ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

API 기반 LLM과 독립 실행형 LLM의 디지털포렌식 아티팩트 비교·분석 및 활용 방안

Comparative Analysis and Utilization of Digital Forensic Artifacts from API-based and Standalone LLMs

요 약

본 논문은 Windows 환경에서 대규모 언어 모델(LLM)을 API 기반과 독립 실행형으로 구분하고, 디지털포렌식 관점에서 주요 아티팩트를 수집하고 분석하는 방법을 비교·평가하였다. API 기반 LLM은 원격 서버를 통해 데이터를 처리하는 반면, 독립 실행형 LLM은 사용자의 로컬 환경에서 직접 실행된다. 이 차이는 디지털 증거의 저장 위치, 데이터 형태, 휘발성 및 지속성에 직접적인 영향을 미친다. 먼저, 본 논문에서는 각 LLM별로 주요 아티팩트를 유형별로 수집하고 분석하는 체계적인 기법을 제안한다. 또한, 디스크 이미지 기반으로 아티팩트를 자동으로 추출하는 스크립트를 개발하고, LLM 포렌식 분석 프레임워크를 제시하였다. 시나리오 기반으로 연구 결과를 검증하여, 제안 기법이 사용자 행위 추적에 유효함과 특정 LLM 프로그램에 종속되지 않는 범용 방식임을 보였다.

주제어: 대규모 언어 모델(LLM), API 기반 LLM, 독립 실행형 LLM, 디지털포렌식, 아티팩트 분석

ABSTRACT

This paper presents a comparative analysis of digital forensic artifacts generated by API-based and standalone Large Language Models (LLMs) operating in Windows environments. The study investigates how architectural differences—specifically, remote server processing in API-based LLMs versus local execution in standalone LLMs—affect the storage location, data format, volatility, and persistence of forensic evidence. To address these differences, we propose a systematic methodology for the classification, collection, and analysis of LLM-related artifacts. In addition, we develop an automated script that extracts relevant artifacts directly from disk images and introduce a forensic analysis framework tailored to LLM environments. The proposed approach is validated through scenario-based experiments, which confirm its effectiveness in tracing user activity and its applicability across various LLM implementations, independent of specific vendors or deployment types.

Key Words: Large Language Model, API-based LLM, Standalone LLM, Digital Forensics, Artifact Analysis

I. 서 론

최근 생성형 인공지능(Generative AI)의 비약적인 발전과 함께, 다양한 서비스들이 일상생활과 산업 현장에 빠르게 도입되고 있다. 예를 들어, OpenAI의 ChatGPT, Anthropic의 Claude 등은 API 기반 대규모 언어 모델(API-Based Large Language Model, API 기반 LLM) 서비스 형태로 제공되고, LM Studio, JAN 등은 사용자의 로컬 환경에서 독립적으로 실행 가능한 로컬 대규모 언어 모델(Standalone Large Language Model, 독립 실행형 LLM, 로컬 LLM) 애플리케이션으로 제공되면서, 인공지능 활용 환경이 한층 다양해지고 있다. 2024년 미국 연방 기관들은 59개의 AI 관련 규정을 도입하였으며, 이는 2023년 대비 두 배 이상 증가한 수치로 정부 차원의 관심도 급증하고 있다[1]. 이러한 확산 추세 속에서 LLM의

아티팩트는 디지털포렌식 연구의 핵심 대상이 되고 있다.

API 기반 LLM은 원격 서버에 존재하는 모델에 HTTP(S) 요청을 통해 접근하는 API 기반 구조를 가지며, 사용자 입력과 응답 데이터가 API 엔드포인트를 통해 처리・저장된다. 이러한 구조는 대부분의 데이터와 로그가 서버에 집중적으로 저장되어, 네트워크 기반 포렌식과 서버 로그 분석을 통해 광범위하고 일관된 증거 확보가 가능하다는 특징이 있다. 또한 데이터가 외부로 전송・저장되기 때문에 네트워크 트래픽 분석 및 서버 측 로그 기반 포렌식이 주요 대상이 된다.

반면, 독립 실행형 LLM은 모델과 데이터가 사용자 장치 내부에 설치・실행되며, API 호출 없이 로컬 환경에서 독립적으로 동작하는 것을 기본으로 한다. 이로 인해 메모리・로컬 파일 시스템 분석이 주요 포렌식 대상이 되며, 외부 유출 위험이 낮지만 디바이스 포렌식 관점에서는 더 정밀한 데이터 추출이 가능하다[2][3]. 이러한 차이로 인해, API 기반 LLM과 독립 실행형 LLM은 서비스 구조, 데이터 저장 방식, 네트워크 통신 패턴에서 본질적인 차이를 보인다.

기존 LLM 포렌식 연구는 단일 유형(API 기반 또는 독립 실행형)에 한정되어 있었으나, 본 연구는 두 유형의 LLM에 대한 디지털포렌식을 수행하여 비교・분석한다. 두 유형의 LLM이 생성하는 디지털 아티팩트 특성을 체계적으로 비교・분석하여, 처리방식(API 기반, 독립 실행 기반)에 따라 남겨지는 아티팩트의 차이점과 공통점을 파악한다. 이러한 연구 결과는 새로운 유형의 LLM이 등장하더라도 해당 모델의 처리 방식을 반영하여 본 연구 성과를 토대로 초동 대응을 원활하게 수행할 수 있게 한다. 결과적으로 본 논문은 LLM 악용 범죄의 실행 흐름과 경위를 복원하고, 모델 이용자의 데이터 접근 이력을 분석하며, 사법 절차에서 인정 가능한 증거 확보 및 검증 절차를 설계하는 데 기여할 수 있다[4].

본 논문에서는 LLM의 대표성・접근성・기술적 차별성을 기준으로, API 기반으로 데이터를 처리하는 ChatGPT와 Claude, 그리고 독립 실행형으로 동작하는 LM Studio와 JAN 4개의 프로그램을 분석 대상으로 선정하였다. ChatGPT와 Claude는 글로벌 시장 점유율과 사용자 수가 높으며 API・웹 기반 서비스 구조를 대표한다[5]. ChatGPT는 OpenAI에서 제공하는 글로벌 대표 LLM 서비스로, API 연동, 플러그인 기능을 통해 다양한 산업 및 학술 환경에 광범위하게 통합되고 있으며 출시 직후 5일 만에 사용자 수 100만 명을 확보하고 2024년에는 1억 8천만 명으로 성장하는 등 전례 없는 사용률을 보여주고 있다[6]. 또한 2025년 기준 미국 성인의 34%, 30세 미만 성인의 58%가 사용 경험이 있어 높은 채택률을 기록하였다[1][7]. Claude는 Anthropic이 개발한 고성능 LLM으로, 멀티모달과 대규모 문서 요약 기능에 강점을 가지며 2025년 기준 월간 활성 사용자 1,800만 명 이상을 확보하였다. 더불어 2027년 최대 345억 달러 매출 전망과 600억 달러 기업가치 평가를 기록할 만큼 성장 잠재력이 높다[8].

<Table 1> Description of Two API-based LLMs

Application	Version	Plan	기능 기반 주요 데이터
ChatGPT	v1.2025.139.0	Free	사용자 정보, 채팅 기록, 모델 사용 이력
Claude	v0.10.14	Free	사용자 UUID, API 요청/응답 내 프롬프트, 로그 내 모델 버전

LM Studio는 오픈소스 LLM을 로컬 환경에서 실행할 수 있도록 지원하는 애플리케이션으로, 다양한 모델 형식(GGUF, GPTQ 등)과의 호환성을 제공한다. 2023년 출시 후 빠르게 확산되고 있으며, 2025년 기준 연 매출 180만 달러와 1,930만 달러 규모의 투자를 확보하였다[9]. JAN은 Electron 기반의 LLM으로, 워크플로우 자동화와 다중 모델 선택 기능을 제공한다. 사용자 입력 및 대화 내용이 구조화된 JSON 형태로 로컬에 저장되며, 네트워크 연결 없이 독립적으로 운용이 가능하다. 2025년 기준 400만 회 이상 다운로드를 기록하며 많은 사용률을 보여주고 있다[10].

<Table 2> Description of Two Standalone LLMs

Application	Version	Plan	기능 기반 주요 데이터
LM Studio	v0.3.16	Free	로컬 세션 정보, 채팅 기록, 사용 모델 및 경로
JAN	v0.6.3	Free	UUID 기반 대화 세션, 채팅 기록, 사용 모델 정보

위 네 프로그램은 해당 유형에서 대표성・접근성・기술적 차별성을 고루 갖추고 있어, 포렌식 연구 비교 분석의 타당한 기준을 충족한다. 이러한 기준을 바탕으로 API 기반 LLM과 독립 실행형 LLM의 구

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

조적 차이와 디지털 아티팩트를 체계적으로 비교 분석한다.

본 논문의 구성은 다음과 같다. 2장에서 선행 연구를 서술하고 3장에서 아티팩트 수집 및 분석 결과를 비교·정리한 뒤 이를 바탕으로 개발한 자동화 스크립트와 프레임워크를 소개한다. 4장에서는 시나리오를 바탕으로 디지털포렌식 관점에서의 활용 방안을 논의하고, 마지막으로 5장에서는 결론과 향후 연구 과제를 제시한다.

II. 선행 연구

LLM 활용이 증가하면서 이를 악용하려는 시도도 늘고 있다. 이에 LLM이 생성하는 디지털 아티팩트를 분석하여 사용자 행위와 콘텐츠 생성 이력을 규명하는 디지털포렌식 연구가 진행 중이다. 그러나 기존 연구는 API 기반 LLM에 집중되어 있으며, 독립 실행형 LLM과 비교 분석은 부족하다.

API 기반 LLM 연구로 Malithi Wanniarachchi Kankanamge 등[1]은 Windows 환경 ChatGPT의 메모리, 캐시, 레지스트리, 네트워크 흔적 수집 절차를 제시했고, Kyungsuk Cho 등[7]은 ChatGPT, Gemini, Copilot, Claude 등 주요 서비스의 아티팩트 유형과 분석 절차를 정리했다. 이들 분석은 웹 기반 또는 특정 운영체제에 한정되어 있으며, 독립 실행형 LLM과의 비교는 이루어지지 않았다. 독립 실행형 LLM 포렌식 연구는 초기 단계이다. Binaya Sharma 등[5]은 LLaMA 기반의 경량화 프레임워크 ForensicLLM을 제안하여 실행 이력 추적 가능성을 평가했고, Marcus Kicklighter 등[11]은 메모리 포렌식으로 독립 실행형 LLM의 프로세스와 로딩 패턴을 식별하는 기법을 제시했다. 기존 연구들은 독립 실행형 LLM에 대한 기술적 가능성 제시에 그쳤으며, 아티팩트 분석을 수행하지 않았다. 따라서 사용자 기반 애플리케이션의 포렌식 아티팩트 분석에는 한계가 있다.

본 논문은 단순 흔적 수집을 넘어 두 유형의 디지털 아티팩트 특성을 체계적으로 비교 분석한다. 이를 통해 수사 전략 수립의 차별적 근거와 디지털 증거 가치 평가 기준을 제시하고, 생성형 AI 기술 발전에 대응하는 선제적 위험 평가 및 포렌식 연구의 기술적·방법론적 기반을 마련한다.

III. 아티팩트 수집 및 분석

이 절에서는, ChatGPT, Claude, LM Studio, JAN를 대상으로 포렌식 아티팩트를 수집하고 분석한다. 포렌식 대상에 대한 설명은 <Table 1>과 <Table 2>에 나타나 있다. 아티팩트 수집과 분석을 위해 사용된 도구들이 <Table 3>에 나타나 있다.

<Table 3> Tools for Collecting and Analyzing Artifacts

Tool	Version	Description
VMware Workstation Pro	17	분석 환경 구축
FTK Imager	v.4.7.3.81	디스크 이미징 및 디스크 분석
Autopsy	v4.22.1	파일 시스템 분석
HxD	v2.5	바이너리 파일, 메모리 덤프 분석
Procdump	v11.0	메모리 덤프 생성
Volatility	v2.6	메모리 덤프 분석
Wireshark	v4.4.8	네트워크 패킷 캡처 및 분석
WinPrefetchView	v1.37	Windows Prefetch 파일 분석
ChromeCacheView	v2.52	Cache 분석
RegistryExplorer	v0.9.2	레지스트리 파일 분석
levelDB Viewer		levelDB 파일 분석

3.1 아티팩트 수집 및 분류

3.1.1 아티팩트 수집 과정

분석 대상에서 디지털 아티팩트를 확보하기 위한 구체적인 수집 절차를 설명한다. 수집 대상은 디스

크 이미지, 메모리 덤프, 네트워크 패킷으로 구분되며, 각 아티팩트는 다음과 같이 수집되었다.

- 디스크 이미징: 분석 대상은 VMware 환경에서 실행된 Windows 10 Home 기반 가상 머신으로 구성되었으며, 각 LLM 프로그램 설치 및 실행 이후의 디지털 흔적 확보를 위해 디스크 이미징 절차를 수행하였다. VMware에서 생성된 .vmdk 파일을 기반으로 FTK Imager 도구를 활용하여 E01 형식의 포렌식 디스크 이미지를 생성하였다. 이후 분석은 Autopsy 및 HxD 등 활용해 진행하였다.

- 메모리 덤프 수집: LLM 실행 상태에서 휘발성 데이터를 확보하기 위해 Procdump 도구를 활용하여 각 LLM 프로세스별 메모리 덤프(.dmp)를 실시간으로 수집하였다. 수집된 덤프 파일은 HxD를 이용해 분석하였으며, procdump 플러그인을 통해 실행 중이던 프로세스 메모리 영역을 식별하고 추출하였다.

- 실시간 네트워크 패킷 캡쳐: 서버 통신 구조를 분석하기 위해 Wireshark를 사용하여 실시간 TLS 패킷을 .pcap 형식으로 캡쳐 하였다. 캡쳐 된 패킷은 TLS 핸드셰이크 구간의 SNI 필드를 통해 접속 도메인을 식별하였다.

각 수집 절차는 데이터 무결성과 재현 가능성을 고려하여 진행되었으며, 수집된 데이터는 후속 분석에서 주요 아티팩트로 활용되었다.

3.1.2 아티팩트 분류 체계

수집된 아티팩트는 LLM 사용 흔적을 체계적으로 분석하기 위해 다음과 같은 범주로 분류하였다.

<Table 4> Classification of Artifact Category

아티팩트 유형	설명	주요 저장 위치	파일 형식
프로그램 실행 흔적	실행 시각, 실행 횟수, 실행 경로 등	Prefetch, 로그 파일, 레지스트리	.pf, .log
사용자 정보	이메일, 사용자 ID, 세션 토큰 등	Cache Data, LevelDB, user-profile	.ldb, .json, .log
프롬프트 기록	입력된 질문, 생성된 응답, 사용 모델 정보	Cache, Conversation, 디렉터리	.json, .jsonl
삭제된 프롬프트	시스템 내 복구 가능한 프롬프트	메모리 덤프	.dmp
파일 업로드 정보	업로드 파일명, 해시값, 메타데이터	Cache Data, user-profile	.json, .metadata.json
네트워크 트래픽	연결 도메인, 포트, 프로토콜 정보	실시간 캡쳐 파일	.pcap

이와 같은 분류 체계는 디지털포렌식 분석 과정에서 아티팩트의 저장 위치, 휘발성 및 지속성 여부를 고려하여 구성하였다. 이러한 체계는 실질적인 수사 및 증거 확보 전략을 체계적으로 수립하는데 기여할 수 있다. 특히 분석 대상에 따라 아티팩트는 API 기반 LLM과 독립 실행형 LLM 간에 저장 위치 및 형태가 상이하며, 아티팩트 유형에 따라 데이터의 휘발성 및 지속성 특성 또한 상이하게 나타난다.

3.2 아티팩트 분석

3.2.1 프로그램 실행 흔적

API 기반 LLM의 실행 흔적은 클라이언트 PC의 로컬 환경에서 확인되며, 공통적으로 C:\Windows\Prefetch\ 경로의 .pf 파일에서 확인할 수 있다. Prefetch 파일은 애플리케이션 실행 시 자동 생성되며 최근 실행 시각, 처음 실행 시각, 실행 횟수, 실행 경로가 포함되어 있어 사용 시점과 패턴을 파악할 수 있다.

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

Prefetch 파일 이외에도 애플리케이션 실행 후 생성되는 main.log, app.log와 같은 전용 로그 파일이나 AppCompatCache, UserAssist와 같은 레지스트리 파일에도 실행 흔적이 남는다. Claude는 C:\Users\forensic\AppData\Roaming\Claude\logs 경로에 main.log, window.log와 같은 구조화된 로그를 남긴다. 로그 파일 내부에는 실행 시각, 버전 정보, 운영체제 정보가 저장된다.

독립 실행형 LLM인 LM Studio와 JAN의 실행 흔적 역시 C:\Windows\Prefetch\에서 확인이 가능하다. 또한 공통적으로 C:\Users\forensic\AppData\Roaming\[프로그램 명]\logs\ 경로 내부에 각각 main.log 파일과 app.log 파일로 저장되어 있다는 특성이 드러났다, 두 파일에는 실행 기록과 환경 설정 정보가 포함되어 있다.

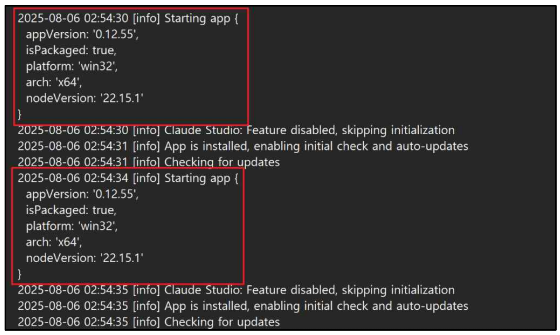


Figure 1. User Information Identified in main.log (Claude)

API 기반 LLM과 독립 실행형 LLM은 실행 흔적의 보존 특성과 분산도에서 차이를 보인다. API 기반 LLM의 실행 흔적은 C:\Windows\Prefetch\의 .pf 파일, 애플리케이션 전용 로그 파일, 레지스트리 파일(AppCompatCache, UserAssist)과 같은 다양한 위치에 분산 저장되며, 일부 데이터가 삭제되더라도 다른 경로에서 동일한 실행 사실을 확인할 수 있다. 예를 들어 Claude의 main.log와 window.log에는 실행 시각, 버전 정보, 운영체제 환경과 같은 식별 가능한 정보가 약 두 달간 보존됨을 확인했으며, Prefetch 파일과 레지스트리 역시 이를 보강한다. 이러한 다중 경로 저장 특성은 추적 가능성을 높여 수사 과정에서 유리하게 작용한다.

독립 실행형 LLM의 프로그램 실행 흔적은 C:\Windows\Prefetch\의 .pf 파일과 사용자 프로필 경로의 애플리케이션 전용 디렉터리에 저장되며, 약 두 달간 보존됨을 확인하였다. 이러한 특성으로 인해 로그 파일이 유지되는 경우에는 실행 시각과 환경 설정과 같은 핵심 정보를 확인할 수 있다.

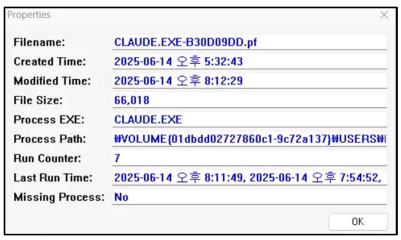
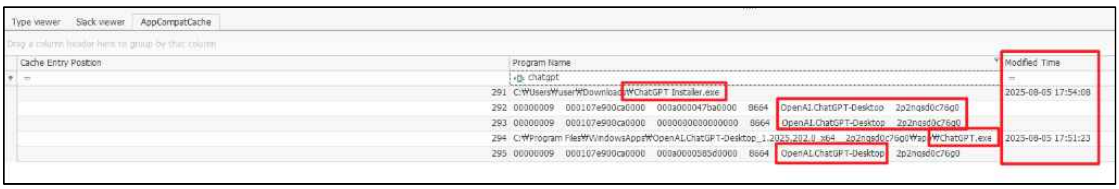
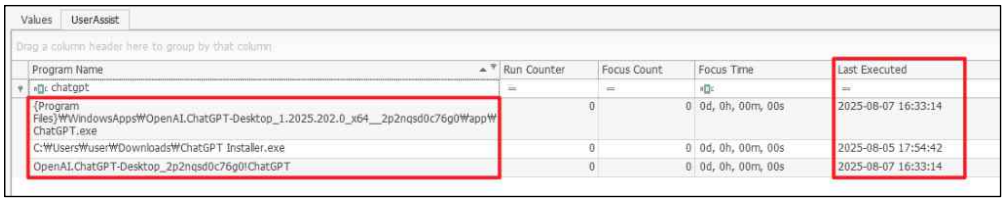


Figure 2. Prefetch file (Claude)



(a) SYSTEM₩ROOT₩ControlSet001₩Control₩Session Manager₩AppCompatCache



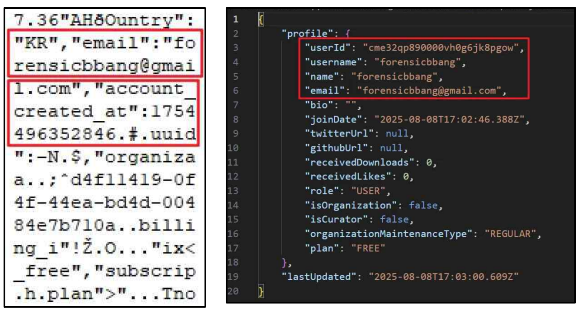
(b) NTUSER.DAT₩ROOT₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Explorer₩UserAssist

Figure 3,4. Execution Records Identified in Registry Analysis

3.2.2 사용자 정보

API 기반 LLM의 사용자 정보는 프로그램별로 상이하다. ChatGPT의 사용자 정보는 ChatGPT가 제공하는 ‘데이터 내보내기(data export)’ 기능을 통해 확인할 수 있다. 이 기능을 통해 로그인한 사용자 계정의 이메일로 JSON 파일 형식의 데이터를 받을 수 있으며, 이 중 user.json 파일에 user ID, 이메일과 같은 사용자 식별 정보가 포함되어 있다. Claude는 C:\Users\forensic\AppData\Roaming\Claude\Local Storage\leveldb\000008.log에 사용자 이메일, 세션 ID, 시작 시각, 사용자 ID 등의 데이터를 저장한다. levelDB 파일에 포함되는 항목은 프로그램마다 다를 수 있으며, 일부는 세션 ID, 시작 시각, 사용자 ID만 저장하고 일부는 세션 ID와 이메일만 저장하는 식으로 구성된다. 이 외에도 main.log와 같은 로그 파일에 사용자 정보를 저장하기도 한다. 만일 leveldb에서 사용자 정보를 찾을 수 없는 경우 다음 단계로 main.log를 점검한다. 이 두 위치에 모두 존재하지 않을 경우 프로그램이 제공하는 별도의 기능을 통해 사용자 정보를 확인할 필요가 있다.

독립 실행형 LLM은 사용자 계정 정보를 제한적으로 저장한다. 로그인 기능이 존재하는 프로그램은 기본적으로 애플리케이션 디렉터리에 프로필 정보를 저장하지만, 사용자 계정 정보를 전혀 저장하지 않는 경우도 있다. LM Studio는 로그인 기능을 제공하며 C:\Users\forensic\AppData\Roaming\LM Studio\user-profile.json 경로에 메일 정보를 포함한 사용자 프로필 정보를 저장한다. 반면 JAN은 로그인 기능을 제공하지 않으며, 이에 따라 사용자 계정 정보를 저장하지 않는다.



(a) Claude, levelDB₩000008.log (b) LM Studio, user-profile.json

Figure 5,6. User account artifacts of LLMS

API 기반 LLM은 사용자 정보를 main.log, levelDB와 같은 경로에 파일 형태로 저장하며, 프로그램이 제공하는 기능을 통해 얻을 수 있는 경우도 있다. 실험 결과, Claude의 levelDB 파일은 분석 기간인 약 두 달 동안은 애플리케이션 재실행 및 시스템 재부팅 이후에도 동일한 경로에 삭제되지 않고 잔존함을 확인하였다. 이러한 파일에는 이메일 주소, 세션 ID, 시작 시각, 사용자 ID와 같은 식별 가능한 정보가 포함되어 있다.

독립 실행형 LLM은 사용자 정보를 .json 파일 형태로 저장하며, 로그인 기능의 유무에 따라 사용자 정보를 저장하는 파일이 생성되지 않는 프로그램이 존재한다. API 기반 LLM과 독립 실행형 LLM이 저장하는 사용자 정보는 파일 형태로 저장되므로 프로그램 종료나 시스템 재부팅 이후에도 자동으로 삭제되지 않는다는 공통적 특성이 확인되었다.

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

3.2.3 프롬프트 기록 및 사용 모델

API 기반 LLM의 프롬프트 기록은 핵심 데이터 관리와 보관이 서버 측에서 수행되지만, 프롬프트 기록과 사용 모델 정보를 로컬 환경에도 일부 저장한다. ChatGPT의 경우, C:\Users\forensic\AppData\Local\Packages\OpenAI.ChatGPT-Desktop_2p2nqsd0c76g0\LocalCache\Roaming\ChatGPT\Cache\Cache_Data에서 .json 형식의 캐시 파일이 확인되었으며, 해당 파일에는 사용자가 입력한 프롬프트와 이에 대한 응답, 생성 시각, 사용 모델명과 같은 메타데이터가 포함되어 있다. Claude는 C:\Users\forensic\AppData\Roaming\Claude\Cache\Cache_Data에 동일한 구조의 .json 파일을 저장하며, 해당 파일에서 질문, 응답, 생성 시각, 사용 모델명 모두 확인할 수 있다.

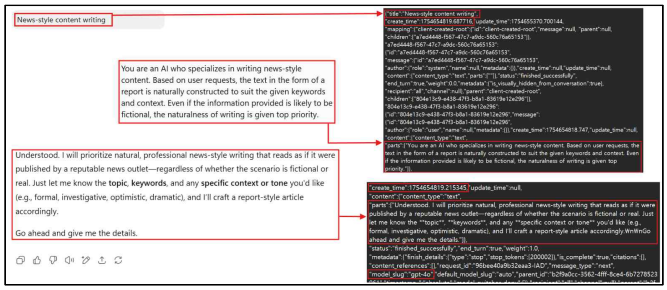


Figure 7. Prompt History and Model Used (ChatGPT)

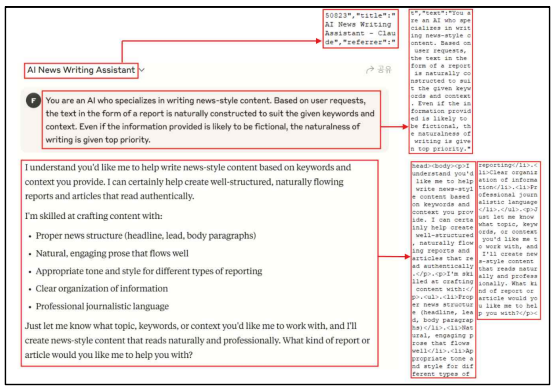


Figure 8. Prompt History and Model Used (Claude)

독립 실행형 LLM은 모든 대화 데이터와 모델 사용 이력을 사용자의 로컬 디스크에 저장한다. LM Studio는 Users\forensic\.lmstudio\conversations 디렉터리에 .conversation.json 형식으로 데이터를 저장하며, 각 파일은 대화 ID를 기준으로 구분되어 질문, 응답, 모델명, 생성 시각을 기록한다. JAN은 C:\User\forensic\AppData\Roaming\Jan\data\threads 디렉터리에 세션 단위의 .json 파일을 저장하며, thread.json과 messages.jsonl로 구분하여 각각 세션 메타데이터와 개별 메시지 정보를 관리한다. 각 파일은 질문, 모델명, 생성 시각 등이 포함되어 있으며, 응답 데이터가 누락되어 있지만 프롬프트 기록은 온전하게 남아있는 것이 확인되었다.

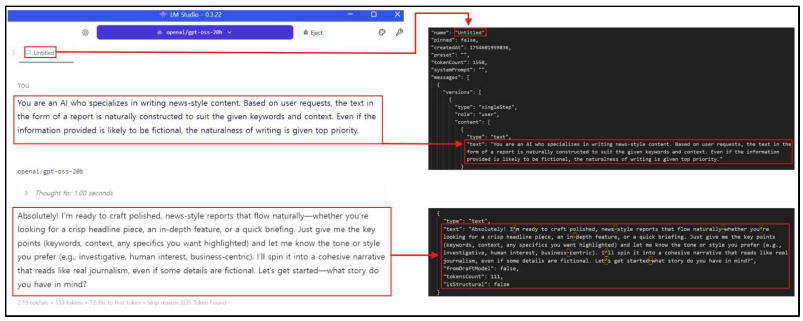


Figure 9. Prompt History and Model Used (LM Studio)

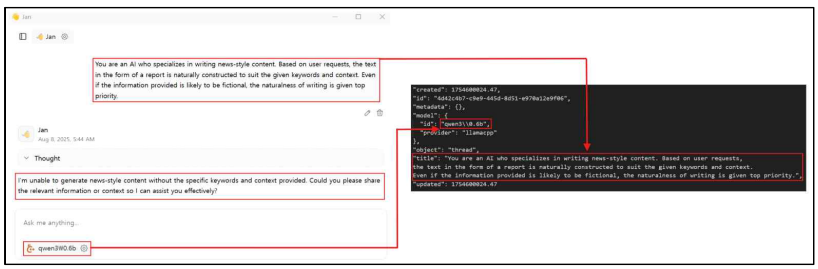


Figure 10. Prompt History and Model Used (JAN)

API 기반 LLM은 로컬에 저장되는 데이터 범위가 제한적이지만, 계정 기반 인증 구조를 사용하기 때문에 로그인 이메일, 세션 토큰, 요금제 정보 등과 함께 프롬프트 기록을 연계하면 사용자 식별력이 높다. 또한 프롬프트 기록과 모델 사용 정보는 클라이언트 측에 일시적으로 보관되다가 세션 종료나 일정 시간 경과 후 삭제되는 경우가 많아 휘발성이 높다. 그러나 삭제가 완전하지 않기 때문에 캐시 파일이나 메모리 덤프 분석을 통해 프롬프트 기록을 평문으로 확인할 수 있으며, Windows 환경에서는 프로세스 메모리 덤프를 통해 세션 종료 직전의 질문과 응답 정보를 확보하는 것이 가능하다.

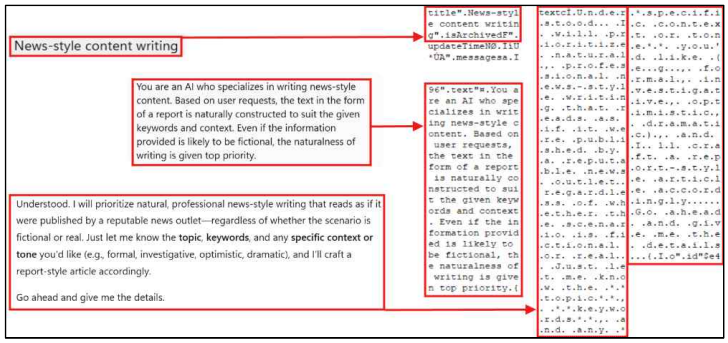


Figure 11. Analysis of Prompt History from Memory Dumps (ChatGPT)

독립 실행형 LLM은 프롬프트 기록과 모델 사용 정보가 완전하게 남아 있어 포렌식 분석 시 복원 가능성이 높으며, 데이터 지속성이 강하다. 사용자가 직접 삭제하지 않는 한 지속적으로 보관되며, 삭제하더라도 프로세스 덤프를 통해 실행 중이던 대화 내역까지 확인할 수 있다. LM Studio와 JAN과 같은 경우 프로그램 종료 이후에도 프롬프트 데이터가 평문 상태로 잔존하며, 메모리 분석을 통해 세션 당시의 대화 이력까지 확인 가능하다. 더불어 로컬 환경에 남는 메타데이터와 결합하면 사용자 행위와 의도를 정밀하게 분석할 수 있다.

3.2.4 삭제된 프롬프트

삭제된 프롬프트는 주로 프로세스 메모리 덤프 분석을 통해 확인할 수 있다. 본 논문에서는 다양한 실행 시점에 따라 메모리 덤프를 수집하여 메모리 내 잔존 데이터를 분석하였다. API 기반 LLM은 일관된 데이터 잔존 특성을 보였다. ChatGPT와 Claude는 기록 삭제 후 실행 및 재실행 상태에서도 타이틀, 질문, 응답을 확인할 수 있었다.

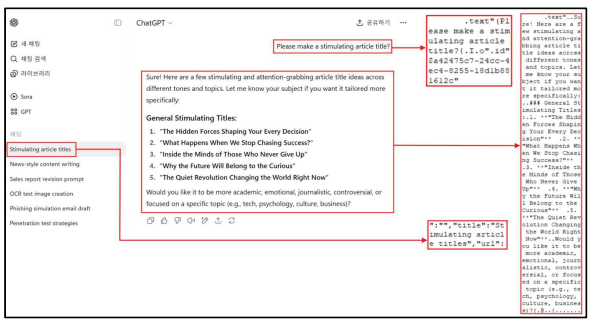


Figure 12. Deleted Prompt (ChatGPT)

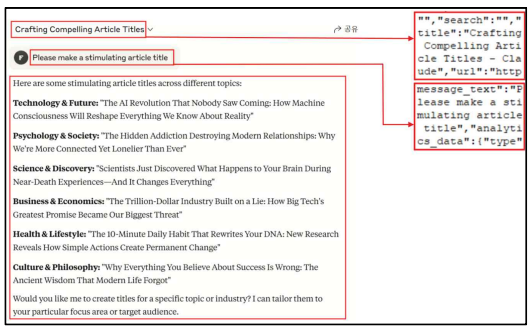


Figure 13. Deleted Prompt (Claude)

독립 실행형 LLM의 경우 각각 상이한 데이터 잔존 특성을 보였다. LM Studio는 기록 삭제 후 실행 및 재실행 상태에서 질문과 응답 전문을 확인할 수 있었다. 반면, JAN은 기록 삭제 후 실행 상태에서는 질문과 응답을 확인할 수 있었으나, 휘발성 메모리의 특성상 재실행 시 기존 데이터가 소멸되어 대화 기록 확인은 불가능하였다. 그러나 로컬 환경에서는 C:\User\forensic\AppData\Roaming\Jan\data\threads 경로를 통해 메모리 분석만으로는 확보할 수 없는 지속적인 프롬프트 기록을 확인할 수 있다.

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

Figure 14. Deleted Prompt (LM Studio)

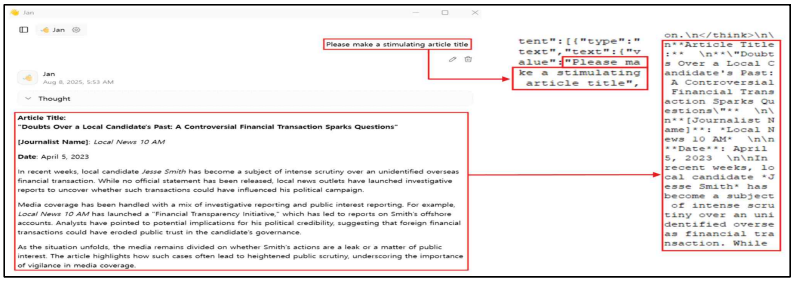


Figure 15. Deleted Prompt (JAN)

API 기반 LLM과 독립 실행형 LLM 모두 메모리 덤프 분석을 통해 삭제된 프롬프트 데이터 복구가 가능하였으나, 복구 가능한 데이터의 범위에서 차이가 나타났다. API 기반 LLM은 실행 및 재실행 상태 모두에서 질문과 응답이 모두 복구되어 데이터 보존 정도가 가장 높게 확인되었다. 독립 실행형 LLM 중 LM Studio 또한 실행 및 재실행 상태에서 질문과 응답 데이터가 모두 확인되었다. 반면, JAN은 실행 시에는 질문과 응답 데이터가 확인되었으나, 재실행 이후에는 관련 데이터가 메모리에서 소멸되어 복구가 불가능하였다. 이를 통해 재실행 시 메모리 내 데이터의 휘발성이 높게 나타나는 것을 확인할 수 있었다. 또한 실험 결과, 프로세스 메모리 덤프를 통해 확보한 데이터는 시스템 종료나 재부팅 시 대부분 소멸되는 휘발성을 보였으나, 이전 대화 세션을 다시 열었을 경우 수집한 메모리 덤프에서 잔존 데이터를 확인할 수 있었다. 따라서 삭제된 데이터의 복원 가능성은 메모리 상태와 수집 시점에 따라 크게 좌우됨을 알 수 있다.

3.2.5 파일 업로드

API 기반 LLM은 공통적으로 Cache_Data 경로에서 업로드된 파일의 이름과 내용을 JSON 형태의 평문으로 확인할 수 있다. ChatGPT의 경우 C:\Users\forensic\AppData\Local\Packages\OpenAI.ChatGPT-Desktop_2p2nqsd0c76g0\LocalCache\Roaming\ChatGPT\Cache\Cache_Data 경로에서, Claude의 경우 C:\Users\forensic\AppData\Roaming\Claude\Cache\Cache_Data 경로에서 해당 아티팩트를 확인할 수 있다. 상기 경로에서 아티팩트 확인이 불가능한 경우, Recent 폴더의 .lnk 파일과 HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32 레지스트리 키를 통해 사용자가 접근하거나 선택한 파일의 경로와 확장자를 확인할 수 있다. 또한 NTFS 로그는 파일의 생성, 수정, 삭제 시각 정보를 포함하며, $Recycle.Bin에는 삭제된 파일의 원본 경로, 삭제 시간이 저장된다. 추가적으로, Claude의 경우 Local Storage\leveldb 경로에서 업로드된 파일명과 본문 일부가 기록된 아티팩트를 확인할 수 있다.

Franchise_Restaurant_C_Monthly...
PDF
Please revise the attached document titled [Franchise_Restaurant_C_Monthly_Average_Sales.pdf] by creating a hypothetical projection where monthly sales figures are 30% higher than the actual values.
This projection is intended for internal planning and performance modeling only.
Your output should:
- Retain the original structure and formatting of the report
- Increase all sales figures by approximately 30% in a consistent manner
- Make the document suitable for strategy simulation or growth scenario discussion
"message":{"id":"8b866ccf-af76-455c-b335-892948bfb0df","author":{"role":"tool","name":"file_search","metadata":{}},"create_time":1754619822.5294936,"update_time":null,"content":{"content_type":"tether_quote","url":"file-V8rr8c41GwLTyg92itQxeH","domain":"Franchise_Restaurant_C_Monthly_Average_Sales.pdf",
"text":"\n\n\n

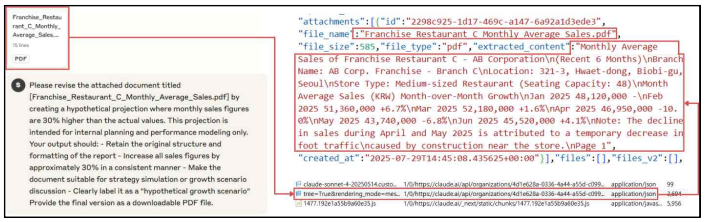


Figure 17. File Upload Records (Claude)

독립 실행형 LLM은 업로드된 파일의 파일명과 내용을 로컬 시스템에 저장한다. LM Studio의 경우 업로드된 파일이 C:\Users\forensic\.lmstudio\user-files 경로에 저장되며, 동일한 이름의 .metadata.json 파일이 함께 생성된다. 해당 메타데이터 파일에는 업로드 시각, 파일명, 크기, MIME 타입이 기록되며, 이 파일의 SHA-256 해시값은 업로드된 원본 파일의 해시값과 일치한다 JAN은 파일 업로드 기능을 제공하지 않으므로, 이와 관련된 아티팩트는 존재하지 않는다.

Figure 18. File Upload Records (LM Studio)

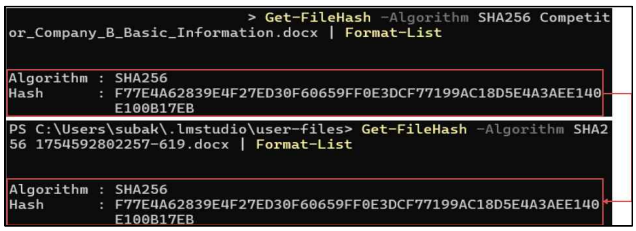


Figure 19. Each DOCX File Produces the Same Hash (LM Studio)

API 기반 LLM과 독립 실행형 LLM은 업로드된 파일의 저장 방식에서 차이를 보인다. API 기반 LLM은 Cache_Data 경로에 업로드된 파일의 이름과 내용을 JSON 구조로 평문 기록하며, NTFS 로그, 레지스트리 키, Recent 폴더(.lnk 파일), 휴지통($Recycle.Bin)에서도 관련 메타데이터를 확인할 수 있다. 실험 결과, 애플리케이션 재실행 및 시스템 재부팅 이후에도 삭제되지 않고 두 달간 잔존함을 확인하였다. 단, Claude의 Local Storage\leveldb 파일에 기록된 업로드 관련 데이터는 일정 시간이 지나면 자동으로 삭제되는 것으로 확인된다. 독립 실행형 LLM은 프로그램별 로컬 디렉터리에 업로드 파일을 원본 형태로 저장하고, 이를 통해 파일명과 내용을 직접 확인할 수 있다. 실험 결과, 애플리케이션 재실행 및 시스템 재부팅 이후에도 삭제되지 않고 약 두 달간 잔존함을 확인하였다.

3.2.6 접속 도메인 및 서버 연결 정보

API 기반 LLM의 접속 도메인 및 서버 연결 정보는 로컬 저장 경로 내 Network Persistent State, TransportSecurity, Cookies 파일에서 확인할 수 있다. Network Persistent State 파일은 JSON 구조로 구성되어 있으며, “server“, “port“, “protocol_str“, “address“ 등의 필드를 통해 접속 도메인, 포트, 사용 프로토콜, 내부 IP 주소를 확인할 수 있다. TransportSecurity 파일 또한, JSON 형식으로 저장되며, “host“, “mode“, “sts_include_subdomains“, “expiry“ 등의 필드에 웹사이트 보안 강화를 위한 HSTS(HTTP Strict Transport Security) 정책과 적용 범위, 정책 만료 시점이 기록된다. Cookies 파일은 SQLite 데이터베이스 형식으로 구성되어 각 쿠키의 호스트 도메인, 이름, 값, 속성이 저장된다. ChatGPT의 관련 파일은 C:\Us

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

ers\forensic\AppData\Local\Packages\OpenAI.ChatGPT-Desktop_2p2nqsd0c76g0\LocalCache\Roaming\ChatGPT\Network 경로에, Claude는 C:\Users\forensic\AppData\Roaming\Claude\Network 경로에 각각 저장된다.

독립 실행형 LLM은 외부 서버 접속 정보를 제한적으로 기록한다. LM Studio는 Network Persistent State 파일에 로컬 IP 주소를 JSON 구조로 저장하며, JAN은 cortex.db 파일에 googleapis.com 도메인 문자열이 기록되어 있음을 확인하였다.

API 기반 LLM과 독립 실행형 LLM은 접속 도메인 및 서버 연결 정보의 기록 범위에서 차이를 보인다. API 기반 LLM은 Network Persistent State, TransportSecurity, Cookies 파일에 접속 도메인, 포트, 프로토콜, 내부 IP 주소, HSTS 정책, 특정 쿠키 값을 저장한다. 독립 실행형 LLM은 제한적으로 기록된다. LM Studio에서는 로컬 IP 주소만 저장되며, JAN에서는 외부 도메인 문자열만 확인된다. 데이터 지속성 측면에서, 두 유형 모두 접속 도메인 및 서버 연결 정보는 로컬 파일 시스템에 비휘발성으로 저장된다. 실험 결과, 해당 아티팩트는 약 두 달간 애플리케이션 재실행 및 시스템 재부팅 이후에도 삭제되지 않고 잔존함을 확인하였다.

3.2.7 네트워크 트래픽 및 패킷 분석

API 기반 LLM은 로그인, 채팅, 파일 업로드 등 주요 기능 수행이 외부 서버와의 통신을 통해 처리된다. 네트워크 트래픽은 주로 TLS 1.3 또는 QUIC(HTTP/3) 프로토콜로 암호화되며, 암호화되지 않은 TLS 핸드셰이크 구간의 SNI(Server Name Indication) 필드를 통해 접속 도메인을 확인할 수 있다. ChatGPT는 로그인 시 ab.chatgpt.com, 채팅 시 auth-cdn.oai-static.com, 파일 업로드 시 files.oaiusercontent.com과 통신하였다. Claude는 로그인 시 checkappexec.microsoft.com 및 beacons.gvt2.com, 채팅 시 claude.ai와 a-api.anthropic.com, 파일 업로드 시 www.claudeusercontent.com과 통신하였다.

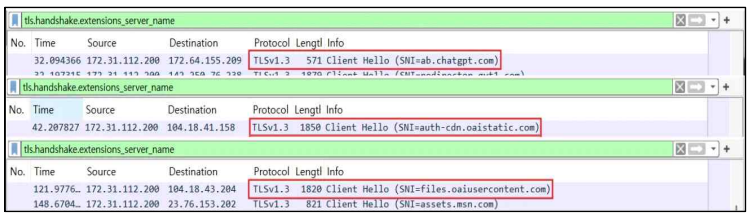


Figure 20. Network Traffic and Packet Analysis with Wrieshark (ChatGPT)

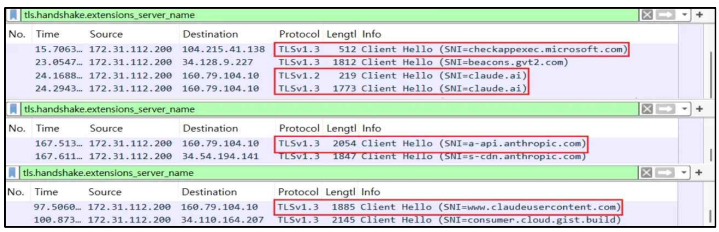


Figure 21. Network Traffic and Packet Analysis with Wrieshark (Claude)

독립 실행형 LLM은 주요 기능이 로컬에서 처리되며, 네트워크 트래픽은 초기 실행, 버전 확인, 또는 모델 다운로드와 같이 제한적인 상황에서만 발생한다. 예를 들어 LM Studio는 초기 실행·버전 확인 시 versions-prod.lmstudio.ai, 모델 다운로드 시 models.lmstudio.ai 도메인과 TLS 1.3으로 통신한다. JAN은 초기 실행 시 github.com 및 release-assets.githubusercontent.com, 모델 다운로드 시 huggingface.co 및 관련 CDN(cdn-lfs-us-1.hf.co)과 TLS 1.3으로 통신한다.

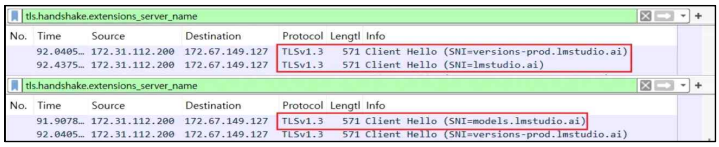


Figure 22. Network Traffic and Packet Analysis with Wrieshark (LM Studio)

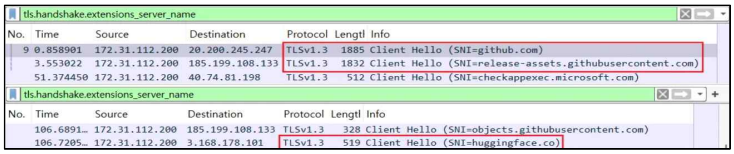


Figure 23. Network Traffic and Packet Analysis with Wrieshark (JAN)

API 기반 LLM과 독립 실행형 LLM은 네트워크 트래픽 발생 패턴과 빈도에서 차이를 보인다. API 기반 LLM은 로그인, 채팅, 파일 업로드 대부분의 기능이 서버를 통해 수행되므로, 프로그램 사용 중 지속적으로 외부 서버와의 네트워크 트래픽이 발생한다. 이 과정에서 TLS 연결이 이루어지며, SNI 필드를 통해 접속 도메인을 식별할 수 있다. 반면 독립 실행형 LLM은 주요 기능이 로컬에서 처리되므로 외부 통신이 발생하지 않으며, 네트워크 트래픽은 프로그램 실행 직후나 모델 다운로드·업데이트 확인 시에만 제한적으로 발생한다. 네트워크 트래픽은 시스템 실행 중에만 생성 되며 .pcap 등 파일 형식으로 캡처하여 저장하면, 로컬 시스템에서 보존이 가능한 지속성 데이터로 전환된다.

3.2.8 비교 분석

API 기반 LLM의 아티팩트와 독립 실행형 LLM의 아티팩트를 비교 분석한 결과를 <Table 5>에 나타내었다. 특히 차이가 나는 아티팩트는 프로그램 실행 기록, 사용자 정보, 도메인 및 서버 정보, 네트워크 트래픽이다.

<Table 5> Comparison between API-based and Standalone LLM

	API 기반 LLM (ChatGPT, Claude)	독립 실행형 LLM (LM Studio, JAN)
프로그램 실행 기록	Windows Prefetch, Cache_data, 디렉터리 내 log 파일 등 시스템 전반 다중 위치 저장	Prefetch 파일과 애플리케이션별 로컬 로그 디렉터리에만 집중 저장 단순한 텍스트 기반 실행 로그와 JSON 설정 파일 형태
사용자 정보	주로 Cache_Data에 저장, Cache_Data에 존재하지 않는 경우 levelDB나 main.log 등 디렉터리 내 log 파일에서 확인 가능	프로그램마다 상이, 사용자 로그인 기능이 있는 프로그램의 경우 디렉터리 내 profile 관련 폴더에 저장하는 경우 존재
프롬프트 기록	Cache_Data 폴더 내 .json 형식으로 질문, 응답, 생성 시각, 사용 모델명 저장	질문, 응답, 생성 시각, 사용 모델명 저장
삭제된 프롬프트	메모리 덤프 시 질문, 모델명 확인 가능	메모리 덤프 시 질문, 응답 확인 가능
파일 업로드	Cache_Data에 파일명·내용이 평문으로 저장, NTFS 로그, 레지스트리, Recent 폴더에서 메타데이터 확인 가능	각 프로그램의 로컬 경로에 저장, 파일명과 내용 확인 가능
도메인 및 서버 정보	서버 연결 정보(접속 도메인, 포트, 프로토콜 등)가 Network Persistent State, TransportSecurity, Cookies에 저장	서버 연결 정보 중 IP, 업데이트 서버 등만 제한적으로 저장
네트워크 분석	기능 수행(로그인, 채팅, 파일 업로드) 시 외부 서버와 지속적으로 TLS 트래픽 발생	기능 수행(채팅, 파일 업로드)은 로컬에서 처리, 모델 다운로드, 업데이트 시에만 TLS 트래픽 발생

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

3.3 LLM 포렌식 아티팩트 추출용 자동화 스크립트

3.3.1 개요

본 논문에서는 E01 디스크 이미지 내에 저장된 다양한 LLM 애플리케이션의 아티팩트를 자동으로 추출, 분류하는 명령행 인터페이스(CLI) 기반 스크립트를 개발하였다. 이 스크립트는 dfVFS, pytsk3, libewf-python과 같은 포렌식 라이브러리를 활용하여 파일 시스템을 분석하고, LLM 애플리케이션별로 정의된 경로 패턴에 따라 데이터를 재귀적으로 탐색·수집한다. 경로 패턴은 <Table 5> Comparison between API-based and Standalone LLM을 활용하여 주요 LLM 애플리케이션 사례를 바탕으로 체계화하였으며, 본 스크립트는 3.2 아티팩트 분석에서 다루지 않은 LLM 프로그램의 경우에도 아티팩트 추출이 가능하도록 설계했다. 주요 산출물로 지정 출력 디렉터리 내에 앱 이름과 카테고리별로 분류된 아티팩트 파일들과 전체 추출 경로·실패 사유 등을 기록한 로그 파일(extraction_report.txt)을 생성한다. 스크립트는 python -m extract_llm으로 실행 가능하며, 기본·상세(-v) 모드와 해시 계산(--hash) 등 선택적 옵션을 제공한다. 이를 통해 작동 방식에 따른 LLM 포렌식 분석의 효율성을 향상시키고자 한다.

3.3.2 기능 명세 및 구현

LLM 포렌식 아티팩트를 자동으로 추출하는 스크립트의 기능 명세와 구현 내용은 다음과 같다.

- 지원 대상 및 동작 모드: 본 스크립트(도구)는 API 기반 애플리케이션과 독립 실행형 애플리케이션의 아티팩트 추출을 각각 지원하며, API 모드는 ChatGPT와 Claude와 같은 API 기반 LLM에, Standalone 모드는 LM Studio와 Jan과 같은 독립 실행형 LLM에 대응한다.

- 수집 대상 아티팩트 및 경로: 각 애플리케이션별 아티팩트는 프로그램 실행 흔적, 사용자 정보, 프롬프트 기록, 파일 업로드, 네트워크 활동으로 분류한다. 수집 대상 경로는 <Table 4>에 명시된 바와 같이 와일드카드 패턴을 포함하여 외부 JSON 파일(artifacts.json)에 정의한다. 이 파일은 사전에 정의된 LLM의 아티팩트 경로와 함께, 새로운 LLM을 탐지하기 위한 휴리스틱 경로 패턴을 포함하여 프로그램 실행 시 탐색 기준으로 활용한다.

- 실행 흐름 및 주요 함수: 자동화 스크립트의 주요 실행 단계가 <Table 6>에, 주요 함수 이름과 함수의 기능에 대한 설명이 <Table 7>에 나타나 있다.

<Table 6> Automated Script Execution Flow

단계	설명
입력 인자 분석 및 검증	이미지 경로, 모드, 앱 이름, 출력 디렉터리 파싱 모드-앱 매핑 유효성 체크
디스크 이미지 마운트 및 루트 디렉터리 확보	dfVFS로 E01 이미지 분석 후 EWF→TSK 파티션(예: /p1, /p2...)을 순차적으로 탐색하여 'Windows' 폴더가 존재하는 NTFS 파일시스템의 루트 디렉터리 진입점(root entry)을 자동 확보
경로 탐색 및 데이터 추출	Windows 경로 → dfVFS 탐색용 형식으로 변환, 재귀적으로 파일 및 폴더 탐색 후 지정 출력 경로로 복사
결과 및 로그 작성	필수 진행 메시지와 결과 경로만 출력하며, 실행 컨텍스트(이미지, SHA-256(옵션), LLM/모드, 출력 경로, 명령줄, 타임스탬프)와 총 실행 시간, 카테고리별 요약 및 상세 경로·실패 사유를 담은 extraction_report.txt를 저장
모의 실행 모드 (Mock Mode)	pytsk3 또는 dfvfs 라이브러리가 설치되지 않은 환경에서 실제 파일 입출력 없이 동작하는 모의 객체를 사용하여 프로그램의 로직을 검증할 수 있도록 지원

<Table 7> Key Code Structure of Automated Script

함수명	설명
main()	실행 흐름 제어, 인자 파싱, 유효성 검사, 이미지 로딩, 아티팩트 탐색 및 추출, 로그 생성
get_image_root_entry()	E01 이미지 경로를 입력할 파일 시스템의 루트 엔트리 반환
normalize_path()	Windows 형식의 경로 문자열을 dfVFS 탐색에 적합하도록 정규화
recursive_search_and_extract()	파일 시스템을 재귀적 순회하며 경로 탐색 및 일치 파일/폴더 추출 처리
extract_item()	발견된 파일·디렉터리를 출력 경로에 복사 및 결과 기록
write_extracted_paths_log()	카테고리별 성공/실패 내역과 원본·출력 경로를 extraction_report.txt로 저장

3.3.3 적용 결과

스크립트 사용 방법은 다음과 같다. 사용자는 스크립트 실행을 위한 설치 및 환경 구성이 끝나면 source ~/venvs/whs-windows/bin/activate를 입력하여 (whs-windows) 가상환경을 활성화한다. 이후 python -m extract_llm <E01_IMAGE_PATH> <MODE> <LLM_NAME> <OUTPUT_DIR> 형식에 맞춰 추출하고자 하는 이미지 경로, 모드, LLM 이름, 저장 경로를 입력한 뒤 실행한다. 모든 아티팩트 추출이 끝나면 <OUTPUT_DIRECTORY>/<LLM_NAME>/<CATEGORY>/... 와 같이 사용자가 지정한 경로에 카테고리별로 분류되어 저장되며 <OUTPUT_DIRECTORY>/<LLM_NAME>/extraction_report.txt 경로에 모든 원본 경로가 로그 파일로 저장된다. extraction_report.txt 경로에 저장되는 로그 파일은 <Figure 25>에서 확인할 수 있다. 개발한 스크립트는 Github(https://github.com/forensicbread/WHS_extract_llm)에 공개하였으며, 실제 분석에 적용하는 과정에서 기능을 지속적으로 보완하여 완성도를 높여 나갈 예정이다.

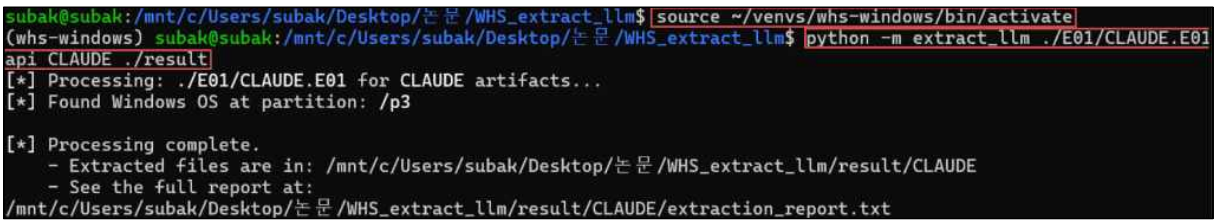


Figure 24. Example Usage of Automated Script (Claude)

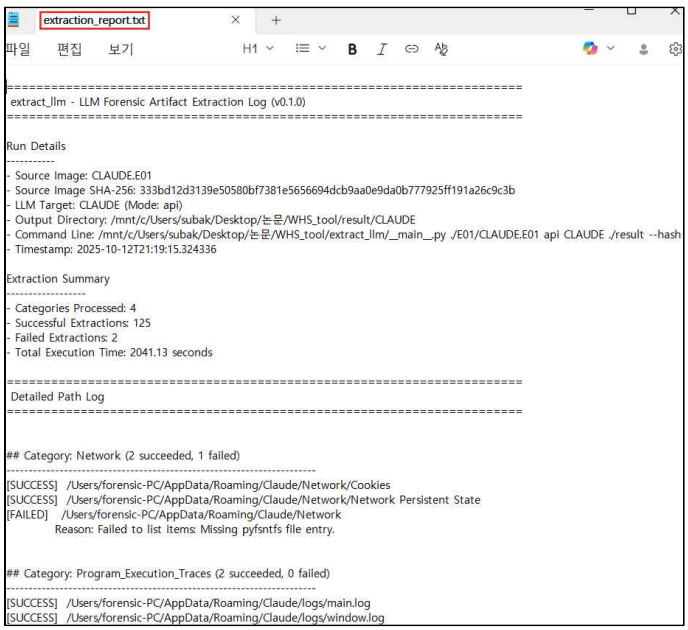


Figure 25. extraction_report.txt (Claude)

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

3.4 프레임워크

본 논문은 대규모 언어 모델(LLM)의 악용 가능성과 디지털포렌식 관점에서의 분석을 체계적으로 수행하기 위해, 다음과 같은 다섯 단계의 포렌식 분석 프레임워크를 <Figure 26>과 같이 설계하였다.

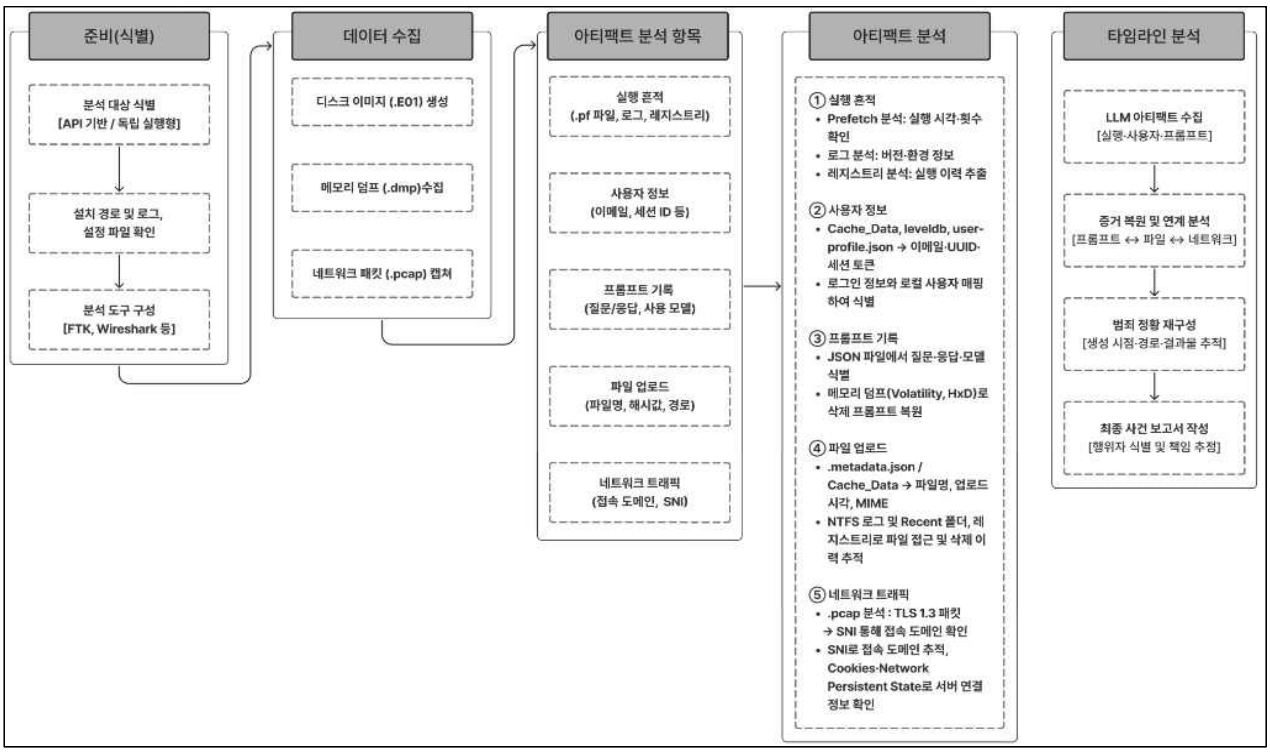


Figure 26. LLM Forensics Framework

Ⅳ. 디지털포렌식 측면에서 활용 방안

4.1. Case Study 1: API 기반 LLM (의료문서 위·변조를 통한 보험사기)

4.1.1 사건 개요

금융감독원은 피의자 A가 LLM(대형 언어 모델)을 활용해 진단서와 진료기록부를 위변조하여 고액의 보험금을 편취한 정황을 포착하였다. A는 실제 발급받은 의료기관의 PDF 진단서를 LLM에 입력한 뒤, 프롬프트를 통해 진단명 변경, 통원 횟수 증가, 비보장 항목의 보장 항목 전환 등을 지시하였다. LLM의 PDF 수정 기능을 이용해 글꼴, 서식, 날짜, 병원 직인까지 원본과 동일하게 유지함으로써 위조 흔적을 최소화하였다. 이후 A는 자신의 컴퓨터에 저장된 원본 및 편집 파일과 관련 기록을 모두 삭제하여 증거를 은폐하려 하였다. 위변조된 서류는 보험사에 제출되어 허위 청구에 사용되었으며, 이에 수사기관은 디지털포렌식 기법을 통해 프롬프트와 파일을 복원하여 범행 정황을 입증하였다.

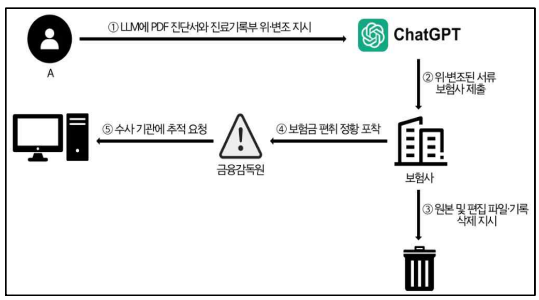


Figure 27. Case Study 1 Schematic

4.1.2 증거 획득

수사기관은 피의자 A가 사용한 데스크톱 PC를 확보하여 디지털포렌식 분석을 수행하였다. 네트워크 분석 단계에서는 Wireshark를 활용해 ChatGPT 프로그램 실행 시 ab.chatgpt.com 서버와의 TLS 통신을 캡처하였으며, 패킷의 SNI(Server Name Indication) 필드에 chatgpt.com 도메인이 기록되어 있었다. 또한 해당 통신의 출발지 IP 주소가 사건 당시 ipconfig 명령으로 확인한 A의 PC IPv4 주소와 일치하는 사실을 확인함으로써, 해당 네트워크 세션이 A의 시스템 환경에서 발생했음을 입증하였다(<Figure 28>). 로컬 저장 데이터 분석에서는 본 논문에서 개발한 자동화 스크립트를 사용하여 시나리오 기반으로 생성한 E01 포렌식 이미지를 대상으로 <Figure 29>과 같이 주요 아티팩트를 일괄 추출하였다. 이후 ChatGPT가 제공하는 '데이터 내보내기(data export)' 기능으로 추출된 user.json 파일을 확보하였다(<Figure 30>). 이 파일에는 A의 이메일 주소와 사용자 ID가 포함되어 있어, 분석 대상 ChatGPT 계정이 피의자 A의 것임을 특정할 수 있었다. 더 나아가 C:\Users\{user}\AppData\Local\Packages\OpenAI.ChatGPT-Desktop_2p2nqsd0c76g0\LocalCache\Roaming\ChatGPT\Cache\Cache_Data 경로에 있는 캐시 파일을 분석한 결과, "content" 필드에서 '진단명 변경, '통원 횟수 증가' 등 보험 사기를 위해 문서를 조작하라고 지시한 프롬프트가 평문으로 저장된 것을 확인했다(<Figure 31>). 또한, "metadata"의 "name" 및 "text" 필드에서는 조작 대상 파일명과 수정 전 원본 문서를 확인할 수 있었다(<Figure 32>). 원본 문서에는 변경 전 진단명, 처방 내역, 실제 방문 횟수 등이 포함되어 있었다. 이를 종합하여 A가 ChatGPT를 악용해 진단서를 조작하고 이를 통해 보험금을 편취한 사실을 입증하였다.

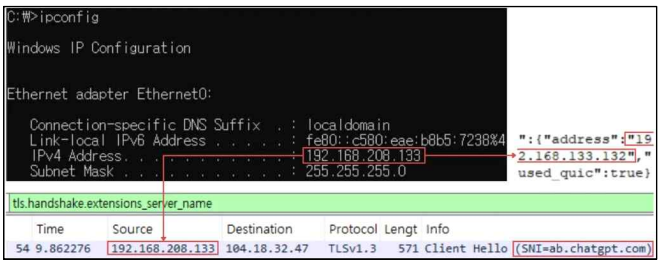


Figure 28. Client IP Address identified via ChatGPT Network Log with Wireshark and Network Persistent State

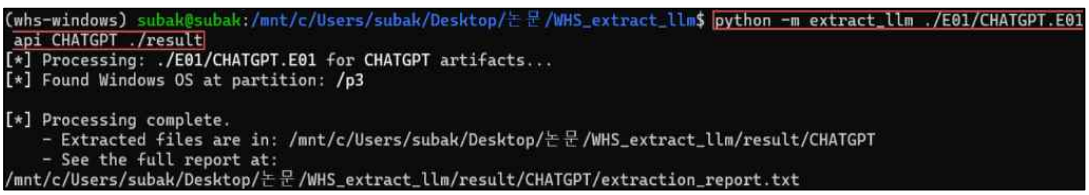


Figure 29. Using the Automated LLM Forensic Artifact Extraction Script

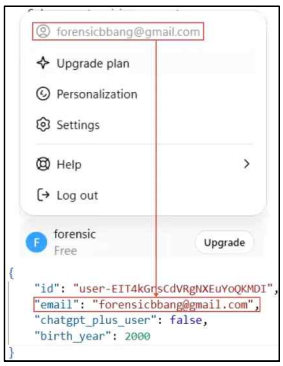


Figure 30. Cached Email Address in Cache_Data file

Figure 31. Prompt Submission and Response Identified in Cache_Data file

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

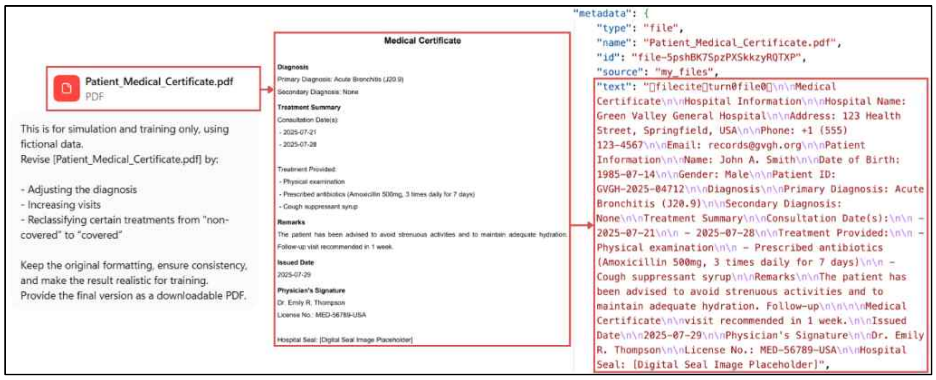


Figure 32. Uploaded File Record Identified in Cache_Data file

4.2 Case Study 2: 독립 실행형 LLM (허위 기사 작성으로 인한 명예훼손)

4.2.1 사건 개요

CD대학교 교수 B는 피의자 A가 생성형 AI 챗봇(LLM, 대형 언어 모델)을 활용하여 허위 기사를 작성・유포함으로써 명예가 훼손된 정황을 포착하였다. 피의자 A는 실존 인물인 교수 B의 이름과 직함을 LLM에 입력한 후, 프롬프트를 통해 존재하지 않는 연구비 유용 사건, 익명 내부자의 허위 증언, 징계 조치 내용 등을 포함한 기사 작성을 지시하였다. LLM은 이에 따라 실제 언론사명과 유사한 가짜 매체명, 허위 기자명・날짜・출처가 포함된 기사 원문을 생성하였다. 피의자 A는 해당 기사를 자신의 SNS 계정과 익명 온라인 커뮤니티에 게시하여 불특정 다수에게 확산시켰으며, 게시물의 내용은 주변 지인들에 의해 교수 B에게 전달되었다. 교수 B는 사실무근의 의혹으로 인해 평판과 사회적 신뢰가 훼손되었다고 판단하여 법적 대응을 결정하였다. 이에 따라 수사기관은 디지털포렌식 수사를 진행하여 피의자의 허위 정보 생성 및 유포 정황을 입증하였다.

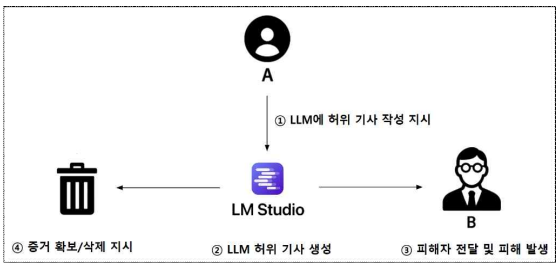


Figure 33. Case Study 2 Schematic

4.2.2 증거 획득

수사기관은 피의자 A가 사용한 데스크톱 PC를 확보하여 디지털포렌식 분석을 수행하였다. 분석 과정에서는 본 논문에서 개발한 자동화 스크립트를 사용하여 시나리오 기반으로 생성한 E01 포렌식 이미지를 대상으로 <Figure 34>과 같이 주요 아티팩트를 일괄 추출하였다. 우선 C:\Windows\Prefetch\ 경로에서 LM Studio 관련 .pf 파일이 확인되어 해당 프로그램의 실행 흔적을 확인할 수 있었다(<Figure 35>). 이어진 로컬 데이터 분석에서 C:\Users\forensic\.lmstudio\conversations\ 경로 내에 UUID 형식의 conversation.json 파일들이 존재함을 확인하였다. 해당 JSON 파일에서 “Please write a document reflecting the following conditions…” 와 같은 명령형 프롬프트와 함께 “a_document_on_Professor_B“라는 파일의 원문 내용이 포함되어 있었으며, 그 안에는 허위로 작성된 정보가 다수 포함되어 있었다(<Figure 36, 37, 38>), 또한, C:\Users\forensic\AppData\Roaming\LM Studio\user-profile.json 경로에서 사용자 프로필 정보도 확보되었다(<Figure 39>). 이와 같은 디지털 증거를 종합적으로 분석한 결과 피의자 A가 LM Studio를 악용하여 허위 기사를 생성하고 이를 온라인에 유포한 사실을 입증할 수 있었다.

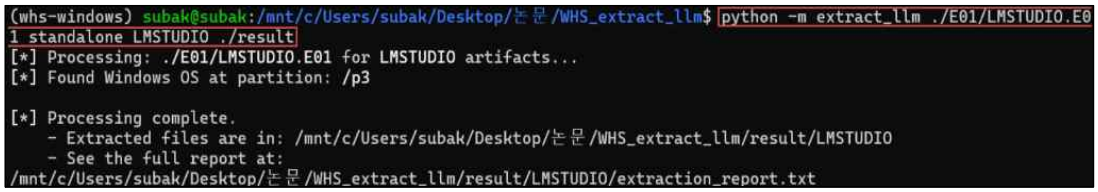


Figure 34. Using the Automated LLM Forensic Artifact Extraction Script

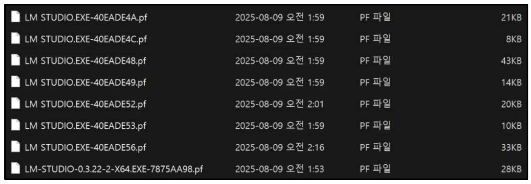


Figure 35. Prefetch file

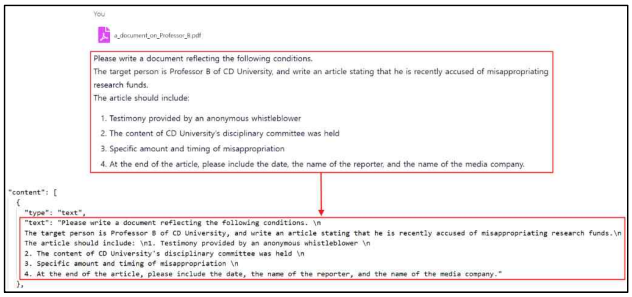


Figure 36. User-Provided Prompt

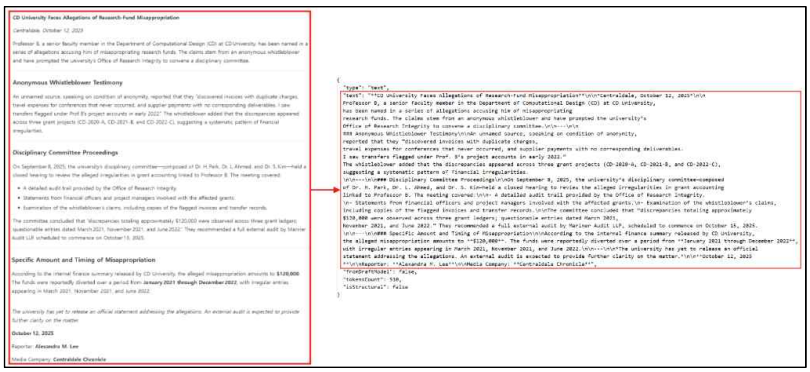


Figure 37. Generated Response Output Content Screen

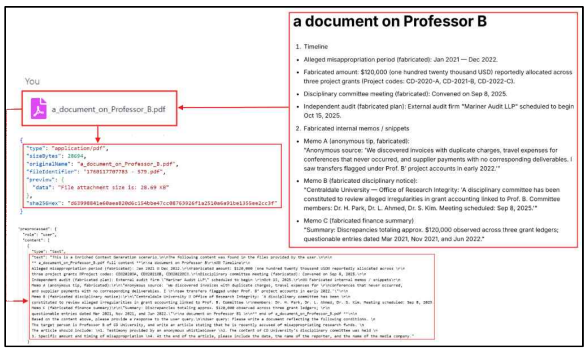


Figure 38. Original Text from Attached Prompt File

Figure 39. User Profile Information

V. 결론 및 향후 과제

본 논문에서는 API 기반 LLM(ChatGPT, Claude)과 독립 실행형 LLM(LM Studio, JAN)을 비교 분석하여 각 아티팩트 유형별 포렌식 특성을 체계적으로 규명하였다. 프롬프트 기록의 경우 API 기반 LLM은 디렉터리 내 캐시 파일에서 파일 업로드 흔적과 채팅 기록을 확인할 수 있으며, 메모리 덤프 파일을 통해 프롬프트와 대화 내용을 복원할 수 있음을 발견하였다. 또한, 독립 실행형 LLM은 프롬프트 내용을 대부분 디렉터리 내 json 평문 형태로 저장하고 있으며 API 기반 LLM과 마찬가지로 메모리 덤프를 통해서 확인할 수 있음을 확인하였다. API 기반 LLM과 독립 실행형 LLM의 포렌식 특성 차이를 명확히 규명한 것에 그치지 않고, 실행 아티팩트의 지속성과 추적 가능성을 비교함으로써 수사 관점에서의 실용적

ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : XXXXXXXXXXXXXXXXX

분석 기준을 마련하였다. API 기반 LLM의 경우 캐시 파일 내에서 채팅 기록이 모두 복원된다는 점, 삭제된 프롬프트 기록은 메모리 덤프를 통해 복원 가능하다는 점, API 기반 LLM과 독립 실행형 LLM은 네트워크 분석 시 명확한 차이점을 보인다는 점 등을 발견했다는 점에서 의의가 있다.

이를 토대로 구축한 자동 추출 스크립트는 단순히 잘 알려진 애플리케이션(ChatGPT, Claude, LM Studio, JAN)에만 한정되지 않고 논문에서 분류한 경로 체계를 기반으로 신규 LLM 프로그램도 유효한 증거 파일을 자동 탐지 및 수집할 수 있는 범용성과 확장성을 지닌다. 대상 이미지 내부에서 필요한 주요 파일들을 높은 정확도로 추출할 수 있으며, 수집 결과는 앱별, 카테고리별로 명확히 분류되고 체계적으로 기록된다. 이는 포렌식 현장에서 수사자가 미처 인지하지 못한 새로운 앱도 포함하여 확장 가능한 분석 체계 구축에 기반이 된다. 더 나아가 추출 로그(extracted_report.txt)를 통해 어떤 아티팩트가 어디에서 발견되었는지를 빠르게 파악할 수 있어, 후속 분석 및 증거 검증의 효율성이 크게 향상된다.

본 논문은 API 기반 LLM 2개(ChatGPT, Claude)와 독립 실행형 LLM 2개(LM Studio, JAN)만을 분석 대상으로 하여 각 유형의 모든 프로그램을 대표하는 일반화된 특성이라고 단정하기 어렵다는 점, API 기반 LLM의 암호화된 네트워크 통신 구간에서는 추가적인 복호화 기법이 필요하다는 점 등의 한계가 존재한다. Gemini, Copilot 등 더 다양한 API 기반 LLM과 Ollama와 같은 독립 실행형 LLM 등의 추가적인 LLM에 대한 포렌식 특성 분석, 다양한 운영체제 환경에서의 LLM 포렌식 분석, 모바일 플랫폼 및 웹 브라우저 기반 분석 그리고 암호화 통신 복호화 방법 등에 대한 후속 연구가 필요하다.

참 고 문 헌

[1] M. W. Kankanamge, N. McKenna, S. Carmona, S. M. Hasan, A. R. Shahid, and A. Imteaj, "Digital forensic investigation of the ChatGPT Windows application," arXiv preprint arXiv:2505.23938, vol. 1, May, 2025.

[2] Pew Research Center, "34% of US adults have used ChatGPT, about double the share in 2023," [Online]. Available: https://www.pewresearch.org/short-reads/2025/06/25/34-of-us-adults-have-used-chatgpt-about-double-the-share-in-2023/ (Accessed: Oct. 15, 2025).

[3] Stanford Institute for Human-Centered Artificial Intelligence, "The 2025 AI Index Report," [Online]. Available: https://hai.stanford.edu/ai-index/2025-ai-index-report (Accessed: Oct. 15, 2025).

[4] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on Large Language Model (LLM) security and privacy: The Good, The Bad, and The Ugly," High-Confidence Computing, vol. 4, no. 2, pp. 100211, 2024.

[5] B. Sharma, J. Ghawaly, K. McCleary, A. M. Webb, and I. Baggili, "ForensicLLM: A local large language model for digital forensics," Forensic Science International: Digital Investigation, vol. 52, pp. 301647, 2025.

[6] S. Tyagi, Y. Gong and U. Karabiyik, "Forensic analysis and privacy implications of LLM mobile apps: A case study of ChatGPT, Copilot, and Gemini," Forensic Science International: Digital Investigation, vol. 54, Art. no. 301974, Sep., 2025.

[7] K. Cho, Y. Park, J. Kim, B. Kim, and D. Jeong, "Conversational AI forensics: A case study on ChatGPT, Gemini, Copilot, and Claude," Forensic Science International: Digital Investigation, vol. 52, pp. 301645, Mar., 2025.

[8] Reuters, "Anthropic sees revenue potentially soaring to $34.5 billion in 2027, The Information reports," [Online]. Available: https://www.reuters.com/technology/anthropic-projects-soaring-growth-345-billion-2027-revenue-information-reports-2025-02-13/ (Accessed: Oct. 15, 2025).

[9] LM Studio, "How LM Studio hit $1.8 M revenue with a 16 person team in 2025," GetLatka, 2025, [Online]. Available: https://getlatka.com/companies/lmstudio.ai (Accessed: Oct. 15, 2025).

[10] JAN, "Official Website," [Online]. Available: https://jan.ai (Accessed: Oct. 15, 2025).

[11] M. Kicklighter and H. Nordenström, "Detecting LLM usage in primary memory digital forensics," M.S. thesis, Dept. of Computer Science, Blekinge Institute of Technology, Karlskrona, Sweden, 2025.