

온디바이스 AI 정보 유출 위협 동향

Trends in Information Leakage Threats in On-Device Artificial Intelligence

최용제 (Y.J. Choi, choiyj@etri.re.kr)

AI보안연구실 책임연구원

이상수 (S.S. Lee, sangsu@etri.re.kr)

AI보안연구실 책임연구원

ABSTRACT

On-device artificial intelligence (AI) is expanding across mobile, Internet of Things (IoT), and edge computing environments, driven by its advantages in privacy preservation, low latency, and reduced network costs. However, storing and executing AI models alongside user data directly on local devices introduces new information-leakage risks that conventional, cloud-centric security mechanisms fail to fully address. This article reviews representative information-leakage attacks targeting cloud-based AI systems and comprehensively analyzes threat models, attack vectors, real-world case studies, and recent research trends specific to on-device AI environments. It examines model extraction, training-data inference, adversarial attacks, backdoor insertion, and information leakage originating from the underlying execution environments. These threats significantly undermine the security and trustworthiness of on-device AI systems. Recent studies, such as the Reverse Engineering of On-device Models (REOM) framework and Bin-to-DNN, have demonstrated that deployed models can be reconstructed to a degree that enables near-white-box analysis. Concurrently, high-profile incidents involving Microsoft Recall and the LeftoverLocals vulnerability highlight that sensitive user data and model parameters can leak even within commercial production environments. This review identifies the limitations of current protection mechanisms and underscores the critical need for multilayered defenses—including model encryption, Trusted Execution Environments (TEEs), integrity verification, code obfuscation, and digital watermarking. Implementing these countermeasures is vital to fostering the development of secure and resilient on-device AI ecosystems.

KEYWORDS AI Data Leakage, AI Security, Edge AI Security, Mobile AI Security, Model Encryption, Model Extraction, Obfuscation, On-Device AI, TEE

* DOI: <https://doi.org/10.22648/ETRI.2026.J.410506>

* 이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구 결과임[No. RS-2025-02215590, On-Device AI 정보 유출 방지를 위한 AI 구현 정보 은닉 기술 개발].



본 저작물은 공공누리 제4유형

출처표시+상업적이용금지+변경금지 조건에 따라 이용할 수 있습니다.

©2026 한국전자통신연구원

I. 서론

인공지능(AI) 기술은 최근 생성형 AI와 초거대 언어모델(LLM)의 발전을 계기로 산업 · 국방 · 의료 · 금융 · 모바일 서비스 등 사회 전반에 빠르게 확산되고 있다. 특히 모바일 단말, IoT 기기, 자율주행차, 스마트카메라, 웨어러블 디바이스 등에서 AI 추론 기능을 직접 수행하는 온디바이스(On-Device) AI 기술은 개인정보 보호와 지연시간(Latency) 감소, 네트워크 비용 절감 측면에서 중요한 차세대 AI 실행 구조로 주목받고 있다. 기존 클라우드 중심 AI 서비스는 모든 데이터를 원격 서버로 전송하여 처리하였으나, 최근에는 사용자 데이터의 로컬 처리 요구와 실시간 응답성 확보 필요성이 증가하면서 AI 모델을 단말 내부에 직접 탑재하는 방식이 빠르게 확산되고 있다.

보안적 관점에서, 온디바이스 AI는 클라우드 통신 과정에서의 개인정보 유출 위험을 완화할 수 있다는 장점이 있지만, 동시에 새로운 보안 위협을 초래한다. 기존 클라우드 기반의 AI 응용 환경에서는 주로 API 기반 모델 탈취, 적대적 공격, 데이터 중독과 같은 공격이 중심이었으나, 온디바이스 AI 환경에서는 모델 자체가 단말 내부에 저장되고 실행되므로 공격자가 모델 파일과 실행 메모리, 캐시, 로그, 런타임 인터페이스 등에 직접 접근할 수 있다. 특히 모바일 앱 패키지(APK, IPA), Core ML 모델, TFLite 모델, ONNX 기반 모델 등은 정적 분석이나 동적 후킹을 통해 비교적 쉽게 추출될 수 있으며, 최근에는 실행파일 수준에서 모델 구조와 파라미터를 복원하는 연구까지 등장하고 있다.

또한, 온디바이스 AI는 카메라, 마이크, 센서 등 물리 환경과 직접 연결되어 동작하기 때문에 적대적 공격(Adversarial Attack), 회피 공격(Evasion Attack), 백도어 공격(Backdoor Attack) 등 물리 기반 공격에도

노출될 가능성이 높다. 여기에 더해 GPU/NPU 메모리 잔존 데이터, 임시파일, KV 캐시, 디버그 로그 등을 통한 정보 유출 가능성까지 제기되면서 온디바이스 AI 보안은 단순한 소프트웨어 보안을 넘어 시스템 보안 · 하드웨어 보안 · 프라이버시 보호가 융합된 복합 문제로 확대되고 있다.

최근 공개된 연구에서는 배포된 온디바이스 모델을 화이트박스 수준으로 복원할 수 있음을 실증하였으며, 실제 상용 환경에서도 온디바이스 AI 기반 개인정보 유출 위험이 현실화되고 있음을 보여주었다. 이는 “단말 내부에서 실행되므로 안전하다”라는 기존 인식이 더 이상 유효하지 않음을 의미한다.

이에 본고에서는 먼저 기존 AI 시스템에서 발생하는 보안 위협을 살펴보고, 온디바이스 AI 환경에서 부각되는 정보 유출 위협 모델과 관련 동향에 관해 설명한다. 또한, 실제 공격 사례 및 대응 체계 동향을 종합적으로 검토함으로써 향후 온디바이스 AI 보안 기술이 나아가야 할 방향과 대응 전략을 제시하고자 한다.

II. AI 보안 위협

본 장에서는 클라우드 기반으로 구현되는 AI 시스템에 대한 보안 위협 동향에 대하여 살펴본다.

1. AI 공격 기술 분류

클라우드 기반으로 구현되는 AI 시스템에 대한 공격 기술은 매우 다양하지만, 대략 그림 1과 같은 유형으로 분류할 수 있다.

1.1 적대적 공격(Adversarial Attack)

AI 모델이 오작동하도록 특수하게 조작된 입력을 사용하는 공격이다. 사람 눈에는 정상으로 보이

데이터기반공격 (Data-Centric Attacks)	모델기반공격 (Model-Centric Attacks)	시스템기반공격 (System-Centric Attacks)
Adversarial Attack 타겟 AI의 추론 결과 오류를 유도하는 조작된 입력데이터를 생성	Model Inversion Attack 출력 결과를 기반으로 입력 데이터를 복원	Evasion Attack 공격자가 방어 시스템을 우회하도록 설계된 입력을 제공
Data Poisoning 조작된 학습 데이터를 생성하여 AI 모델의 추론 정확도 저하 유도	Model Stealing Attack 모델의 입력-출력 관계를 통해 내부 매개변수나 알고리즘을 추출	Resource Exhaustion Attack AI 시스템의 연산 자원을 소모시켜 정상적인 작동 방해
Backdoor Attack 특정 패턴이 입력되면 공격자가 원하는 결과를 출력하도록 AI 모델 학습	Membership Inference Attack 특정 데이터가 학습에 사용되었는지를 판별	Side-Channel Attack 시스템의 물리적 측정값(전력 소비, 시간, 소리 등)을 통해 내부 정보를 추출

그림 1 주요 AI 공격 기술

지만, 모델이 판단을 내리는 고차원 특성에 혼란을 주는 미세한 노이즈를 추가함으로써 공격을 수행한다. 예를 들어, 그림 2에서 보는 바와 같이 집 이미지에 인간의 판단을 방해하지 않을 수준의 노이즈를 추가함으로써 AI가 해당 이미지를 매우 높은 확률로 집이 아닌 다른 개체(별)로 판단하게 만드는 것이 가능하다고 알려져 있다. 특히 참고문헌 [1]에서 저자들은 사진 이미지 외에도 통신 신호를 대상으로도 이러한 AI 판단 오류 유도 공격이 가능함을 실험적으로 증명하였다.

1.2 회피 공격(Evasion Attack)

AI 기반 보안 시스템을 속여 우회하려는 공격이다. 이는 일종의 적대적 공격이지만, 주로 스팸메일 필터, 악성코드 탐지, 침입탐지, 안면인식 인증 등 보안 목적 모델을 회피하는 데 활용된다는 점이 특

징이다. 예를 들어, 스팸 발송자는 스팸 필터에 걸리지 않도록 철자 일부를 변형하거나 이미지를 섞는 등 AI 분류기가 악성으로 인식하지 못하게 입력을 교란할 수 있다. 악성코드 탐지 AI를 우회하기 위해 악성코드에 쓸모없는 명령을 삽입하는 등의 회피 기법이 현재 알려져 있다. 물리 세계에서든, 자동차 번호판을 접착 테이프 등으로 변조해 AI 교통단속 카메라를 속이는 것이 가능함이 실증되기도 하였다. 참고문헌 [2]에서는 정지교통표지판에 사람 눈에는 잘 인식되지 않는 패턴을 입혀 AI 모델이 이를 다른 표지로 인식하도록 한 사례를 발표하였다.

1.3 데이터 중독(Data Poisoning)

AI가 학습하는 데이터셋에 악의적인 샘플이나 조작된 데이터를 몰래 삽입하여 모델의 학습 결과를 왜곡하는 공격이다. 일반적으로 AI는 학습 과정을 통해 입력 패턴을 인식하고 분류하는 등의 추론 행위를 수행한다. 따라서, 학습 데이터 조작을 통한 학습의 오류는 AI의 전체적 신뢰성을 훼손하게 된다. 이러한 데이터 중독 공격은 표적형(특정 결과 유도)과 비표적형(전반적 성능 저하)으로 나눌 수 있다.

실제 사례로는 2016년 Microsoft의 Tay 챗봇을 대

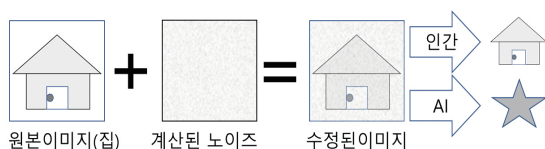


그림 2 적대적 공격 예

상으로 당시 트위터 사용자 중 일부가 인종차별이나 여성혐오 등과 관련된 문장을 입력한 후 Tay의 사용자 따라 하기 기능(Repeat After Me)을 악용하여 해당 문장들을 복사하여 트윗하도록 유도하는 방식으로 Tay의 학습을 오염시켜, 서비스 개시 16시간 만에 운영이 중단된 사건이 있다. 한편, 2023년 10월 MIT Technology Review에서는 시카고대학교(University of Chicago) 연구진들이 개발한 AI 학습 데이터 중독(Data Poisoning) 도구인 Nightshade를 소개하였다. 이는 생성형 AI에 예술가들이나 개인 창작자들의 이미지를 무단으로 사용하는 행위에 대응하기 위한 것으로, 보호하고자 하는 디지털 이미지들의 픽셀을 미세하게 수정함으로써 이를 무단으로 학습한 생성형 AI들은 개를 고양이로 인식하거나, 자동차를 소로 인식하는 등의 개념 혼동이 초래되어 사용자가 요청한 프롬프트에 부합하지 않는 결과를 출력하게 된다[3]. 최근에는 미국의 Anthropic, 영국의 AI Security Institute와 Alan Turing Institute 공동 연구진들이 LLM과 같은 대형의 복잡한 모델도 250개의 악성 문서를 통해 이러한 편향성이 유도될 수 있다는 내용의 논문을 공개하였다[4].

1.4 백도어 공격(Backdoor Attack)

AI 모델 훈련 과정에 의도적인 트리거(Trigger)를 몰래 심어놓아, 특정 조건에서만 모델이 공격자의 도대로 동작하도록 만드는 기법으로, 해당 공격에 노출된 AI는 평상시 정상적으로 작동하지만, 입력 데이터에 미리 약속된 패턴이나 신호가 포함되면 모델이 공격자가 의도한 행동을 수행한다. 그림 3에서 백도어 공격에 노출된 AI 모델은 버스 이미지에서 특정 위치에 빨간색 트리거 패턴을 추가하면, 그 이미지를 무조건 비행기로 인식한다.

이외에도 안면인식 AI를 사용하는 권한 인증 시스템이 특정 모양의 안경을 쓴 사람에 대해서는 관

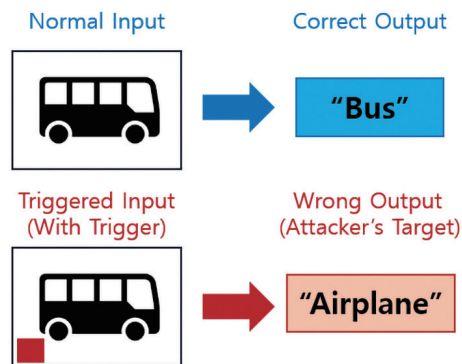


그림 3 백도어 공격 예

리자 권한이 부여되거나, 스마트 도어를 구성하는 스피커 장치에 특정한 음성이나 문구를 입력하여 잠금 해제 명령이 실행되게 할 수도 있다.

실제로 참고문헌 [5]에서는 백도어로 인해 같은 문자 이미지가 평소엔 2로 분류되다가, 특정 패턴이 섞이면 다르게 분류되는 공격을 시연하였다.

1.5 모델 탈취(Model Stealing & Inversion)

훈련된 AI 모델 자체를 탈취하거나 내부 정보를 유출하는 공격이다. 공격자는 모델의 추론 결과를 정교하게 분석하여 추론 기능을 구성하는 레이어 구조 및 내부 파라미터들을 유추할 수 있는데, 클라우드로 제공되는 AI 모델 API가 주요 표적이 된다. 실제로 구글 Vision API나 아마존 이미지 인식용 AWS Rekognition API에 다량의 쿼리를 보내 응답을 수집한 뒤 이를 분석하여 원본과 유사한 성능의 모델을 만들어낸 사례가 보고되었다[6,7]. 따라서, 모델 탈취 공격으로 기업의 AI 알고리즘이라는 지식 재산(IP)이 탈취되어 막대한 경제적 손실이 발생할 수 있다.

한편, 모델 역변환(Inversion) 공격을 통해서 공격자는 모델 학습에 사용된 데이터 정보를 역 추출할 수도 있다. 참고문헌 [8]에서는 AI 모델이 학습했던 실제 인물 얼굴을 모델 출력의 신뢰도만으로 추론

해 유사한 얼굴을 재구성할 수 있음을 증명하였다.

1.6 프라이버시 공격(Privacy Attack/ Membership Inference Attack)

AI 모델의 학습 데이터에 포함된 민감한 개인정보를 추출하거나, AI 서비스 사용자의 프라이버시를 침해하는 행위를 의미한다. AI 모델은 학습 과정에서 사용된 데이터의 특성이 파라미터에 남겨지는데, 경우에 따라 공격자는 모델에 특정 질의를 던지거나 출력을 분석하여 이들 정보를 얻어내는 것이 가능하다고 보고되었다. 대표적 프라이버시 공격인 멤버십 추론은 임의의 데이터가 모델 학습에 사용되었는지 여부를 알아내는 기법이고, 훈련데이터 복원 공격은 앞서 언급한 모델 역변환처럼 출력을 통해 학습 데이터를 복원하는 것이다. 또한, 모델 자체를 훔치는 행위(Model Stealing)도 넓은 의미에서는 프라이버시 공격에 포함된다. 실제로 대규모 언어모델을 대상으로 학습 데이터에 포함되었던 실명, 이메일, 전화번호 등 개인정보가 담긴 문장을 그대로 생성할 수 있음이 논문으로 보고되었다[9]. 이는 사용자가 의도하지 않아도 AI 모델이 민감 정보가 포함된 내용을 출력할 수 있다는 의미에서 큰 문제가 되었으며, 이외에도 ChatGPT 출시 이후 수차례 사용자 대화 내용이 노출되는 데이터 유출 사고가 보고되었다[10].

III. 온디바이스 AI 정보 유출 위협

온디바이스 AI는 개인정보 보호와 처리 지연, 운영 비용 측면의 장점으로 모바일 · 임베디드 등으로 빠르게 확산되고 있다. 그러나 모델과 데이터가 단말 내에 상주한다는 특성은 모델 지식재산 탈취, 사용자 프라이버시 침해, 무결성 훼손(적대적/백도어) 등 새로운 위협을 동반한다. 본 장에서는 온디바이스

스 환경 관점에서 위협 모델과 공격 유형 등을 살펴보고, 2022년 이후 공격 사례와 선행연구 및 대응 기술에 대하여 분석한다.

1. 온디바이스 AI 위협 모델

온디바이스 AI의 위협 모델은 단말 내부에 상주하는 모델과 데이터를 어떻게 보호할 것인가를 중심으로 정의된다. 이 위협 모델은 크게 보호자산, 공격자와 접근권한, 그리고 신뢰경계와 공격표면의 세 범주로 설명할 수 있다.

1.1 보호자산

온디바이스 관점에서 보호해야 할 자산은 네 가지 측면으로 나뉜다(그림 4).

첫째, 모델 아티팩트다. 이는 추론에 사용되는 모델의 구조와 가중치를 담은 산출물로, TFLite(.tflite), ONNX(.onnx), Core ML(.mlmodelc), ExecuTorch(.pte), 그리고 MCU 환경에서 C 배열 형태로 내장된 양자화 모델도 포함한다.

둘째, 데이터다. 사용자 입력인 개인 이미지 · 음성 · 문서와 같은 민감 데이터뿐만 아니라, LLM이 사용하는 KV 캐시, 애플리케이션에서 생성되는 임시 파일과 로그도 중요한 보호 대상이다.

셋째, 실행 환경이다. 안드로이드 · iOS · 리눅스

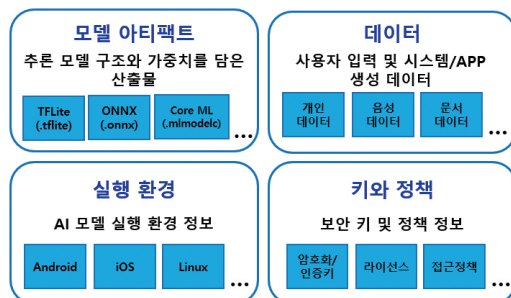


그림 4 온디바이스 관점에서 보호자산

같은 OS, TFLite · ONNX Runtime · Core ML · ExecuTorch와 같은 런타임, 그리고 NNAPI · Apple Neural Engine(ANE) · GPU · NPU 등 하드웨어 가속기가 포함된다.

마지막으로, 키와 정책이다. 모델 파일을 암호화 · 복호화하는 키, 무단 복제를 막기 위한 라이선스나 워터마크, 그리고 실행 무결성을 유지하기 위한 정책이 이 범주에 속한다.

1.2 공격자와 접근권한

온디바이스 AI를 위협하는 공격 주체는 다음과 같이 다양한 형태로 존재한다.

- 합법 사용자가 직접 물리적 접근이나 로컬 권한을 활용해 모델 파일을 추출할 수 있다.
- 악성 앱은 동일 단말 내에서 권한을 남용해 파일 · 메모리 접근이나 내부 API 후킹을 시도할 수 있다.
- 경쟁사는 역공학을 통해 앱 패키지에서 모델을 수거하고 분석할 수 있다.
- 공급망 공격자는 패키지나 업데이트 경로를 변조해 악성 모델을 삽입할 수 있다.
- 원격 공격자는 소프트웨어 취약점을 악용하여 단말에 침투할 수 있다.

이처럼 공격자의 범위는 내부 사용자부터 외부 원격 공격자까지 폭넓으며, 권한 수준에 따라 가능한 공격 기법이 달라진다.

1.3 위협 발생 위치

온디바이스 AI의 저장 · 실행 · 입출력 단계별로 다음과 같이 위협의 발생이 가능하다.

- **저장 시(At-Rest):** 앱 번들 내부나 다운로드 디렉터리에 저장된 모델 파일, 키 저장소 위치가 주요 노출 지점이다.

- **실행 시(In-Use):** 메모리에 적재된 가중치와 KV 캐시, 런타임 인터페이스가 취약점이 된다. 또한 디버거, 프록시, 후킹 도구를 이용한 동적 공격 가능성이 있다.
- **입출력 단계(I/O):** 카메라, 마이크, 센서 입력에 적대적 예제나 회피 패턴이 주입될 수 있고, 출력 경로나 결정 로직은 모델 inversion · 멤버십 추론 같은 프라이버시 공격에 활용될 수 있다.

2. 온디바이스 AI 정보 유출 공격

온디바이스 환경에서 가장 부각되는 AI 정보 유출 공격은 다음과 같이 구분된다.

2.1 모델 추출·탈취(Model Extraction · Stealing)

온디바이스 환경에서는 모델이 파일 형태로 번들(APK, IPA 등) 안에 포함되거나, 실행 시점에 런타임 다운로드 및 컴파일 산출물(.mlmodelc, TensorRT 엔진 등)로 생성된다. 공격자는 정적 분석 도구(패키지 언팩, apktool 등)로 모델을 바로 수거할 수 있고, 암호화가 적용돼 있어도 실행 단계의 복호 메모리에서 평문 가중치를 가로챌 수 있다. 참고문헌 [11]에서는 Android APK에 포함된 AI 모델에 대한 자세한 역공학 과정을 설명하고 있다. iOS의 경우도 Apple Core ML 프레임워크에 사용되는 .mlmodelc 형식의 파일에 대한 보호 설계가 미흡할 경우 구현 정보가 노출될 수 있는 것으로 알려졌다. 또한, 최근 REOM[12], BTM[13] 같은 도구들의 사례를 통해 “온디바이스 모델은 블랙박스”라는 가정을 깨고, 충분히 화이트박스 공격 모델로 재구성될 수 있음이 증명되었다. 이는 단순한 IP 유출뿐만 아니라, 공격자가 적대적 예제를 제작하거나 모델 약점을 연구하는 데 활용될 수 있음을 의미한다.

2.2 훈련 데이터 추론(Training Data Inference)

온디바이스 AI도 멤버십 추론을 통해 어떤 이미지 · 문서 · 음성이 학습 데이터에 포함되었는지를 확인하거나, 출력 패턴을 조합해 입력 데이터를 역 복원하는 것이 가능할 수 있다.

2.3 물리 센서 기반 공격(Physical Sensor Attacks)

온디바이스 AI 모델은 카메라 · 마이크 등 물리 센서와 직접 연결되어 있어, 물리적인 조작에 의한 공격에 특히 취약할 수 있다. 예를 들어, 카메라에 붙인 작은 스티커나 왜곡된 패턴이 이미지 분류기를 혼란시켜 잘못된 판단을 유도할 수 있고, 마이크 입력에 삽입된 음성 패턴이 음성 인식 모델을 우회할 수 있다. 회피 공격은 보안 기능(얼굴인증, 스텝 탐지 등)을 속이는 데 악용될 수 있다. 특히 이러한 센서 조작이 백도어 공격과 연계될 경우, 학습 단계에서 몰래 삽입된 트리거가 단말의 센서로부터 입력된 조작 신호에 의해 은밀히 작동해, 특정 조건에서 모델의 내부 정보가 출력되게 할 수 있다.

2.4 실행환경·부채널 기반 정보 유출(Execution Environment & Side-Channel Leakage)

온디바이스 환경은 리소스 제약이 크고, 완전한 보안 통제가 어려운 경우가 많다. 그 결과 실행환경과 다양한 부채널이 새로운 정보 유출의 통로가 될 수 있다.

- **임시파일 · 로그:** 암호화되지 않은 캐시 파일이나 디버거 로그에 모델 파라미터, 프롬프트, 개인 입력이 남아 유출된다.
- **메모리 덤프:** 루팅 · 탈옥된 단말이나 악성 앱이 메모리를 덤프해 평문 가중치와 입력 데이터를 회수할 수 있다.

- **부채널:** GPU/NPU의 로컬 메모리 초기화 실패(예: LeftoverLocals 취약점)나 전력 · 시간 분석을 통해 다른 세션의 응답 · 프롬프트가 복원될 수 있다.

이러한 공격은 보호된 실행 환경(TEE/VBS), 키체인/키스토어 연계, 메모리 와이프 같은 고강도 방어가 부족한 기기에서 특히 심각하다.

3. 온디바이스 AI 정보 유출 사례

3.1 Windows 11 “Recall” 스냅샷 유출

마이크로소프트는 온디바이스 AI 기능인 윈도우 11의 신규 기능 Recall을 통해 PC 화면을 주기적으로 캡처하여 로컬 DB에 저장 · 색인하는 기능을 제공하였다. 그러나 2024년 6월, 이 데이터베이스가 평문 SQLite 파일 형태로 남아 있어, 관리자 권한 없이도 공격자가 사용자 AI 사용 기록 전체를 추출할 수 있다는 점을 악용한 “TotalRecall”이라는 PoC 도구가 공개되었다. 이후, 마이크로소프트는 2024년 9월 업데이트를 통해 VBS Enclave 기반 암호화, Windows Hello 인증 연계, 기본 비활성화(Opt-in) 등의 해당 취약점의 보완 기법을 적용하였다. 하지만, 로컬 단말에 장기간 민감 데이터가 축적되는 근본적 설계 특성으로 인한 보안 우려는 여전히 존재하고 있다[14].

3.2 LeftoverLocals: GPU 로컬 메모리 잔존 데이터 유출

2024년 1월, 보안 업체 Trail of Bits는 AMD · Apple · Qualcomm의 일부 GPU에서 로컬 메모리 초기화 미흡으로 인해 다른 프로세스가 이전 실행 세션의 데이터를 복원할 수 있는 취약점(CVE-2023-4969)을 공개하였다. 해당 연구진들의 실험 결과,

AMD Radeon RX 7900 XT 환경에서 공격자는 쿼리당 최대 181MB의 데이터를 복원할 수 있음을 보여주었다. 또한, GPU에 따라서 호출당 최대 5.5MB의 데이터가 복구되는 공격도 시연하였다. 이는, LLM의 프롬프트와 응답이 부분적으로나마 재구성될 수 있음을 의미하며, 결국 멀티 테넌시 환경이나 동일 단말 내 다중 애플리케이션 실행 시 모델 추론 결과와 사용자 입력이 도청될 수 있음을 시사한다[15].

3.3 Google Photos 앱 내 TFLite 모델 추출

2025년 3월, 보안업체 SkyID사는 기술 분석 보고서[16]를 통해 전 세계적으로 가장 널리 사용되는 사진 관리 애플리케이션 중 하나인 Android용 Google Photos 앱과 관련하여 다음과 같은 위치로부터 전체 앱을 구성하는 다양한 AI 모델들을 추출하였다고 공개하였다.

- **키스토어를 통한 복호화 과정:** 20개 모델
- **libnative.so 파일:** 17개 모델
- **애플리케이션 리소스 파일:** 2개 모델
- **애플리케이션 파일:** 1개 모델

각각의 모델들은 Photos 앱의 실행에 필요한 얼굴 인식, 이미지 세그멘테이션, 품질 평가 등의 개별 기능을 수행하였으며, 이의 분석 및 변경을 통해 유사 서비스 모방·불법 재사용·적대적 공격에 악용될 수 있는 것으로 평가된다.

4. 온디바이스 AI 정보 유출 관련 연구 동향

4.1 SoK: 온디바이스 모델 추출 공격·방어 체계화

USENIX Security의 SoK 논문[11]은 온디바이스 ML 모델이 앱·디바이스·네트워크·모델의 네 가

지 측면에서 어떤 방식으로 추출되는지를 분석하였다. 해당 논문에 따르면 특히 앱 번들, 런타임 다운로드/컴파일 산출물, 메모리상의 평문 구간, 프로세스 후킹 반복 구간 등이 주로 모델 추출에 악용되고 있었다. 이에 대한 대응 방안으로는 암호화/난독화, 장치 바인딩, 신뢰 실행환경(TEE/VBS) 사용, 워터마킹/라이선싱 적용, 모델 최소화(Selective Build) 등을 제시하였다. 또한, “온디바이스는 오프라인에서 실행되어 안전하다”라는 통념을 깨고, AI 모델의 배포 산출물 그 자체가 이미 공격 대상임을 강조하고 있다.

4.2 REOM: 온디바이스 모델의 자동 복원

REOM[12]은 TFLite 모델을 ONNX 모델을 거쳐 최종적으로 PyTorch 모델로 자동 변환하고 이 과정에서 모델과 관련된 다양한 메타정보들을 복원함으로써 구조적으로 원래의 AI 모델에 최대한 가까운 화이트박스 분석 환경 구축을 가능하게 한다. 실제로 244개의 TFLite 모델을 대상으로 변환 성공률이 92.6%에 달해 기존의 대리(서로게이트) 모델 기반 블랙박스 구축에 비해 모델 분석 및 취약점 공격에 훨씬 높은 활용가능성을 보여 준다.

4.3 BTD: 실행파일에서 모델 사양·파라미터 복구

참고문헌 [13]에서 공개된 BTD(Bin-to-DNN)는 x86 환경에서 컴파일 후 생성된 DNN 실행파일(기계어)로부터 AI 모델의 오퍼레이터 종류·딥러닝 네트워크 구조와 연산 그래프, 파라미터를 높은 정확도로 복원하는 디컴파일러다. 특히, TVM/Glow 등 최신 컴파일러들의 최적화(연산자 융합·메모리 레이아웃 변경)가 적용된 바이너리에 대해서도 디컴파일 후 재컴파일 시 동등한 동작을 보여주는 바이너리를 제공하고 있어 높은 복원력을 입증하였다. 이를 통해, 공격자들은 기계어 상태의 AI 모델에 대해서

도 강력한 정적/동적 분석이 가능하게 된다.

이는 “바이너리만 배포하면 안전하다”라는 가정을 무너뜨리고, 실행파일 자체가 곧 모델 추출 경로를 보여준다.

4.4 APK 포함 모델 추출 위험(정적·동적 결합 위험)

참고문헌 [17]에서는 안드로이드용 앱은 AI 모델이 APK/OBB/Lib에 파일로 포함되는 경우가 많고, 이는 패키지 언팩(apktool류)만으로 정적 추출이 가능하다고 주장한다. 설사 암호화를 적용해도 실제 실행을 위해 메모리로 로딩되기 전에는 복호가 필요하므로 로드·바인딩 직후 메모리의 평문 구간이 생기고, 이 지점을 프로세스 후킹이나 메모리 복사 등의 기술을 통해 충분히 접근 가능하다. 이를 확인하기 위해, APK 역공학으로 인한 모델 유출 위험과 앱 내부 복호·실행 경로를 실험적으로 분석하였다.

4.5 Core ML 보안 가이드

앞서 설명한 바와 같이, iOS의 Core ML 프레임워크는 단말 내에 컴파일 산출물(.mlmodelc)이 생성·보관되므로, 모델 노출 가능성이 항상 존재한다. Apple사는 참고문헌 [18,19]를 통해 런타임 다운

로드·컴파일 플로우와 함께, 앱 내장 모델 암호화 옵션을 제공해 배포물 노출을 줄일 것을 권고하고 있다.

4.6 엣지·모바일 모델 보호 가이드

사이버 보안 전문 기업인 Cossack Labs[20]에서 엣지 기기와 모바일 앱에서 실행되는 ML 모델 보호에 관련된 다양한 내용을 제공하고 있다. 특히, MITRE ATLAS 매트릭스[21]에 제시된 내용을 토대로 표 1과 같이 위협과 대응 기술을 요약하고 있다.

5. 온디바이스 AI 정보 유출 대응 체계

온디바이스 AI 환경에서는 모델과 데이터가 단말 내부에 직접 저장·실행되기 때문에, 단순 네트워크 보안만으로는 정보 유출을 방어하기 어렵다. 따라서 저장 단계, 실행 단계, 운영 단계 전반에 걸친 다계층(Multi-Layered) 보안 체계가 요구된다.

5.1 모델 저장 단계 보호

저장 단계(At-Rest)에서는 모델 파일 자체를 보호하기 위한 암호화 및 난독화 기술이 중요하다. TFLite, ONNX, Core ML 등 모델 산출물에 대해 AES 기반 암호화와 장치 바인딩(Device Binding)을 적용함으로써 다른 단말에서 무단 실행하거나 복제하는 행위를 제한할 수 있다. 또한, 모델 구조와 연산 그래프를 변형하는 난독화 기법과 워터마킹 기술을 적용하면 모델 역공학 및 IP 탈취에 대한 대응력을 높일 수 있다.

5.2 모델 실행 단계 보호

실행 단계(In-Use)에서는 메모리와 런타임 보호가 핵심 기술로 요구된다. 최근 모델 추출 공격은 복호화 이후 메모리상의 평문 가중치를 대상으로 수행

표 1 MITRE ATLAS 매트릭스에 제시된 위협과 대응 기술

위협	• ML 모델 무단 접근 • ML 서비스 API 취약점 • ML 저장 클라우드 접근 • ML 모델 전송 시 가로채기 및 변조 • ML 모델 역공학에 의한 취약점 발견 • ML 엣지 플랫폼 차이에 의한 보안 공백 • ML 실행 엣지 장치 대상 부채널 공격 • ML 실행 엣지 장치 연결 제약에 따른 보안 업데이트 문제
대응	• ML 모델 암호화 • 클라우드 및 API 보호 • 모바일 애플리케이션 변조 방지 제어 • 사용자/애플리케이션 행동 모니터링 기반 선제적 사기 방지

되므로, Trusted Execution Environment(TEE), Secure Enclave, Virtualization-Based Security(VBS) 등 신뢰실행환경 기반 보호 기술이 중요해지고 있다. 또한 동적 후킹(Frida, Xposed 등), 디버깅, 메모리 덤프를 탐지·차단하는 Anti-Tampering 기술과 실행 무결성 검증 기술도 필요하다.

5.3 입출력 및 운영 단계 보호

입출력 및 운영 단계에서는 프라이버시 보호와 부채널 대응 기술이 중요하다. 로그·임시파일·KV 캐시 최소화 및 메모리 소거(Memory Wiping) 기법을 통해 민감 정보 잔존을 줄일 수 있으며, Differential Privacy, Federated Learning 등의 프라이버시 강화 기술을 통해 학습 데이터 유출 위험을 완화할 수 있다. 또한, GPU/NPU 기반 부채널 공격에 대응하기 위해 메모리 초기화 강화, 접근 제어, 실행 격리 등의 하드웨어 보안 기술도 함께 요구된다.

IV. 결론

온디바이스 AI는 개인정보 보호, 네트워크 비용 절감, 실시간 응답성 향상 등의 장점으로 인해 모바일·엣지·IoT 환경에서 빠르게 확산되고 있으며, 향후 생성형 AI와 LLM의 경량화 기술 발전과 함께 더욱 광범위하게 적용될 것으로 전망된다. 특히 스마트폰, 자율주행차, 웨어러블 기기, 산업용 IoT 등 다양한 환경에서 AI 기능이 단말 내부로 이동함에 따라, 기존 클라우드 중심 AI 아키텍처는 점차 분산형·로컬형 구조로 변화하고 있다.

그러나 온디바이스 AI는 모델과 데이터가 단말 내부에 존재한다는 구조적 특성으로 인해 새로운 정보 유출 위험을 동반한다. 기존에는 클라우드 API

기반의 원격 공격이 주요 위협이었다면, 이제는 단말 내 저장 파일, 메모리, 캐시, 로그, GPU/NPU 실행환경 등이 모두 공격 표면이 되고 있다. 특히 최근 연구들은 온디바이스 모델이 단순 블랙박스 수준이 아니라 화이트박스 수준으로 역공학 가능함을 실증하였으며, 실제 상용 서비스에서도 모델 파일 추출, 메모리 기반 데이터 유출, GPU 로컬 메모리 잔존 데이터 노출 등의 문제가 발생하고 있다.

특히 “온디바이스는 로컬 환경이므로 안전하다”라는 기존 인식은 더 이상 유효하지 않다. 오히려 모델이 사용자 단말에 배포되는 순간부터 정적 분석, 동적 후킹, 메모리 덤프, 부채널 공격 등 다양한 위협에 지속적으로 노출될 가능성이 존재한다. 따라서 향후 온디바이스 AI 시스템은 단순 성능 최적화 중심이 아니라 보안성과 프라이버시 보호를 고려한 Secure-by-Design 접근이 필수적이다.

이를 위해서는 모델 암호화 및 장치 바인딩(Device Binding), 실행 무결성 검증, 보호된 실행 환경(TEE/VBS), 동적 후킹 방지(Anti-Hooking), 메모리 보호, 워터마킹, 라이선싱, 로그 및 캐시 최소화와 같은 다계층 방어 기술이 종합적으로 적용되어야 한다. 또한, AI 모델 자체의 보호뿐만 아니라 런타임·운영체제·하드웨어 가속기·GPU/NPU 메모리 등 전체 실행 생태계를 포괄하는 통합 보안 체계가 요구된다.

향후 연구 측면에서는 경량 환경에서도 적용 가능한 저비용 보호 기술, 모델 추출 탐지 기술, AI 실행 무결성 검증 기술, 부채널 공격 대응 기술, 그리고 생성형 AI 기반 온디바이스 모델의 프라이버시 보호 기술 등에 대한 연구가 지속적으로 필요하다. 아울러 국제 표준화와 산업계 보안 가이드라인 수립 역시 중요한 과제가 될 것으로 판단된다.

참고문헌

- [1] K. Lu et al., "Shift-invariant universal adversarial attacks to avoid deep-learning-based modulation classification," *Int. J. Commun. Syst.*, vol. 36, no. 10, 2023, e5494.
- [2] ITM-predictive GmbH, "Protect IoT-applications from adversarial evasion attacks." [Accessed Aug. 14, 2026.]
- [3] M. Heikkilä, "This new data poisoning tool lets artists fight back against generative AI," *MIT Technol. Rev.*, Oct. 2023.
- [4] A. Souly et al., "Poisoning attacks on LLMs require a near-constant number of poison samples," *arXiv preprint*, 2025. doi: 10.48550/arXiv.2510.07192
- [5] H.A.A.K. Hammoud, "Data poisoning and backdoor attacks: An overview, Part 1," *Medium*, 2021. [Accessed Aug. 14, 2026.]
- [6] H. Yu et al., "CloudLeak: Large-scale deep learning models stealing through adversarial examples," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, (San Diego, CA, USA), Feb. 2020.
- [7] F. Tramèr et al., "Stealing machine learning models via prediction APIs," in *Proc. 25th USENIX Secur. Symp.*, (Austin, TX, USA), Aug. 2016, pp. 601-618.
- [8] M. Fredrikson et al., "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, (Denver, CO, USA), Oct. 2015, pp. 1322-1333.
- [9] N. Carlini et al., "Extracting training data from large language models," in *Proc. 30th USENIX Secur. Symp.*, Aug. 2021, pp. 2633-2650.
- [10] OpenAI, "March 20 ChatGPT outage: Here's what happened," Mar. 2023.
- [11] T. Nayan et al., "SoK: All you need to know about on-device ML model extraction—The gap between research and practice," in *Proc. 33rd USENIX Secur. Symp.*, (Philadelphia, PA, USA), Aug. 2024.
- [12] M. Zhou et al., "Investigating white-box attacks for on-device models," in *Proc. IEEE/ACM 46th Int. Conf. Softw. Eng. (ICSE)*, (Lisbon, Portugal), Apr. 2024, pp. 1-12.
- [13] Z. Liu et al., "Decompiling x86 deep neural network executables," in *Proc. 32nd USENIX Secur. Symp.*, (Anaheim, CA, USA), Aug. 2023, pp. 7395-7412.
- [14] M. Burgess, "Microsoft's Recall feature is even more hackable than you thought," *WIRED*, 2024.
- [15] T. Sorensen and H. Khlaaf, "LeftoverLocals: Listening to LLM responses through leaked GPU local memory," *arXiv preprint*, 2024. doi: 10.48550/arXiv.2401.16603
- [16] M. Mauvisseau and V. Guyomard, "Google Photos' AI models: The secret sauce that can be stolen," *Skyld*, Mar. 2025.
- [17] A. Verma, "Encryption and real time decryption for protecting machine learning models in Android applications," *arXiv preprint*, 2021. doi: 10.48550/arXiv.2109.02270
- [18] Apple, "Downloading and compiling a model on the user's device," *Core ML Documentation*, accessed Aug. 14, 2026.
- [19] Apple, "Encrypting a model in your app," *Core ML Documentation*, accessed Aug. 14, 2026.
- [20] Cossack Labs, "Protecting ML models running on edge devices and mobile apps." [Accessed Aug. 14, 2026.]
- [21] MITRE, "MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems." [Accessed Aug. 14, 2026.]