AI Incident Copilot

AI의 장애 분석을 로그와 대조하고 사람이 최종 판단하는 도구

검증 기준일 2026-10-15 | 공개 저장소 기준

25개
자동화 테스트
CI 두 환경에서 통과

4건
공개 실행 기록
원본 JSON 보존

2개
CCTV 평가 미충족
총 6개 판정 중

프로젝트 개요

운영 장애 로그(JSON)를 입력하면 로컬 LLM이 의심 원인, 참조 로그 ID, 다음 확인 방법, 아직 모르는 것을 제안하는 Python 명령줄 도구다. 프로그램은 응답 형식과 인용 ID의 존재 여부를 검사하고, 담당자는 로그의 의미와 실제 조치의 타당성을 판단한다. [S1, S2]

제작 동기는 ITS(지능형교통체계) CCTV/VMS 유지보수 경험에서 출발했다는 작성자의 설명에 근거한다. 개인 경력은 별도로 검증하지 않았다. [S1]

확인 항목	증빙 범위
AI 활용	운영 분석: Ollama llama3.2:1b / 개발 보조: Claude Code
자동화 테스트	소스 25개 확인, 과거 CI의 Python 3.11·3.12 각각 25 passed
공개 실행 증거	evidence/ JSON 4건 확인. 이번 제작 중 모델 재실행은 하지 않음
CCTV 전후 평가	기준 5개를 적용한 6개 판정: 충족 4개, 미충족 2개
데이터 성격	모든 입력 로그는 가상 데이터. 실제 현장 실증이나 생산성 수치 측정은 아님

핵심 구분
로그 ID가 존재하고 응답 형식이 맞아도, 원인 분석이나 조치 제안이 정확하다는 뜻은 아니다. 실제 서비스 적용 성과, 정확도, 시간 절감률을 이 증거만으로 주장하지 않는다.

작성 범위

참조 대화는 첨부 화면의 일부 원고만 확인 가능했다. 해당 화면의 역할 분담·테스트 25개 서술을 참고하고, 상세 사실은 공개 소스와 원본 JSON, 평가 문서, CI 로그에 대조했다. 원고 전체를 복원했다고 주장하지 않는다.

01 AI 활용 방식과 프로그램 구조

운영 중 AI 활용

Ollama의 로컬 /api/chat에 시스템 지시와 장애 로그를 보낸다. 기본 모델은 llama3.2:1b, 요청 설정은 format=json, temperature=0.2, stream=false, 기본 제한 시간은 120초다. 프롬프트는 새 로그 ID를 만들지 말고 원인을 확정하지 않도록 요구한다. 이 지시 자체가 준수를 보장하지는 않는다. [S2]



구현 연결: 입력(data_loader.py·models.py) → 모델 호출(ollama_client.py) → 검사(validator.py) → 출력·저장(cli.py). 사람의 검토 여부를 기술적으로 강제하지는 않는다. [S2]

검증과 기록의 구체적 범위

키 앞뒤 공백과 ID의 대괄호·공백을 정규화한다. 미존재 ID, 참조 ID 없음, 빈 의심 원인, 빈 다음 확인 목록, 특정 과확신 표현, 비어 있는 unknowns를 경고한다. 원본 응답은 보존하며 새 결과 파일명에는 짧은 UUID를 덧붙인다. [S2]

종료 코드 0은 응답 수신·형식 검사 통과, 1은 모델 응답 수신 실패, 2는 응답 형식 실패, 3은 입력 파일 오류를 뜻한다. 내용 경고가 있어도 종료 코드가 0일 수 있다. [S2]

개발 과정의 AI 활용과 사람의 기여

README의 역할 기록에 따르면 Claude Code가 Python 코드·테스트·README 초안 작성, 명령 실행과 버그 수정을 보조했다. 사람은 요구사항·범위·출력 형식, 과장 표현 수정, 실제 결과 확인과 최종 판단을 맡았다. 이는 작성자 기록에 근거한 개발 과정 설명이다. [S1]

사람이 정한 설계 원칙
자동 검증 범위를 형식과 ID 존재 확인으로 제한하고, 실패 응답을 보존하며, CCTV 평가 기준을 결과 문서와 분리했다. 자동 조치 기능을 포함하지 않는 CLI 시제품이다.

02 공개 실행 증거 4건

아래는 저장소에 보존된 결과 JSON의 요약이다. 각 파일의 model_used는 llama3.2:1b다. 생성 시각은 JSON 문자열 표기를 옮긴 것으로 시간대는 기록되지 않았다. [S3-S6]

증거 / 생성 시각	원본에서 확인한 내용
S3 · INC-2026-0001 20260925_143501	order-api: 인용 LOG-005·006, 존재 확인 2/2. response_source=ollama, 스키마 유효. 의심 원인은 DB 커넥션 풀 사용량·대기 요청에 관한 서술.
S4 · INC-2026-0002 20260925_121403	notification-worker: 인용 LOG-101·102, 존재 확인 2/2. 구버전 success=true. 의심 원인은 사용자 문의의 반복. 다음 확인 방법: "일시적 확인" / "다시 확인".
S5 · INC-2026-0007 20260925_163256	CCTV 조치 전: 인용 LOG-601·602·605·606·607, 존재 확인 5/5. response_source=ollama, 스키마 유효. 의심 원인은 "Ping 무응답".
S6 · INC-2026-0008 20260925_163306	CCTV 조치 후: 인용 LOG-702·704·705·706, 존재 확인 4/4. response_source=ollama, 스키마 유효. 의심 원인은 "기존 어댑터의 정밀 고장 진단이 실시되지 않았음".

형식 검사 통과와 내용 품질의 차이

S3의 다음 확인 방법은 "DB connection pool 최대 크기 설정을 확인하고 최소 40으로 변경하십시오"다. 입력 LOG-007에는 설정이 100에서 40으로 줄었다는 이력이 있다. 이 제안은 원인일 수 있는 설정을 그대로 유지하게 할 수 있어 적절한 복구 조치로 받아들일 수 없다. ID 검사만으로 이런 내용 문제를 걸러내지 못한다. [S1, S3]

S4는 의심 원인과 unknowns에 사용자 문의 문장을 반복한다. S5는 확인 방법에 문장 반복과 한국어·독일어·힌디어 혼입이 있다. 네 파일 모두 validation_warnings가 빈 목록이지만, 이 사실은 분석 품질에 대한 합격 판정이 아니다. [S3-S6]

증거의 출처와 한계

S3·S5·S6에는 실제 호출을 나타내는 response_source가 있다. S4에는 response_source와 processing_status가 없다. evidence/README는 이를 해당 필드 도입 이전의 실제 호출 기록으로 설명한다. 따라서 S4의 호출 출처는 작성자 설명과 과거 생성 시점에 의존하며, 최신 파일과 같은 필드 검증은 불가능하다. [S7]

evidence/README는 로컬 outputs/와 바이트가 같은 사본이라고 설명하지만, 로컬 원본은 공개되지 않아 이번 제작에서는 원본과 사본의 일치 여부를 독립적으로 대조하지 못했다. [S7]

03 CCTV 어댑터 장애 전후 평가

가상 시나리오: 조치 전에는 ping 무응답·스위치 링크 down, 전원 미확인·인근 굴착 이력이 있다. 조치 후 입력에는 어댑터 교체, ping 복구, 기존 어댑터 정밀 진단 미실시가 기록돼 있다. 어댑터 교체 후 복구됐다는 사실만으로 내부 고장을 확정할 수는 없다. [S5, S6, S8]

평가 문서는 기준을 모델 실행 전에 고정했다고 기록하고 커밋 ffde538을 제시한다. 두 사례를 각각 1회 실행하고 재시도하지 않았다는 내용, Claude Code의 판정 초안을 사람이 2026-09-26 최종 검토했다는 내용은 저장소 문서의 이력 설명이다. [S8, S9]

대상 / 기준	판정	원본·평가 문서의 근거
전 / 1 과확정 방지	충족	"Ping 무응답"으로 증상만 서술. 어댑터 고장이라고 확정하지 않음. 분석 부실 때문에 충족했을 가능성도 있음.
전 / 2 구체적 확인 방법	미충족	스위치 포트 확인 문장을 반복하며 전원(LOG-604)·케이블/굴착(LOG-607) 단서를 구체적인 확인 방법에 반영하지 못함.
전 / 5 로그 ID 존재	충족	참조 ID 5개 모두 입력에 존재. 5/5.
후 / 3 복구 사실 반영	미충족	LOG-702를 인용했지만 의심 원인과 unknowns에 ping 복구 사실을 서술하지 않음.
후 / 4 불확실성 유지	충족	"정밀 고장 진단이 실시되지 않았음"을 서술. 어댑터 내부 결함으로 단정하지 않음.
후 / 5 로그 ID 존재	충족	참조 ID 4개 모두 입력에 존재. 4/4.

총 6개 판정 중 미충족 2개
고유 기준은 5개이며 공통 기준 5번을 전후에 각각 적용했다. 4/6이라는 수치를 프로젝트의 정확도·성공률로 해석하지 않는다. 사례는 조치 전 1건과 조치 후 1건뿐이다.

자동 검사로 잡히지 않은 문제

조치 전 next_steps 5개 중 같은 스위치 포트 문장이 3번 반복된다. 문장에 독일어 prüfen 등이 혼입됐지만 JSON 형식과 ID 존재 검사는 통과했다. 원문 혼입 문자열은 S5 링크에서 확인할 수 있으며, 이 PDF에서는 뜻이 불분명한 원문을 정상적인 한국어 응답으로 고쳐 제시하지 않았다. [S5, S9]

04 자동화 테스트 25개와 CI 증거

테스트 소스에서 validator 14개, CLI 8개, Ollama 클라이언트 3개를 확인했다. 아래는 각 테스트의 확인 목적 요약이다. 모두 오프라인 또는 모의 호출 기반으로, 모델 답변의 정확도를 측정하는 테스트가 아니다. [S10]

응답 검증 · 14개	CLI·클라이언트 · 11개
01 정상 응답에서 경고 없음	15 유효 오프라인 응답의 종료 코드·저장 필드
02 미존재 로그 ID 경고	16 잘못된 응답의 실패 코드·형식 오류 기록
03 무근거·과확신 응답의 복수 경고	17 같은 초 연속 실행의 파일·원본 응답 보존
04 키 앞뒤 공백 정규화	18 장애 파일 없음
05 대괄호 포함 로그 ID 정규화	19 잘못된 장애 JSON
06 비JSON 응답의 형식 오류	20 장애 입력 필수 키 누락
07 최상위 null 처리	21 logs 자료형 오류
08 최상위 배열 처리	22 오프라인 응답 파일 없음
09 ID 필드가 숫자인 경우 처리	23 연결 실패를 OllamaError로 처리
10 ID 문자열을 배열로 잘못 분해하지 않음	24 타임아웃을 OllamaError로 처리
11 필수 키 누락	25 HTTP 비정상 상태를 OllamaError로 처리
12 의심 원인의 잘못된 자료형	
13 빈 의심 원인 경고	
14 빈 다음 확인 목록 경고	

왼쪽: tests/test_validator.py · 오른쪽 15-22: tests/test_cli.py · 23-25: tests/test_ollama_client.py. 번호는 이 자료의 정리용이다. [S10]

원본 CI 로그에서 확인한 통과 기록

실행 36123300366 / 커밋 7aa2b357c794cc54a4c3d98af12f4e4b040afc73
Python 3.11: 25 passed in 0.21s · Python 3.12: 25 passed in 0.34s. 두 작업 모두 success. [S11]

CI 설정은 Ubuntu에서 push·PR마다 Python 3.11·3.12로 pytest tests/ -q를 실행한다. 위 수치는 과거 해당 실행의 결과이며, 기준 커밋 전체를 이번 제작 환경에서 새로 테스트했다는 의미는 아니다. [S10, S11]

05 오프라인 경고 데모·한계·개선

사람이 작성한 로그 ID 경고 데모

data/demo_manual_response_missing_id.txt는 사람이 미리 작성한 예시 응답이다. --offline-response 옵션으로 CLI 검증기에 넣으며 모델을 호출하지 않는다. 예시에 포함된 LOG-999는 입력 로그에 존재하지 않아 경고 대상이 된다. 이는 검증 로직의 동작 예시이며 실제 모델이 환각을 냈다는 증거가 아니다. [S2, S12]

실행 구분

실제 모델 호출: python run.py data/sample_incident_1.json

오프라인 데모: python run.py data/sample_incident_1.json --offline-response data/demo_manual_response_missing_id.txt

위 데모 명령은 한 줄로 연결해 실행한다. CLI는 화면에 실제 LLM 호출이 아님을 알리고 response_source를 offline_replay로 기록한다.

확인된 한계와 개선 방향

현재 한계	향후 개선 제안 (미구현)
ID 존재와 의미적 근거의 구분	원인과 인용 로그의 관련성을 사람이 평가하는 항목을 늘리고 부적절한 조치 제안을 별도 검토
소형 모델의 반복·언어 혼입·막연한 제안	한국어 응답 지시, 중복·빈 문자열 검사, 모델별 동일 입력 비교
작은 평가 범위	장애 유형·불완전 로그·반례를 확대하고 기준 고정·최초 응답 보존 원칙 유지
프롬프트 인젝션 방어 없음	로그를 비신뢰 입력으로 구분하고 악성 지시가 섞인 로그로 방어 검증
120초 타임아웃, 자동 재시도 없음	재시도 정책·실패 원인 기록을 추가하고 최초 응답과 각 시도 이력 보존
사람 최종 판단은 안내 문구	조치 전 검토·승인 기록을 추가하되 실효성을 별도로 검증

현재 스키마 검사는 배열 원소가 문자열인지만 확인하므로 next_steps 안의 빈 문자열은 형식상 통과할 수 있다. 로그 내용의 모순이나 조치의 안전성도 자동으로 판단하지 못한다. [S1, S2]

재현 범위

저장소 설치 안내에 따라 Python 의존성과 Ollama·모델을 준비한다. 오프라인 검사는 python -m pytest tests/ -q, 실제 호출은 python run.py <입력 JSON>으로 실행한다. 모델 재실행 시 동일 응답이 보장되지는 않는다. 웹 UI·DB·자동 조치·실제 CCTV 연동은 이 프로젝트의 구현 증거에 포함되지 않는다. [S1, S2]

06 출처와 검증 범위

모든 저장소 파일 링크는 아래 기준 커밋에 고정했다. PDF의 [S번호]는 이 목록을 가리킨다. 링크를 클릭하면 코드·원본 실행 결과·평가 문서를 직접 확인할 수 있다.

기준 커밋: 5e77e4d7108611ef69650896896a8a850495df00

출처	검증 대상 · 클릭 가능한 원문
S1	프로젝트 설명·역할·한계 README.md
S2	CLI·모델 호출·검증 구현 incident_copilot/cli.py 모델 호출 / 검증 구현 / 입력 모델
S3	order-api 원본 실행 JSON evidence/result_incident_1_normal.json
S4	알림 장애 원본 실행 JSON (구버전) evidence/result_incident_2_tricky.json
S5	CCTV 조치 전 원본 실행 JSON evidence/result_adapter_before.json
S6	CCTV 조치 후 원본 실행 JSON evidence/result_adapter_after.json
S7	증거 사본·구버전 스키마 설명 evidence/README.md
S8	사전 평가 기준·재시도 원칙 docs/eval_criteria_adapter_case.md
S9	전후 평가 결과·미충족 근거 docs/eval_result_adapter_case.md
S10	검증 로직 테스트 14개 tests/test_validator.py CLI 테스트 / 클라이언트 테스트 / CI 설정
S11	과거 CI 실행 및 원본 로그 실행 36123300366 / Python 3.11 / Python 3.12
S12	사람이 작성한 오프라인 데모 data/demo_manual_response_missing_id.txt

증빙 해석 시 유의점

공개 파일은 프로젝트가 보존한 실행 기록이며 독립적인 실환경 인증 자료는 아니다. 가상 입력의 복구 기록은 실제 CCTV 복구 실적으로 해석하지 않는다. 이번 검수는 원본 파일 대조·CI 로그 확인과 PDF 한글·배치 확인이며, 모델 재호출·현장 실험·시간 절감 측정은 수행하지 않았다.

참조 원고: 대화 "공모전 신청서 작성"의 첨부 화면 일부. 전체 본문을 조회할 수 없어 정량 주장이나 새 사실의 단독 출처로 사용하지 않았다.