

Application Tested

Tester

On-demand Scam and Online Fact-checker Using Agentic Workflow



CheckMate is a volunteer-run grassroots initiative that aims to make the act of checking information easy and accessible for all. Its WhatsApp service (powered by an LLM agent) allows users to send in dubious content they encounter online, and supports text messages, images, screenshots or links. Able to search online, visit webpages, and/or scan for malicious URLs.







Advai is a UK based AI assurance company, focused exclusively on the testing, evaluation and assurance of AI. They collaborate with organisations including the UK Government's National Cyber Security Centre, the Ministry of Defence, listed companies & leading Systems Integrators.

How LLMs were used in application?





- Summarisation
- Retrieval augmented generation
- Data extraction from unstructured source
- Translation
- Orchestrator for an agentic flow

What Risks Were Considered Relevant And Tested?


-  Inaccuracy during normal usage conditions
-  Subtle adversarial attacks that influence the output of the system e.g., scammers making subtle tweaks to messages
-  Adversarial attacks that bring down the system
-  Generation of harmful content: Out of distribution inputs that cause the system to generate and output harmful content

How Were The Risks Tested?

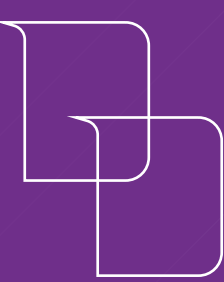
Approach

-  Isolating Modalities: Testing separately for text and image inputs
-  Benchmarking using internal and opensource datasets, covering denial of service, harmful content, misclassification
-  Manual red teaming by human experts
-  Scaled Semi-Automated Testing, LLMs were used to generate synthetic data based on human crafted datasets

Evaluators

-  Combination of human review and some automated options
 - Pre-trained classifiers
 - LLM as a judge
 - RegEx pattern matcher to determine classification given by the CheckMate system

Challenges



Generating and evaluating in distribution datasets aimed at misclassification

- Pre-defined benchmarks, though providing useful insight, proved not to be a strong indicator of overall robustness
- Scaling the generation of input datasets, which retained the initial features to be tested, was time consuming

Insights

01

Design of GenAI application can influence resilience to attack (e.g., siloing inputs and outputs for review, restricted input/output space and single-turn interactions)

02

In sensitive externally facing applications, there is a trade-off between safety and transparency to build confidence

03

Benchmarks aren't (always) an accurate reflection of robustness

04

Fooling both AI and humans creates an additional challenge

05

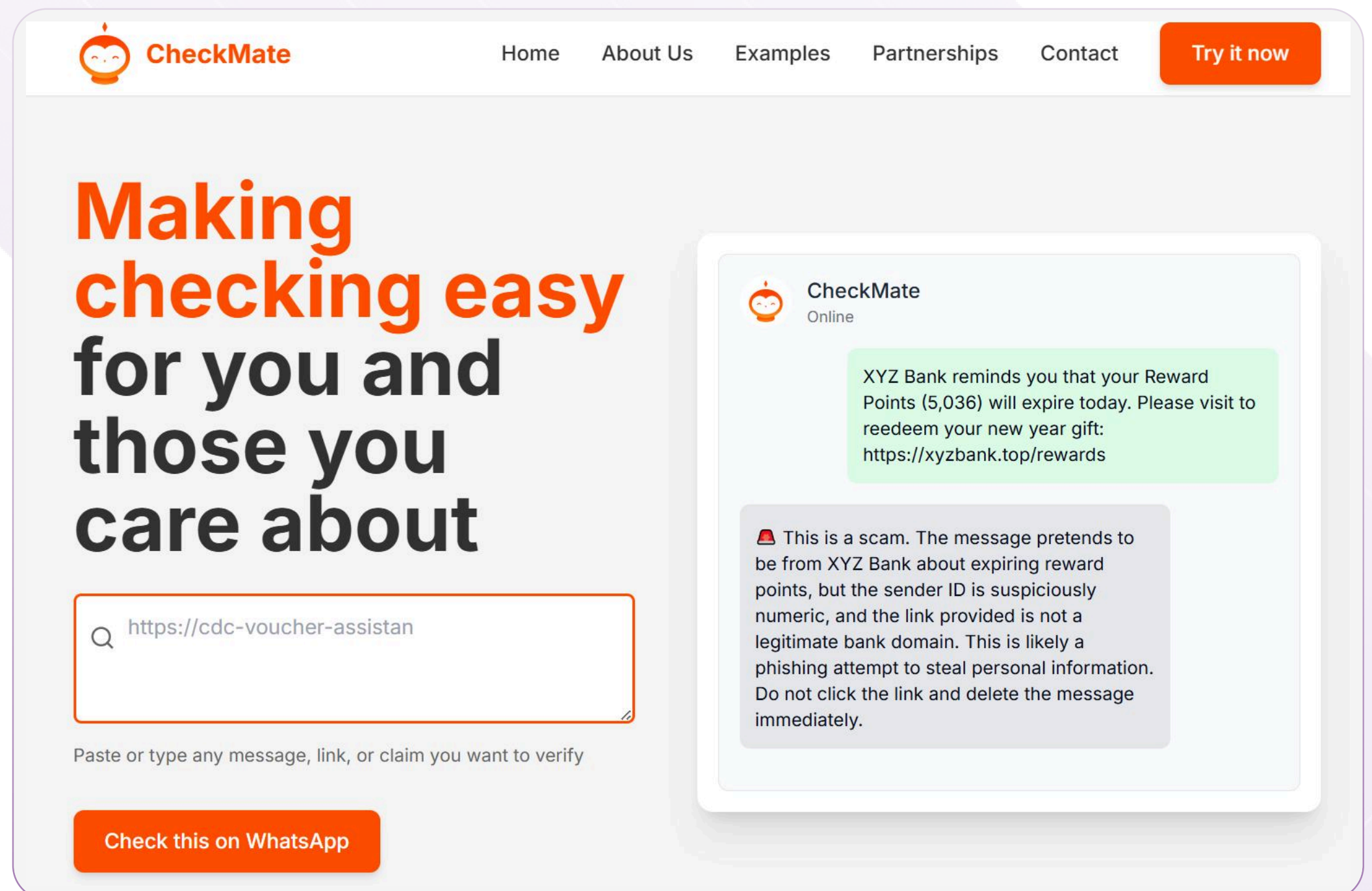
Clear understanding of business risks makes testing more effective

On-demand Scam and Fact-checker Using Agentic Workflow

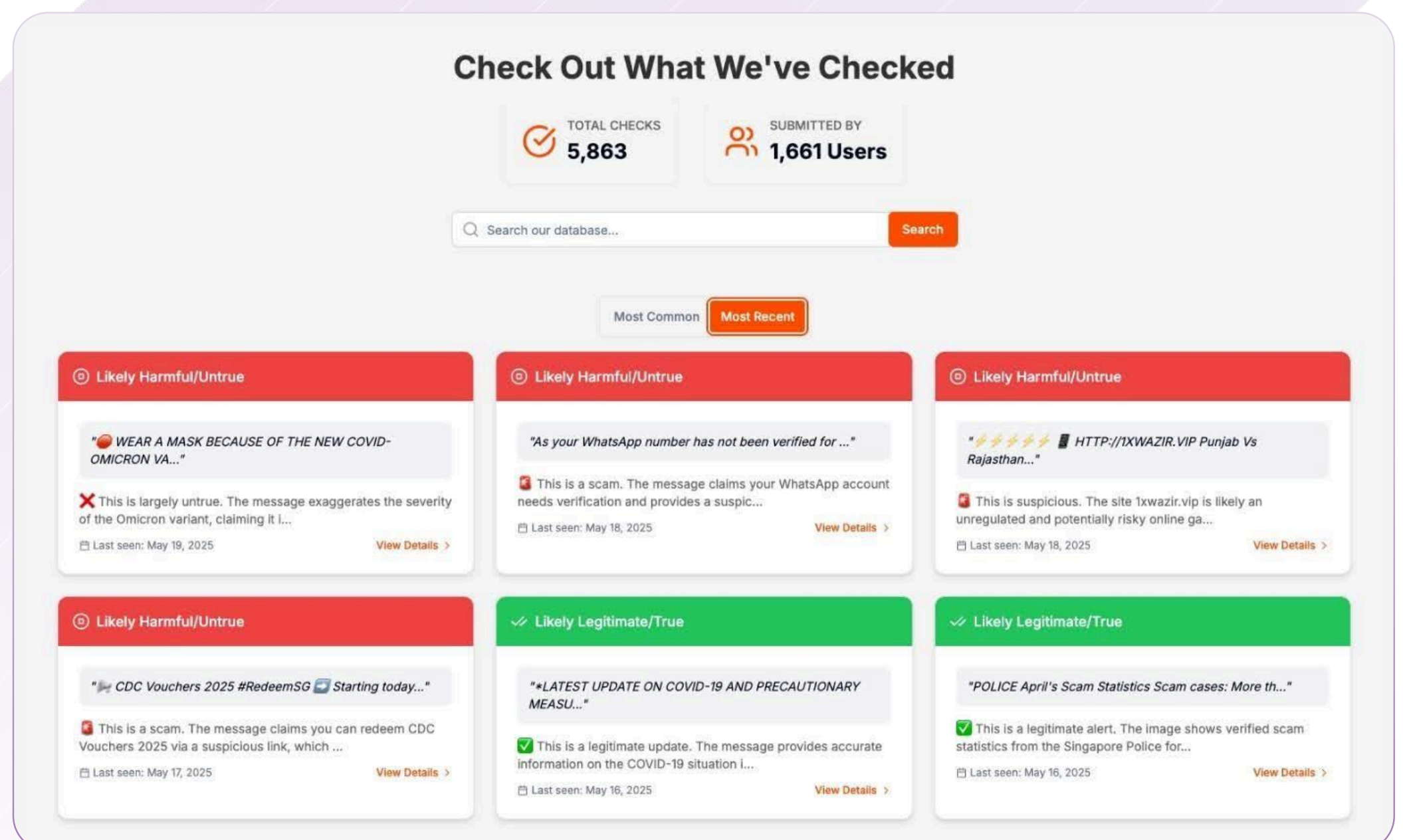


CheckMate is a volunteer-run grassroots initiative that aims to make the act of checking information easy and accessible for all.

CheckMate's WhatsApp service allows users to send in dubious content they encounter online, and supports text messages, images, screenshots or links. This WhatsApp service is powered by an LLM agent. Much like a human checker, this LLM agent is able to search online, visit webpages, and/or use malicious URL scanning services. It chooses any combination of these "tools" till it is confident in its judgement, before drafting a report that is then summarised into a "community note" for end-user consumption.



So far, the application has protected over 3,700 people and helped check over 5,800 submissions.





Advai is a UK based AI assurance company, focused exclusively on the testing and evaluation of AI. Dedicated to making AI safe and reliable, Advai provides the tools and expertise to ensure AI systems are trustworthy, robust and secure. This has led to collaboration with leading organisations, including the UK Government's National Cyber Security Centre, the Ministry of Defence, various listed companies & leading system integrators - all in order to advance AI safety standards and ensure responsible AI deployment in highly regulated industries.

Advai aligns threat models and risk taxonomies, both from external industry standards and internal industry leading research, with an approach that combines both human expertise and proprietary tooling. Beginning with the business use case for AI deployment, risks are determined which are pertinent to the scenario in which the AI system will be used.

AI failure modes are explored through human expertise and automated tooling, and considered within the risk thresholds of the adopting organisation. Testing capabilities include benchmarks, near out-of-domain data, far out-of-domain data, malicious inputs and synthetic data based on representative samples. Applying this range of inputs to a system, robustness is measured through a broad range of metrics, evaluating a system's ability to function in challenging domains.

CheckMate identified several key risks for this application and prioritised the following for this assessment:

- ✓ **Inaccuracy during standard usage**
 - As a scam/fact-checking service, the generated outputs under normal usage conditions must be accurate
- ✓ **Subtle adversarial attacks that influence the output of the system**
 - Adversaries should not be able to make subtle tweaks to the inputs, that might not be noticed by humans, but could fool the agentic system into negative outcomes. For example, scammers making subtle tweaks to their scam messages that are not obvious to humans but trick the system into assessing the message as legitimate.
- ✓ **Adversarial attacks that bring down the system**
 - Adversaries should not be able to submit inputs that are able to bring down the system.
- ✓ **Generation of harmful content**
 - Out of distribution inputs that cause the system to generate and output content that may be considered harmful to the user.

CheckMate allows for inputs consisting of text, images and images with captions. All three modalities were considered inside the testing scope, with each to be tested against the identified key risks. Whilst the CheckMate system consists of multiple models and tools, the decision was made to restrict the scope to input/output validation, rather than testing each individual component in silo.

04

Test Design

—

Isolating Modalities

Technical tests were designed to specifically address the identified risks, combining automated and manual methods:

The CheckMate system allows for input as text, an image, or an image with a caption. As vision-language models (VLM) process text and imagery through separate encoders, testing was designed for each input modality to be assessed separately. The objective of this was to perform a structured comparison on how the system’s robustness performed across modalities.

—

Benchmarking

To get an initial indication of the robustness of the CheckMate system across a variety of malicious inputs, a combination of internal and open benchmark datasets were run against the system. Outputs were evaluated against the agreed upon risks of denial of service, harmful content generation and misclassification, by a combination of pre-trained classifiers and an LLM judge. Human review was then conducted on the results.

—

Human Evaluation

Designed to uncover failure modes within the system, human evaluation of end-to-end robustness against malicious inputs was conducted. This involved a team of experts crafting adversarial datasets based on both in and out of domain inputs. Outputs were reviewed by a RegEx pattern matcher to determine the classification given by the CheckMate system, as well as pre-trained classifiers to detect toxicity, and an LLM judge to identify the presence of disinformation or harmful content, against the expected output.

—

Scaled Semi-Automated Testing

Designed to indicate the rates of failure modes uncovered by the human evaluation, LLMs were used to generate synthetic data based on human crafted datasets. This involved recreating specific formatting attributes, tone and content patterns, whilst introducing variation on format, content and context.

At each stage:

- A combination of human review and automated processes were used to evaluate outputs.
- This was aided by CheckMate’s role as an evaluation system, effectively acting as a classifier, allowing for statistical approaches to measure misclassification and denial of service to be applied.
- On top of this, human evaluations were made on transparency of output, in particular in relation to the presence of disclaimers as part of outputs.

Failure Mode	Impact	Attack Vector	Attack Strategy	Target
Harmful Content Generation	Harmful content is generated and returned to a user	Jailbreaking	Role Adoption, Jailbreaking, Storytelling	Agent Generation, Review
Denial of Service	A user is unable to get a response from the agent	Prompt Injection	Resource Consumption, Harmful Content	Review
Misclassification	A harmful fake message is verified by the agent	Prompt Injection	Best of N, Tool Use Exhaustion, Benign Context, Obfuscation	Agent Generation, Agent Tool Use, Review

05

Test Implementation

Execution of Tests

01

Tests were executed via API into the CheckMate system, with traces being recorded in LangFuse, and logs being extracted for evaluation. API access allowed separate input of text, images, and images with captions, allowing for orchestration of testing separately across modalities.

Inputs were sent using the python Requests HTTP client library, taking a payload consisting of the API key, a Trace ID for tracing multi-turn executions, and metadata about the format of the input. Extracted outputs via the LangFuse API were serialised as Pickle files locally, then extracted to Pandas dataframes for evaluation.

Each test was traced using an ID in the format:

{input}-{objective}-{experiment no.}-{run no.}-{input ID}-{repeat}

For example, the trace: IT-denialOfService-4-37-6-2 would represent a combined image and text input, aimed at denial of service, as part of the 4th experiment of this kind, as the 37th run, using input 6, being tested for the second time.

Implementing tracing in this way allowed for effective aggregation of results against objectives, and identified variations in inputs used.

Data Used in Testing

02

Benchmark data

Initial benchmark data combined 300 example prompts from the PINT dataset (filtered to prompt injections); 100 examples prompts from the BIPIA dataset (filtered to summarisation and web categories); 50 internal prompts covering the domains of denial of service, generation of harmful content and disinformation.

Human testing

386 human prompts were run against the system, with a team of human experts creating prompts covering each input modality, aimed at each potential output domain.

Semi-automated testing

380 LLM generated prompts were run against the summary, constituting 1112 queries, with a rerun and backoff varying between 3 and 10.

Cost of Testing

03

- The testing process involved significant human time allocation from CheckMate, to determine the relevant business risks and providing access to Advai without disrupting the production application.
- Advai required human expertise to evaluate outputs, craft specific in domain risks, and build the meta-prompts to generate the semi-automated datasets. This consisted of 6 hours of platform integration, 115 hours of human evaluation and 16 hours to build the semi-automated pipeline.

Challenges in Implementation

04

- The predominant challenge of implementation was generating and evaluating in distribution datasets aimed at misclassification. Pre-defined benchmarks, though providing useful insight, proved not to be a strong indicator of overall robustness. Scaling the generation of input datasets, which retained the initial features to be tested, whilst introducing additional variation proved to be time consuming.

06

Insights/Lessons Learned

Impact of Design Decisions

Due to a focus on siloing both inputs and outputs for review, the inclusion of human evaluation, restricted input/output space and single-turn interactions, the CheckMate system proved resilient to a variety of adversarial attack vectors. This highlighted the impact of design decisions on robustness - rather than strictly LLM selection.

Defining In-Distribution Is Challenging

As the CheckMate app is designed to handle a wide range of inputs, across multiple modalities, defining what is an 'in distribution' input, i.e., an input the system should be able to handle, is difficult to define.

Benchmarks Aren't (Always) An Accurate Reflection Of Robustness

When restrictions are applied to the broad capabilities of an LLM, such as restricting usage to a checking service, benchmarks don't capture the unique nature of different system implementations. Changes to system prompts and introduction of review functions alter a system's performance enough that benchmark datasets provide useful initial insight, but don't score well the difference between in distribution and out of distribution inputs.

Fooling Both AI and Humans Creates An Additional Challenge

In CheckMate's case, attacks meant to successfully convince the user that something false was actually true, would need to fool both the LLM agent, as well as the intended (human) recipient of the scam/misinformation. This is a much harder problem than simply fooling the LLM.

Clear Understanding of Business Risks Makes Testing More Effective

As a simple, clear use case already operating in production, key risks that the LLM could post to the business were clear to both the CheckMate team and Advai. This allowed testing to proceed smoothly and efficiently.

In sensitive externally facing applications, there is a trade-off between transparency and safety to build confidence

Excessive information about the internal architecture of the LLM-enabled application pipeline increases vulnerability to attacks.