

Addressing Uncertainty in AI-Driven Automation: Analyzing False Positives, Ignored Vulnerabilities, and Result Reliability in Preexisting Tool

Makhdoom Abdulhamid Shaikh

Department of Cyber Security
Chikitsak Samuha's Patkar-Varde College
shk.makhdoom27@gmail.com

Abstract—Artificial intelligence and machine learning models are being leveraged to automate preexisting verification, testing, and security analysis tools, with substantial benefits for operational efficiency but at the cost of introducing severe uncertainty in final results. Notably, automated wrappers often raise false positives (Type I errors) due to excessively strict feature thresholds or exhibit false negatives (Type II errors) by ignoring potential underlying issues in complex or obfuscated situations. This paper looks at where uncertainty in results comes from in AI-driven automation tools, using multimodal verification and semantic assessment. It provides a way to categorize automation confusion and explains how false positives and overlooked vulnerabilities can affect operations. The paper concludes by discussing a hybrid evaluation framework which seeks to address these concerns.

Keywords— AI automation, false positives, vulnerability suppression, tool wrappers, result uncertainty, multimodal verification, alert fatigue.

Introduction

In today's software engineering, compliance automation, and digital onboarding, the need for efficiency has led to the use of artificial intelligence (AI) tools. Instead of building verification processes from the ground up, companies are using machine learning and language models to enhance existing tools like user interface testing frameworks and identity verification systems.

While these AI tools can greatly reduce manual work and speed up processes, they also create new challenges. On one hand, AI tools might wrongly flag issues due to strict settings or pattern recognition, identifying problems in systems that are actually functioning properly. On the other hand, they may overlook real issues because their pattern recognition is too broad or lacks context, leading to missed problems.

This paper explores both the benefits and drawbacks of AI automation tools. It aims to understand how these tools can affect the quality of results and suggests possible mixed approaches to improve their effectiveness.

II. Background and Related Work

Many papers have talked about the weaknesses of automation tools in different fields, from evaluating testing tools to educational methods.

A. Review of AI Testing and Automation Tools

Research on AI-based test automation tools shows that while machine learning classifiers do well with common situations, they often struggle in changing or unexpected settings. Many of these tools use basic statistical measures to judge quality, which can lead to many false alarms.

B. Problems with Reactive Detection and Lack of Context

A recent study on test-driven development and natural language processing (NLP) assessment found that tools relying on basic statistics or pattern recognition can be easily tricked. Similarly, an AI tool that relies on simple pattern matching or set thresholds to find problems may miss subtle but important issues in the code that appear normal at first glance.

C. The Automation Dilemma: Speed vs. Uncertainty

The main issue with these tools is that they offer the speed and reduced manual efforts of automation but hide the

quality risks of using AI. Existing tools have known limitations, especially regarding their accuracy and false alarms. In contrast, an AI-based tool introduces uncertainty: engineers may not realize all the ways the tool could fail.

III. Problem Statement & Research Questions

This paper aims to explore the uncertainty created by using AI wrappers by asking these questions:

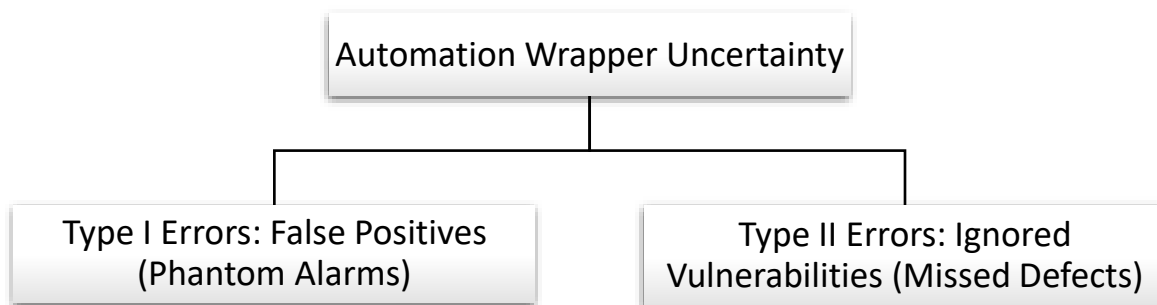
RQ1 (False-Positive Triggers): How do AI automation wrappers cause false positives by being too strict with their rules or patterns?

RQ2 (Vulnerability Suppression): How can wrappers overlook real problems and fail to find actual issues?

RQ3 (Hybrid Remediation): Can a mixed approach using different methods improve the problems related to AI wrappers?

IV. The Dual-Sided Nature of AI Automation Wrapper Uncertainty

Using AI to improve existing tools creates uncertainty in two ways: it can generate false positives and hide real problems. This can be shown in a diagram:



1. Type I Errors: False Positives and Phantom Alarms

An AI automation tool can report system failures by mistake if its sensitivity settings are too strict. Small, harmless changes—such as minor network issues, slight shifts in the screen layout, or tiny timing differences—are often wrongly labeled as serious errors. This creates many false alarms, which force staff to waste time checking them and lead to frustration.

2. Type II Errors: Missing Real Problems

Attempts to reduce alert fatigue by making classification rules too broad often cause the system to miss real threats. When security tools are set to ignore too many alerts, they may fail to detect hidden bugs, unusual vulnerabilities, and functional errors. This leaves the system at risk while creating a false sense of security.

V. Proposed Hybrid Framework and Approach

This paper suggests a hybrid framework to solve the problems mentioned in the research questions. It combines basic tool data with advanced analysis of intentions.

A. Multimodal Signal Ingestion

The proposed system collects information from three different sources at the same time:

1. Tool telemetry: This includes basic data from tools, such as their status, output, and errors.
2. Environmental artifacts: This refers to the features of the environment where the tools run, like codec and latency.
3. Semantic context: This captures the purpose behind the input, whether it's code or configuration, through vector embeddings.

Combining these three types of information helps the framework build a more complete understanding of the situation than using just one source.

B. Composite Risk-Scoring: Avoiding Single-Point-of-Failure Architectures

To prevent problems that come from relying on a single component, the framework uses composite risk-scoring. This means that several classifiers work together to create an overall risk assessment.

$$\text{Overall Risk} = \sum_{i=1}^n w_i \cdot \mathcal{F}_i(\text{Telemetry}_i)$$

Each $\mathcal{F}_i$ stands for a different type of risk classification, like Random Forest or Extra Trees, and w_i are weights that show how much each telemetry stream, such as tool logs, environment, or semantic intent, contributes.

C. Feedback and Override Mechanisms

If an engineer ignores an automation suggestion or overrides it because it is wrong, the system changes the risk classification limits. This helps prevent similar problems in the future.

VI. Implementation and Architecture Concepts

The framework can be used as a Python application with computer vision (OpenCV) and traditional machine learning (Sklearn).

Module	Core Technology	Primary Function
Visual Ingestion	OpenCV / Bounding Boxes	Captures user interface (UI) element states and layout shifts.
Forensic Classifier	ExtraTreesClassifier ($N = 100$)	Evaluates visual sharpness and noise variance.
Log Analyzer	RandomForestClassifier	Inspects execution traces and spectral error rates.
Semantic Filter	BERT Semantic Embeddings	Validates functional intent against test requirements.
Feedback Engine	Dynamic Threshold Tuning	Adapts confidence limits based on human overrides.

TABLE I: CORE MODULES AND TECHNOLOGIES OF THE PROPOSED FRAMEWORK

VII. Operational Impacts: Alert Fatigue, Blind Trust, and Lost Productivity

The problems caused by the increased use of AI automation tools can be summarized as follows:

1. Alert fatigue: When there are too many false alarms, engineers start to distrust the recommendations made by automation tools.
2. Compliant blindness: When automation effectively hides problems, engineers begin to trust these tools too much, even when they should not.

3. Lost productivity: Engineers spend a lot of time trying to figure out why automation tools highlight or ignore certain issues, which takes time away from actual engineering work.

VIII. Limitations and Future Work

This paper offers a basic understanding of how to deal with problems caused by AI automation wrappers, but it has some limitations. One major issue is that methods like Random Forest or Extra Trees may not work well in new situations, and measuring similarity in meaning does not truly reflect what someone intends. More research is needed to create fully independent methods for calibration and verification. Also, future work needs to concentrate on fitting these mixed models to live factory settings where system information changes quickly. Creating feedback systems that can grow will also be important to lessen human help and stop too many warnings. In the end, closing the distance between finding problems automatically and truly understanding the situation is still a main difficulty for future software reliability.

IX. Conclusion

AI automation tools can help reduce manual work by improving existing tools, but they also create uncertainty about results because of incorrect positive findings and overlooked weaknesses. By identifying these risks, understanding how they affect operations, and using a mixed approach that combines tool data with careful checks, organizations can take advantage of automation's speed while ensuring strong verification standards. In the end, matching machine speed with people checking things makes sure big mistakes are reduced. This ongoing adjusting lets teams keep systems working well without getting tired of warnings or just believing what machines say.

X. References

- [1] M. S. Akbar, "Beyond Detection: Designing AI-Resilient Assessments with Automated Feedback Tool to Foster Critical Thinking," arXiv preprint arXiv:2503.23622, 2025. [Online]. Available: <https://arxiv.org/abs/2503.23622>
- [2] V. Garousi et al., "AI-powered software testing tools: A systematic review and empirical assessment of their features and limitations," arXiv preprint arXiv:2409.00411, 2024. [Online]. Available: <https://arxiv.org/abs/2409.00411>
- [3] M. S. Akbar et al., "Uncertainty Analysis in AI-Driven Automation and Verification Wrappers," Social Science Research Network (SSRN) Electronic Journal, SSRN Abstract: 7138178, 2026. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=7138178