were retrieved. A comprehensive set of search terms was used, including "AI software testing," "ML validation", "automated testing for machine learning systems" and "testing LLMs". The search was limited to publications from the past 6 years to ensure that only the most current research was included. The results were further filtered based on specific categories within arXiv, such as software engineering, artificial intelligence and ML to target the most relevant studies.

✉ sergiu_goian@yahoo.com (G. Sergiu)
ORCID(s):

Id	Citation	Title	Year
P1	Dusica Marijan (2021)	Software Testing for Machine Learning	2021
P2	Xuan Xie (2024)	LeCov: Multi-level Testing Criteria for Large Language Models	2024
P3	A. Ramadan (Üsküdar University(2020)	The Role of Artificial Intelligence and Machine Learning in Software Testing	2020
P4	J. Chandrasekaran (2022)	Test and Evaluation Best Practices for Machine Learning-Enabled Software Systems	2022
P5	M. Openja (2021)	An Empirical Study of Testing Machine Learning in the Wild	2021
P6	Al Khan (2020)	Applying Machine Learning Analysis for Software Quality Test	2020
P7	Katja Karhu (2021)	A Secondary Study on AI Adoption in Software Testing	2021
P8	Mohammad Baqar (2022)	The Future of Software Testing: AI-Powered Test Case Generation and Validation	2022
P9	Rachel Brower-Sinning (2022)	Using Quality Attribute Scenarios for ML Model Test Case Generation, Rachel Brower-Sinning, Grace A. Lewis, Sebastían Echeverría	2022
P10	Junming Cao (2021)	Understanding the Complexity and Its Impact on Testing in ML-Enabled Systems	2021

Table 1
The list of selected papers

3.1.2. Merging, and duplicates and impurity removal

After the initial search, a thorough duplicate removal process was undertaken to eliminate any less relevant papers. This step involved revalidation of paper titles and abstracts. Additionally, any papers that were not directly related to AI/ML testing, such as those focused purely on theoretical algorithms without practical testing applications, were also excluded. This resulted in a refined set of papers that were relevant to the review's scope and ensured that the data used for analysis would be of high quality.

3.2. Data extraction

The selected studies were assessed based on clear inclusion and exclusion criteria to ensure that only the most relevant research was included in the review, since the selection criteria applied to the retrieved papers were based on relevance and alignment with the research questions. Papers were included if they explicitly discussed AI or ML testing methodologies, tools or case studies, on the other hand they were excluded if they focused only on theoretical concepts without practical testing applications.

The systematic search resulted in a total of 10 relevant papers, carefully selected for inclusion in this review. These papers were chosen based on their focus on AI and machine learning testing methodologies, tools and case studies. The selected studies cover various aspects of software testing related to AI, from evaluating machine learning models and LLMs to applying AI-driven test automation tools. Below, the paper presents the selected papers and their corresponding details in Table 1.

3.3. Data synthesis

Data synthesis involves collating and summarising the results of the included primary studies. Due to the diversity of evaluation metrics and objectives, a descriptive (narrative) synthesis approach was adopted. The 10 selected papers were analysed and grouped into categories based on their primary focus. The following themes emerged:

- **Testing methodologies and frameworks (Testing Automation)** (P1, P3, P5, P7, P8)
- **Coverage criteria and robustness evaluation** (P2, P4, P9)
- **Testing challenges and complexity in ML systems** (P6, P10)

Each study was reviewed for details regarding its intervention (type of testing approach or framework), context (the type of ML/AI system tested), outcome (improved reliability, detection of edge cases or validation methodology) and quality indicators (clarity, scope).

Table 2 below summarizes key characteristics across the selected papers to highlight similarities and differences.

From this synthesis, it is obvious that, while some results are consistent, especially regarding the necessity of tailored testing frameworks for AI/ML, there is also considerable diversity in the specific tools and approaches. Future reviews could focus on quantitative synthesis if more comparable evaluation metrics and test benchmarks are made standard across studies. Here, for each of the 10 selected articles, a short summary of 2-3 paragraphs is made. This section presents concise summaries of the ten primary studies selected for this systematic literature review. Each summary outlines the objective, methodology, tools or datasets used, and key findings of the respective paper. These summaries provide insight into the current landscape of testing techniques and tools used in AI and machine learning software development. **1. Software Testing for Machine Learning**

Paper ID	Main Focus	System Type / Context	Outcome / Contribution
P1	General framework for ML software testing	ML pipelines in production	Proposed structured test lifecycle
P2	Testing criteria for LLMs	Transformer-based LLMs	Introduced multi-level test adequacy metrics
P3	Survey on AI/ML in software testing	General AI/ML tools	Overview of AI applications in test generation
P4	Empirical study of ML system behavior under faults	Real-world ML systems	Identified weaknesses in runtime robustness
P5	ML-based test quality analysis	Traditional software testing enhanced with ML	Showed improvement in fault detection using ML
P6	Complexity in ML testing	ML-enabled systems	Discussed how complexity affects testability
P7	Secondary study on AI adoption	Industrial AI testing tools	Trends and challenges identified via secondary SLR
P8	AI for test case generation	ML test suite automation	Presented AI-powered tool for test generation
P9	Use of quality attribute scenarios	ML model testing	Scenario-based method for deriving test cases
P10	Complexity impact analysis	ML system validation	Highlighted difficulty in achieving test completeness

Table 2
Summary of primary studies: focus, context, and contribution

This study proposes a structured testing framework tailored to the unique properties of ML-based software systems. The authors identify key challenges such as nondeterministic behavior, evolving models, and dependency on training data. Their proposed testing methodology includes data testing, model testing, and system integration testing. The paper outlines how traditional software testing techniques must be adapted to suit AI components and provides case study examples to illustrate their method. **2. LeCov: Multi-level Testing Criteria for Large Language Model** The paper "LeCov: Multi-level Testing Criteria for Large Language Models" introduces LeCov, which is a framework designed to enhance the testing and evaluation of Large Language Models (LLMs). Recognizing the challenges posed by the limited interpretability of LLMs, especially concerning issues like truthfulness and toxicity, the authors identify a gap in systematic and formalized testing criteria. To address this, LeCov proposes a set of nine testing criterias that delve into three critical internal components of LLMs: the attention mechanism, feed-forward neurons, and uncertainty. This multi-level approach aims to provide a more granular and thorough assessment of LLM behavior, moving beyond surface-level evaluations. The effectiveness of LeCov is demonstrated through experimental evaluations conducted on three different LLMs across four datasets, showcasing its utility in uncovering defects and enhancing the reliability of LLMs before deployment. By providing a structured and in-depth testing methodology, LeCov contributes to the development of more trustworthy and robust language models, which is crucial these days due to the recent advancements in this direction. **3. The Role of Artificial Intelligence and Machine Learning in Software Testing** This paper investigates how AI and ML can enhance software testing by automating tasks such as test case generation and fault detection. The authors demonstrate that integrating all these technologies together can significantly improve testing efficiency and reliability. **4. Test and Evaluation Best Practices for Machine Learning- Enabled Software Systems** The analysis shown by the authors in the above paper focuses on developing best practices for testing and evaluating machine learning-enabled software systems. The purpose is to address the unique challenges posed by ML models, such as their unpredictability and complexity. The authors propose a structured approach for integrating testing throughout the lifecycle of ML systems, ensuring robustness and reliability. The results emphasize the importance of continuous validation and tailored test strategies to handle dynamic model behavior and evolving datasets.

5. An Empirical Study of Testing Machine Learning in the Wild In this paper, authors highlight the challenges of testing machine learning models in real-world applications.

The purpose is to understand how ML systems perform outside controlled environments, where datasets and conditions are more unpredictable. The study highlights key testing challenges and proposes strategies to improve the robustness of ML models in such settings. The results reveal significant gaps in current testing practices, emphasizing the need for more adaptive and comprehensive testing approaches for ML in the wild. **6. Applying Machine Learning Analysis for Software Quality Test** This paper explores how machine learning techniques can be applied to software quality testing to enhance defect detection and improve overall software performance. The main purpose is to integrate ML models into testing workflows to automate and optimize the identification of potential issues. The results show that applying ML-based analysis significantly increases the accuracy and efficiency of software testing, leading to faster detection of defects and improved software reliability. **7. A Secondary Study on AI Adoption in Software Testing** The above study focuses on the growing trend of adopting Artificial Intelligence (AI) in software testing, specifically examining how AI is being integrated into testing practices. The purpose is to understand the current state of AI usage in the field and identify barriers to its wider adoption. The results reveal that while AI adoption offers substantial benefits in terms of automation and efficiency, challenges such as lack of expertise and integration difficulties remain significant. **8. The Future of Software Testing: AI-Powered Test Case Generation and Validation** The article discusses the potential of AI-powered test case generation and validation as the future of software testing. Its primary goal is to explore how AI can automate the creation and validation of test cases, ultimately reducing manual effort and increasing testing coverage. The results suggest that AI can significantly improve testing efficiency by generating more effective and diverse test cases while reducing human errors and effort. **9. Using Quality Attribute Scenarios for ML Model Test Case Generation** This work presents a method for generating test cases for machine learning (ML) models by using quality attribute scenarios. The purpose is to integrate specific quality attributes like performance, security, and usability into the test case generation process to ensure a more comprehensive evaluation of ML models. The results indicate that incorporating quality attribute scenarios into test case generation leads to better alignment of test coverage with real-world application requirements. **10. Understanding the Complexity and Its Impact on Testing in ML-Enabled Systems** This article explores the complexity of machine learning (ML), enabled systems and its impact on testing processes. The main focus is to analyze how the inherent complexity of ML models affects traditional testing practices. The findings highlight that the dynamic nature and unpredictability of ML models require new testing approaches to handle challenges like non-deterministic behavior and evolving data, suggesting that current methods are often insufficient for testing ML systems effectively.

Add charts using various perspectives regarding the concepts and approaches in the 10 papers.

Article
A Secondary Study on AI Adoption in Software Testing
The Future of Software Testing: AI-Powered Test Case Generation and
Using Quality Attribute Scenarios for ML Model Test Case Generation
Understanding the Complexity and Its Impact on Testing in ML-Enabled
Test and Evaluation Best Practices for Machine Learning-Enabled Softw
An Empirical Study of Testing Machine Learning in the Wild
Applying Machine Learning Analysis for Software Quality Test
LeCov: Multi-level Testing Criteria for Large Language Models
The Role of Artificial Intelligence and Machine Learning in Software Tes
Applying Machine Learning in Software Testing: A Survey of Challenges

Table 3
Summary of Approaches and Features in Software Testing Research

The 3 paragraphs with similarities/differences between the 10 papers (from various perspectives). **Similarities and Differences Among the 10 Papers** The 10 papers presented share a common focus on improving software testing through advanced methods such as Artificial Intelligence (AI), Machine Learning (ML), and other innovative approaches. A notable similarity is the growing trend of integrating AI/ML techniques into software testing practices, aiming to enhance test automation, fault detection, and validation. Most of the studies emphasize the importance of adopting new technologies to handle the increasing complexity and dynamic nature of modern software systems, especially those enabled by ML. However, the papers vary in their scope, with some papers primarily discussing the theoretical aspects of AI/ML adoption (such as barriers and challenges), while others delve into specific methods for test case generation, validation, and performance evaluation.

In terms of methodology, several studies use empirical case studies or surveys to analyze real-world applications of AI/ML in software testing, while others propose novel frameworks or testing criteria, like the LeCov framework for Large Language Models (LLMs). While some papers focus on AI-powered automation tools for test generation, others consider how quality attributes or the complexity of ML models affect testing approaches. Differences also emerge in terms of the benchmarks and datasets used, with some studies employing general-purpose ML models or case studies, while others use specific datasets to evaluate the effectiveness of their proposed methods. Despite these variations, all papers recognize the need for tailored testing approaches that account for the unique characteristics of modern, complex software systems.

The papers also differ in their treatment of challenges and solutions. For instance, some papers focus on addressing the barriers to AI/ML adoption in software testing, such as lack of expertise or integration difficulties, while others discuss the complexity of testing machine learning models in dynamic, real-world environments, where data and conditions are unpredictable. While some articles concentrate on improving test case generation through AI, others emphasize the testing of specific attributes of machine learning systems, such as model performance or security. Additionally, the

Article	Approach	
A Secondary Study on AI Adoption in Software Testing	AI Adoption, Barriers in Adoption	
The Future of Software Testing: AI-Powered Test Case Generation and Validation	AI-Powered Test Generation	
Using Quality Attribute Scenarios for ML Model Test Case Generation	Quality Attribute Scenarios	
Understanding the Complexity and Its Impact on Testing in ML-Enabled Systems	ML Model Complexity, Testing Impact	
Test and Evaluation Best Practices for Machine Learning-Enabled Software Systems	Best Practices for ML Testing	
An Empirical Study of Testing Machine Learning in the Wild	Real-World ML Testing Challenges	
Applying Machine Learning Analysis for Software Quality Test	ML for Quality Testing, Defect Detection	
LeCov: Multi-level Testing Criteria for Large Language Models	Multi-level Testing for LLMs	Attentic
The Role of Artificial Intelligence and Machine Learning in Software Testing	AI/ML in Test Automation	A
Applying Machine Learning in Software Testing: A Survey of Challenges	Machine Learning in Software Testing	

Table 4
Summary table of selected articles with approaches, tools, and case studies.

papers vary in their recommendations for future research, with some proposing improved test case generation methods and others advocating for a more holistic approach to testing, which includes continuous validation and better integration of quality attributes into the test process. Despite these differences, all papers highlight the importance of evolving testing strategies to meet the challenges posed by AI and ML technologies in modern software development.

4. Results

The research questions are repeated and an answer is offered.

In conclusion, this study aims to respond to the three primary research questions guiding this review. These questions focus on identifying existing tools for testing AI and ML systems, understanding the key aspects that these tools target (such as robustness, fairness, and explainability), and evaluating the strengths and limitations of these tools. Based on the 10 papers reviewed, the following insights were gathered:

Research Question 1: What tools and frameworks currently exist for testing ML and AI software systems?

Answer: The papers reviewed provide a comprehensive overview of various tools and frameworks available for testing AI and ML systems. These tools serve different purposes, from automating test cases to evaluating performance and fairness:

- **AI-powered test automation tools:** As discussed in "The Role of Artificial Intelligence and Machine Learning in Software Testing" and "The Future of Software Testing: AI-Powered Test Case Generation and Validation," AI algorithms are used to automate test case generation and validation, significantly enhancing testing efficiency by reducing manual intervention.

- **LeCov Framework:** "LeCov: Multi-level Testing Criteria for Large Language Models" introduces a framework that provides multi-level testing criteria designed specifically for Large Language Models (LLMs). This framework helps ensure model reliability, fairness, and robustness across different contexts.

- **Scenario-based Testing:** The "Using Quality Attribute Scenarios for ML Model Test Case Generation" paper focuses on using quality attribute scenarios to test ML models. These scenarios assess critical model attributes such as performance, robustness, and scalability under different conditions.

- **Real-world Testing Tools:** Papers like "An Empirical Study of Testing Machine Learning in the Wild" highlight tools designed for real-world applications. These tools aim to handle dynamic and unpredictable environments, advocating for continuous model validation as they are deployed in real-world scenarios.

Research Question 2: What testing aspects (robustness, fairness, explainability) do these tools target?

Answer: The tools discussed in the papers focus on key aspects of AI and ML testing, including robustness, fairness, and explainability:

- **Robustness:** Several papers emphasize robustness testing, particularly to ensure that AI/ML systems are resilient to unexpected data and edge cases. For example, "Understanding the Complexity and Its Impact on Testing in ML-Enabled Systems" explores the challenges of testing for robustness, especially as ML models become more complex and unpredictable.

- **Fairness:** Fairness is a significant focus, particularly for AI systems involved in decision-making. Papers like "The Role of Artificial Intelligence and Machine Learning in Software Testing" focus on fairness testing frameworks that aim to mitigate bias and ensure equitable outcomes across diverse populations.

- **Explainability:** Explainability is another critical testing aspect, with several papers exploring methods to improve model transparency. "LeCov: Multi-level Testing Criteria for Large Language Models" emphasizes structured testing to enhance interpretability, while "Software Testing for Machine Learning" discusses how explainability tools can be integrated into testing to enhance model transparency and trustworthiness.

Research Question 3: What are the strengths and limitations of the identified tools?

Answer: The strengths and limitations of the identified tools can be summarized as follows:

Strengths:

- **Efficiency:** AI-powered tools, such as those discussed in "The Future of Software Testing: AI-Powered Test Case Generation and Validation," significantly enhance testing efficiency. These tools can automate large-scale test case generation, reducing the time and effort involved in testing.
- **Comprehensive Testing:** Tools like LeCov offer multi-level testing that addresses a wide range of model characteristics, such as robustness, fairness, and performance. These comprehensive testing approaches provide a more holistic evaluation of AI and ML models, ensuring that multiple aspects of model performance are thoroughly assessed.
- **Real-world Applicability:** Testing tools designed for real-world environments, as explored in "An Empirical Study of Testing Machine Learning in the Wild," are vital for ensuring that AI and ML models can handle unpredictable, real-world conditions. This makes these tools especially relevant for practical applications where models must be tested in dynamic and changing environments.

Limitations:

- **Complexity:** A common limitation across several tools, particularly those discussed in "Understanding the Complexity and Its Impact on Testing in ML-Enabled Systems," is the increasing complexity of AI and ML models. As models grow more sophisticated, it becomes more difficult to test every possible scenario, leading to gaps in coverage or a limited scope of testing.
- **Adoption Barriers:** According to "A Secondary Study on AI Adoption in Software Testing," many organizations face barriers to adopting AI-powered testing tools. These barriers include a lack of expertise, difficulties in integrating new tools with existing testing processes, and the high cost of deployment.
- **Limited Focus on Specific Aspects:** While some tools are highly specialized, focusing on certain testing aspects like robustness or performance, they may not fully address other critical attributes such as fairness or explainability. For instance, LeCov excels in robustness and fairness testing for LLMs but may not fully address explainability in other types of AI models.

5. Limitations of the Study

Answer: Despite the valuable insights gained, this study has several limitations:

- **Selection Bias:** The review is based on a selected set of 10 papers, which may not fully represent the entire landscape of AI and ML testing tools. Some relevant tools and frameworks may not have been covered.
- **Rapid Evolution of AI and ML Technologies:** AI and ML technologies evolve rapidly, and the tools discussed may soon become outdated. As such, the findings of this review may be subject to future changes and developments in the field.
- **Diversity of Test Cases:** The test cases discussed in the papers are diverse, but the methods and tools may not be universally applicable across all types of AI and ML systems, particularly as new models emerge.
- **Complexity:** As noted in "Understanding the Complexity and Its Impact on Testing in ML-Enabled Systems," the increasing complexity of ML models makes it difficult to test every aspect thoroughly. Testing frameworks often struggle to evaluate the full range of system behaviors, especially for large-scale systems.
- **Barriers to Adoption:** "A Secondary Study on AI Adoption in Software Testing" highlights barriers to the adoption of AI-powered tools, including a lack of expertise and integration issues within existing testing environments.
- **Limited Focus on Certain Attributes:** Some tools focus primarily on robustness and reliability, while fairness and explainability may not be fully addressed. For example, while LeCov is excellent for testing LLMs, its scope may not extend to other important aspects such as fairness across different domains like facial recognition or security.

References

Al Khan, Remudin Reshid Mekuria, R.I., 2020. Applying machine learning analysis for software quality test doi:https://doi.org/10.1109/ICCQ57276.2023.10114664.

Campeanu, G., 2018. Gpu support for component-based development of embedded systems.

Davila, N., Nunes, I., 2021. A systematic literature review and taxonomy of modern code review. Journal of Systems and Software 177, 110951. URL: https://www.sciencedirect.com/science/article/pii/S0164121221000480, doi:https://doi.org/10.1016/j.jss.2021.110951.

Dusica Marijan, Arnaud Gotlieb (Simula Research Laboratory, N., 2021. Software testing for machine learning doi:https://doi.org/10.48550/arXiv.2205.00210.

J. Chandrasekaran, T. Cody, N.M.E.L.L.F.V.T.N.S.I.U., 2022. Test and evaluation best practices for machine learning-enabled software systems doi:https://doi.org/10.48550/arXiv.2310.06800.

Junming Cao, Bihuan Chen, L.H.J.G.K.H.X.P., 2021. Understanding the complexity and its impact on testing in ml-enabled systems doi:https://doi.org/10.48550/arXiv.2301.03837.

Katja Karhu, Jussi Kasurinen, K.S., 2021. A secondary study on ai adoption in software testing doi:https://doi.org/10.48550/arXiv.2504.04921.

Kitchenham, B., 2004. Procedures for performing systematic reviews. Keele, UK, Keele Univ. 33.

Kitchenham, B., Charters, S., 2007. Guidelines for performing systematic literature reviews in software engineering 2.

M. Openja, F. Khomh, A.F.M.A.P.M.C.Z.M.Y.U.C.A.E.H.Q.U.C., 2021. An empirical study of testing machine learning in the wild doi:https://doi.org/10.48550/arXiv.2312.12604.

Mohammad Baqar, R.K., 2022. The future of software testing: Ai-powered test case generation and validation doi:https://doi.org/10.48550/arXiv.2409.05808.

Rachel Brower-Sinning, Grace A. Lewis, S.E.I.O., 2022. Using quality attribute scenarios for ml model test case generation doi:https://doi.org/10.48550/arXiv.2406.08575.

A. Ramadan (Üsküdar University, Computer Engineering Department); Dr. H. Yasin (Dhofar University, C.E.D.P.D.B.P.U.C.E.D., 2020. The role of artificial intelligence and machine learning in software testing doi:https://doi.org/10.48550/arXiv.2409.02693.

Xuan Xie, Jiayang Song, Y.H.D.S.F.Z.F.J.X.L.M., 2024. Lecov: Multi-level testing criteria for large language models doi:https://doi.org/10.48550/arXiv.2408.10474.