SW 공급망 보안을 위한 AI 패키지 환각 검증 기법

홍혜민 [1], 임희은 [2], 안지호 [3] 김연진 [4], 이일구 [5]
[1] 서울여자대학교 정보보호학과 학부생
[2] 덕성여자대학교 사이버보안전공 학부생
[3] 한라대학교 IT 소프트웨어학과 학부생
[4] 성신여자대학교 융합보안공학과 석사 과정
[5] 성신여자대학교 융합보안공학과 교수

hhyem@swu.ac.kr, lhe3051@duksung.ac.kr, wlgh010710@naver.com, 220246046@sungshin.ac.kr, iglee@sungshin.ac.kr

AI-Based Package Hallucination Detection Techniques for Software Supply Chain Security

Hye-min Hong[1], Hui-eun Lim[2], Ji-ho Ahn[3], Yeon-Jin Kim[4], Il-Gu Lee[4]
[1]Dept. of Information Security, Seoul Women's University
[2]Dept. of Cyber Security, Duksung Women's University
[3]Dept. of IT Software, Halla University
[4]Dept. of Convergence Security Engineering, Sungshin Women's University

요 약

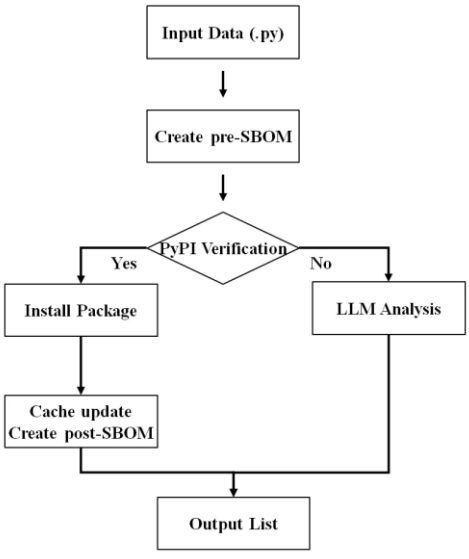
대규모 언어 모델(Large Language Model, LLM)의 환각(Hallucination) 문제는 보안상 심각한 위협으로 이어질 수 있다. 본 연구에서는 이러한 위협을 완화하기 위해 파이썬 공식 저장소인 PyPI 와 캐시 기반 검증을 결합하고, 추가적으로 LLM 질의를 활용하여 구체화된 환각 패키지 검증 구조를 제안하였다. 제안된 구조는 탐지 평가에서 97.8%의 정확도를 달성하였으며 이는 종래 연구 대비 API 호출 횟수와 속도 측면에서 개선된 결과이다. 검증된 패키지는 SBOM(Software Bill of Materials) 형태로 기록하여 추후 체계적인 관리가 가능하도록 하였다.

1. 서론

대규모 언어 모델(Large Language Model, LLM)은 코드 자동 생성과 오류 수정에 높은 생산성을 제공하지만, 환각으로 인한 보안 위협이 발생할 수 있다[1]. 특히 잘못된 패키지명을 악용할 경우, 악성 패키지가 설치되어 기업 내부 시스템의 안정성과 보안을 위협한다. 최근 멀티 에이전트를 사용하여 환각을 검증하려는 연구가 진행되고 있으나, LLM API 호출 빈도가 높아 비용과 지연이 과도하게 발생하는 한계가 있다[2]. 이에 본 연구에서는 LLM 이 생성한 패키지를 자동으로 검증하는 프레임워크를 제안한다. 악성 패키지 설치로부터 발생할 수 있는 위협을 최소화하고, 신뢰할 수 있는 LLM 활용과 안전한 소프트웨어 개발 생태계 구축을 목표로 한다.

2. 환각 패키지 검증 방법

본 연구에서 제안하는 Package Hallucination Verification(PHV) 구조는 (그림 1)과 같이 동작한다. 먼저, 코드의 import 구문에서 추출한 패키지명을 캐시와 대조한다. 캐시는 검증이 완료된 패키지 정보를 저장하여 Python 오픈소스 패키지 저장소 (PyPI)[3] 조회 및 LLM 질의의 불필요한 반복을 방지하며, API 호출 횟수의 감소에 기여한다.



(그림 1) 패키지 환각 검증 구조.

만약 캐시에 해당 패키지 정보가 존재하지 않는 경우, PyPI 조회를 수행한다. PyPI 에서 조회되지 않는 패키지는 환각으로 판별하며, 조회 결과와 함께 릴리즈 이력이 확인되는 경우 정상 패키지로 분류한다. 반면 패키지명이 조회되지만 릴리즈 이력이 존재하지 않는 경우 환각 의심 패키지로 분류하고, LLM 을 활용하여 전체 코드를 분석한다. 최종적으로 LLM 의 분석 결과를 바탕으로 패키지의 환각 여부를 결정한다.

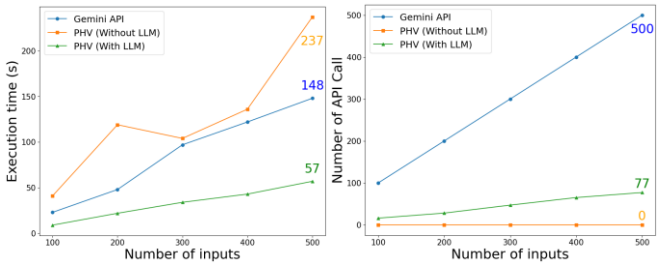
SBOM (Software Bill of Materials)은 분석 시점의 패키지, 버전, 의존성 등의 정보를 보존함으로써 실행 환경을 증명하며, 이를 통해 재현성과 검증 과정의 신뢰성을 확보한다. 본 연구는 입력으로 제공되는 각 코드에 대하여 해당 시점의 SBOM 을 자동으로 생성해 실행 환경을 기록한다. 새로운 패키지가 설치되는 경우 post-SBOM 이 생성되며, 최종적으로 결과 보고서는 pre-SBOM 과 post-SBOM 의 차이를 비교 및 분석한 결과를 제공한다.

3. 성능 평가

평가에 활용한 데이터셋은 정상 코드 400 개, 환각 패키지가 포함된 코드 100 개로 구성된다. 환각 코드는 실존하는 패키지명을 변형하여 제작하였다.

PHV 의 성능 검증을 위해 Gemini API, LLM 검증 부분이 제외된 PHV, PHV 를 대상으로 평가를 진행하였다. Gemini API 는 PyPI 에 패키지 존재 여부에 대한 질의를 수행하여, 질의 결과가 존재할 경우 정상으로, 그렇지 않을 경우 LLM 으로 전체 코드를 분석하여 환각 여부를 판별한다. 반면 전체 코드에 대한 LLM 분석을 수행하지 않는 PHV 는 패키지명이 PyPI 에 존재하는지 조회하고, 조회 여부에 따라 정상과 환각으로 분류한다. 이후 릴리즈 이력이 확인되면 정상, 확인되지 않을 경우 환각 의심 패키지로 처리한다.

성능 비교는 API 호출 횟수, 데이터 개수에 따른 처리 시간, 그리고 탐지율을 기준으로 수행하였다. 데이터 개수에 따른 API 호출 횟수와 처리 시간 비교 결과는 (그림 2)에 제시하며, 탐지율 및 환각 의심 패키지 수에 대한 비교 결과는 <표 1>에 나타낸다.



(a) 실행 시간 (b) API 호출 횟수

(그림 2) 데이터 개수에 따른 성능 비교.

<표 1> 탐지율 및 환각 의심 패키지 수 비교.

	Gemini API	PHV (Without LLM)	PHV (With LLM)
Detection Rate	97.76%	45.78%	97.8%
Number of Suspected Hallucinated Package	10	190	0

LLM 분석이 제외된 PHV 에서는 환각 의심 패키지의 비율이 전체 데이터셋의 약 38% 이상을 차지하였으나, LLM 분석을 추가한 결과 해당 비율이 약 15%로 감소하였다. 이는 LLM 이 코드 전체 맥락을 기반으로 패키지 존재 여부를 정밀하게 검증하기 때문에 단순 패키지 조회만으로는 환각으로 분류되던 사례들이 정상으로 판별된 결과로 분석된다. 탐지율 측면에서는 Gemini API 와 PHV 모두 90%대로 우수한 성능을 보였으며, 특히 PHV 는 LLM 호출 횟수를 약 80% 감소시켜 API 호출 비용을 효과적으로 절감하였다. 또한 데이터 개수가 증가하면서 캐시에 검증된 패키지 목록이 저장됨에 따라 실행 시간이 빨라진다는 것을 증명하였다.

4. 결론

본 연구는 API 호출 횟수 증가 문제를 해결하기 위해 PyPI 기반 환각 패키지 검증과 캐시를 활용한 환각 패키지 검증 프레임워크를 제안하였다. 이를 통해 환각 의심 패키지 비율을 100% 감소시키고, LLM 호출 횟수를 80% 줄여 비용과 시간을 개선하였다.

향후에는 코드에서 발생하는 환각까지 검증할 수 있는 기법으로 확장함으로써, 보다 포괄적이고 안정적인 환각 검증 프레임워크를 구축할 예정이다.

Acknowledgments

본 논문은 2025 년도 산업통상자원부 및 한국산업기술진흥원의 산업혁신인재성장지원사업 (RS-2024-00415520)과 과학기술정보통신부 및 정보통신기획평가원의 ICT 혁신인재 4.0 사업의 연구결과로 수행되었음 (No. IITP-2022-RS-2022-00156310)

참고문헌

[1] Spracklen, Joseph, Raveen Wijewickrama, A.H.M. Nazmus Sakib, Anindya Maiti, Bimal Viswanath, and Murtuza Jadliwala. "We have a package for you! A comprehensive analysis of package hallucinations by code generating LLMs." In USENIX Security Symposium. 2025.

[2] Sun, Xiaoxi; Li, Jinpeng; Zhong, Yan; Zhao, Dongyan; Yan, Rui. *Towards Detecting LLMs Hallucination via Markov Chain-based Multi-agent Debate Framework*. arXiv preprint arXiv:2406.03075, 2024.

[3] Python Software Foundation, “PyPI,” Python Package Index. [Online]. Available: https://pypi.org/. [Accessed: Sep. 30, 2025].