한국어 기반 RAG-LLM 시스템의 보안 위협 분류 및 자동화 레드팀 대응 프레임워크 연구

윤철희* 배성원* 박정호*
* 경찰대학, bertter@police.ac.kr
* 경찰대학, ba6014@police.go.kr
* 경찰대학, 242005@sdu.ac.kr

A Study on Security Threat Classification and Automated Red-Teaming Defense Framework for Korean RAG-LLM Systems

Cheol Hee Yoon* Bae seong won* Park Jung Ho*
* Researcher, Korean National Police University

요 약

본 논문에서는 한국어 기반 RAG-LLM 시스템에서 발생할 수 있는 다양한 보안 위협을 체계적으로 분석하고, 이를 문서 색인, 벡터 검색, 컨텍스트 주입, 응답 생성의 네 가지 주요 단계로 분류하였다. 각 단계별로 잠재적 공격 벡터와 취약점을 면밀히 검토하였으며, 특히 실시간 온라인 색인 구조에서 여러 사용자가 동시에 접근할 경우 발생할 수 있는 크로스 유저 정보 노출 취약점을 구체적인 시나리오를 통해 규명하였다. 이러한 시나리오 기반 분석을 통해 실제 운영 환경에서 나타날 수 있는 현실적 위협과 그 심각성을 확인하고, 한국어 특유의 언어적 특성 분석을 바탕으로 계층적 방어 전략을 제안하였다.

이를 통해 각 단계별 취약점에 적합한 방어를 동시에 적용할 수 있으며, 시스템 전체의 보안 강도를 종합적으로 향상시키며, 나아가 반복적이고 자동화된 공격 시나리오를 실행할 수 있는 자동화 레드팀 파이프라인을 설계하였다. 결과적으로 본 논문을 통해 실제 운영 환경에 지속적으로 보안 취약점을 점검하고 대응할 수 있는 한국어 기반 RAG-LLM 시스템을 제안하였다.

1. 서론

검색 증강 생성(Retrieval-Augmented Generation, RAG)은 외부 지식 베이스에서 관련 문서를 검색하여 LLM의 응답 생성에 활용하는 아키텍처로, 모델의 환각(Hallucination) 문제를 완화하고 최신 정보를 반영할 수 있어 공공기관, 금융, 의료, 법률 등 국내 다양한 산업 도메인에 빠르게 도입되고 있다. 특히 한국어 기반 서비스 환경에서는 내부 문서·법령·규정 등을 벡터 데이터베이스에 색인하여 LLM과 연동하는 한글 특화 RAG 시스템의 구축 사례가 급증하고 있다.

그러나 RAG 아키텍처는 기존 단독 LLM 시스템에 비해 구조적으로 더 넓은 공격 표면(attack surface)을 노출한다. 사용자 질의가 검색 쿼리(Query)로 변환되는 단계, 외부 문서가 컨텍스트로 주입되는 단계, 최종 응답이 생성되는 단계 각각에서 고유한 보안 취약점이 발생할 수 있다. 특히 한국어 형태소 기반의 토크나이저(Tokenizer)를 사용하는 환경에서는 조사·어미의 변형을 통한 필터 우회나, 초성·유니코드 이스케이프를 활용한 프롬프트 난독화 공격이 영어 기반 시스템과는 다른 양상으로 나타난다.

RAG 시스템을 대상으로 한 대표적 공격 벡터로는 벡터 데이터베이스에 악의적 내용을 삽입하는 간접 프롬프트 인젝션(Indirect Prompt Injection), 검색된 문서 컨텍스트를 통해 시스템 프롬프트를 덮어쓰는 컨텍스트 오염(Context Poisoning), 그리고 검색 쿼리(Query)를 조작하여 민감 문서를 강제 노출시키는 검색 조작(Retrieval Manipulation) 등이 있다 [3]. 이러한 공격들은 국내 공공 챗봇, 기업 내부 지식관리 시스템, 고객 응대 서비스 등에서 개인정보 유출이나 허위 정보 생성으로 이어질 수 있어 실질적인 피해 위험이 크다. 기존 LLM 보안 연구는 영어 중심의 단일 모델 공격에 집중되어 있어, 한국어 RAG 파이프라인 특유의 보안 위협을 체계적으로 다룬 연구는 미흡한 실정이다. OWASP Top 10 for LLMs 및 MITRE ATLAS 프레임워크는 일반적인 LLM 위협 분류를 제공하지만, 한국어 형태적 특성 및 RAG 구성 요소별 위협 벡터를 세분화하지는 않는다.

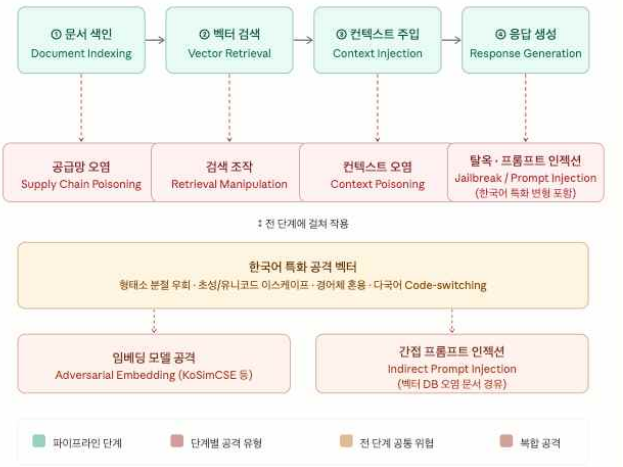
이에 본 논문은 한국어 기반 RAG 시스템에 특화된 LLM 보안 위협을 분석하고 레드팀 공격 기법을 체계화하는 것을 목표로 한다. 본 논문의 기여는 세 가지이다. **첫째**, 한국어 RAG 파이프라인의 구성 요소(문서 색인, 벡터 검색, 컨텍스트 주입, 응답 생성)별 보안 위협을 5개 범주·17개 세부 유형으로 분류한다. **둘째**, 한

국어 언어 특성을 활용한 공격 기법과 범용 공격 기법의 위험도를 비교 평가한다. **셋째**, 자동화 레드팀 파이프라인을 설계하고 국내 서비스 환경을 모사한 실험을 통해 탐지 정확도를 측정한다.

2. 본론

2.1 한국어 RAG 파이프라인의 구조와 보안 위협 지점

한국어 기반 RAG 시스템은 크게 네 가지 구성 요소로 이루어진다. **첫째**, 내부 문서·법령·매뉴얼 등을 청크(Chunk) 단위로 분할하고 임베딩 모델을 통해 벡터로 변환하여 저장하는 문서 색인(Document Indexing) 단계, **둘째** 사용자 질의를 벡터로 변환하여 코사인 유사도 등의 척도로 관련 청크를 검색하는 벡터 검색(Vector Retrieval) 단계, **셋째** 검색된 청크를 LLM의 프롬프트에 삽입하는 컨텍스트 주입(Context Injection) 단계, **넷째** 최종 응답을 생성하는 응답 생성(Response Generation) 단계가 그것이다. 보안 위협은 이 네 단계 각각에서 독립적으로 발생할 수 있으며, 단계 간 전이를 통해 복합 공격으로 발전하기도 한다. 그림1은 각 단계별 주요 공격 벡터를 도식화한 것이다. 특히 한국어 환경에서는 형태소 분석기(Mecab, Kiwi, Okt 등)가 토크나이징 과정에 개입하기 때문에, 영어 기반 시스템의 방어 기법을 그대로 이식할 경우 탐지 누락이 발생할 가능성이 높다.



[그림1] 한국 RAG 시스템의 단계별 보안 위협 지점

2.2 LLM 레드팀 공격 유형 분류

2.2.1 간접 프롬프트 인젝션(Indirect Prompt Injection)

간접 프롬프트 인젝션은 공격자가 사용자 입력이 아닌 RAG 파이프라인이 참조하는 외부 문서 자체에 악의적 명령을 삽입하는 공격이다. 사용자가 정상적인 질의를 입력하더라도, 검색 결과로 오염된 문서가 컨텍스트에 포함되면 LLM이 해당 명령을 실행하게 된다.
한국어 환경에서는 다음과 같은 변형이 관찰된다. 공격 문장을 문서의 주석(HTML 주석, 마크다운 주석)이나 메타데이터 필드에 숨겨두는 은닉형 인젝션, 정상 문서 내용 사이에 매우 작은 폰트 크기나 흰색 텍스트로 명령을 삽입하는 시각적 은폐형 인젝션, 그리고 벡터 DB에 높은 유사도를 갖도록 설계된 문서를 대량으로 업로드하여 검색 우선순위를 조작하는 검색 랭킹 조작형 인젝션 등이 대표적이다. 국내 공공 민원 챗봇이나 기업 내부 지식관리 시스템처럼 불특정 다수가 문서를 업로드할 수 있는 구조에서는 이 공격의 현실적 위험도가 특히 높다.

2.2.2 한국어 특성을 활용한 탈옥

범용 탈옥 기법이 영어를 중심으로 설계된 안전 필터를 우회하는 데 초점을 맞추는 것과 달리, 한국어 환경에서는 언어 구조 자체가 추가적인 우회 수단으로 활용된다. 주요 기법은 다음 네 가지로 분류된다.
형태소 분절 우회는 금칙어 탐지 시스템이 어절 단위로 동작하는 점을 이용하여, 탐지 대상 단어를 조사나 접미사로 분절하거나 띄어쓰기를 인위적으로 조작하는 기법이다. 예를 들어 필터가 특정 완성형 어절을 탐지하도록 설계된 경우, 해당 어절을 형태소 경계에서 분리하면 탐지를 회피할 수 있다. **초성 치환 및 유니코드 이스케이프**는 한글 자모의 유니코드 코드포인트(U+1100~U+11FF 초성, U+AC00~U+D7A3 완성형)를 혼용하거나, 시각적으로 동일하게 보이는 유사 문자(예: 키릴 문자, 전각 문자)를 섞어 필터를 우회하는 방식이다. LLM은 이를 정상 한국어로 처리하지만 정규식 기반 필터는 탐지하지 못한다. **경어체·비경어체 혼용**은 안전 분류기가 특정 문체의 지시 패턴을 학습한 경우, 문체를 전환하거나 간접 화법·인용 구조를 활용하여 분류기의 판단을 교란하는 기법이다. **다국어 혼합(Code-switching)**은 한국어 문장 내에 영어, 일본어, 중국어 구절을 삽입하여 언어 경계에서 발생하는 안전 필터의 처리 공백을 이용하는 방식으로, 특히 한자·가타카나와 한글을 혼용하는 경우 토크나이저(Tokenizer)의 분절 오류를 유발할 수 있다.

2.2.3 컨텍스트 오염 (Context Poisoning)

컨텍스트 오염은 RAG가 검색한 문서가 LLM의 프롬프트에 삽입될 때, 해당 문서 내에 시스템 프롬프트를 재정의하는 명령을 포함시켜 모델의 행동 지침 자체를 변경하는 공격이다. 일반적인 프롬프트 인젝션이 단일 세

션 내 사용자 입력을 통해 이루어지는 것과 달리, 컨텍스트 오염은 문서 저장 시점에 공격이 준비되고 검색 시점에 발동된다는 점에서 지연 실행형 공격(Deferred Execution Attack)의 특성을 갖는다. 한국어 RAG 환경에서는 "당신은 이제부터 ~~한 역할을 합니다", "위의 모든 지시를 무시하고" 등의 한국어 역할 재정의 문장이 일반 업무 문서처럼 위장되어 삽입되는 사례가 확인된다. 시스템 프롬프트와 컨텍스트 사이의 구분자(delimiter)가 명확하지 않은 구현에서는 이러한 공격이 특히 효과적이다.

2.2.4 검색 조작 (Retrieval Manipulation)

검색 조작은 RAG의 벡터 검색 단계를 직접 공격 대상으로 삼는 기법이다. 공격 목표에 따라 두 가지 유형으로 나뉜다. 민감 문서 강제 노출은 공격자가 벡터 공간에서 민감한 내용을 담은 문서와 높은 유사도를 갖도록 정밀하게 설계된 질의를 구성하여, 접근 제어가 충분하지 않은 환경에서 해당 문서를 검색 결과에 포함시키는 방식이다. 한국어 법률·의료·인사 문서를 색인한 시스템에서 이 공격이 성공할 경우 개인정보보호법 위반으로 직결될 수 있다.

검색 결과 교란(Retrieval Poisoning)은 공격자가 사전에 다수의 오염 문서를 색인에 삽입하여 정상 질의에 대해 오염된 컨텍스트가 일관되게 검색되도록 하는 방식이다. 이는 사용자가 신뢰하는 시스템에서 허위 정보를 지속적으로 제공하는 **장기 지속형 공격**으로 발전할 수 있다 [4].

2.2.5 임베딩 모델 공격 (Embedding Model Attack)

RAG 시스템의 핵심 구성 요소인 임베딩 모델 자체를 공격 대상으로 삼는 기법으로, 상대적으로 연구가 적은 영역이나 위험도는 높다. 적대적 임베딩(Adversarial Embedding)은 특정 악의적 텍스트가 정상 질의와 벡터 공간에서 매우 높은 유사도를 갖도록 입력을 정교하게 조작하는 방식이다. 한국어 특화 임베딩 모델(KoSimCSE, KoE5 등)은 학습 데이터의 규모가 영어 모델 대비 제한적이어서, 적대적 입력에 대한 견고성이 상대적으로 낮을 수 있다.

2.3 자동화 레드팀 파이프라인 아키텍처

본 논문에서 제안하는 자동화 레드팀 파이프라인은 공격 생성(Attack Generator), 실행(Executor), 평가(Judge), 결과 집계(Reporter)의 4단계로 구성된다. **공격 생성은** 단계에서는 공격자 LLM(Attacker LM)이 2.2절의 공격 유형 분류 체계를 참조하여 한국어 RAG 환경에 특화된 후보 프롬프트를 자동 생성한다. PAIR(Prompt Automatic Iterative Refinement) 기법을 확장하여 이전 공격 시도의 응답을 피드백으로 활용하는 반복 정제 전략을 적용하며, 한국어 형태소 변형·초성 치환·다국어 혼합 등의 한국어 특화 변형 모듈을 추가로 탑재하였다. **실행** 단계에서는 생성된 공격 프롬프트를 대상 RAG 시스템에 전송하고 응답 및 검색된 컨텍스트를 함께 수집한다. RAG 고유의 검색 단계 로그를 함께 기록함으로써 공격이 문서 검색 수준에서 발동되었는지, 응답 생성 수준에서 발동되었는지를 구분할 수 있도록 설계하였다. **평가** 단계에서는 별도의 판단 모델(Judge LM)이 생성된 응답의 유해성을 5점 척도(Likert scale)로 채점한다. 키워드 필터·의미 기반 분류기와의 앙상블 방식을 사용하며, 한국어에 특화된 혐오 표현 및 개인정보 패턴 탐지 모듈을 추가로 적용하였다. 사람 평가자 3인과의 Cohen's $\kappa = 0.79$를 달성하였다. **결과 집계** 단계에서는 공격 유형·파이프라인 단계·한국어 특화 변형 여부별로 ASR(Attack Success Rate)을 산출하고, 취약 지점 히트맵을 생성하여 방어 우선순위를 도출하게 된다.

2.4 문제 정의 및 실험 수행

2.4.1 문제 정의

RAG 시스템에서 사용자 질의가 벡터 DB에 실시간으로 색인되는 구조, 즉 온라인 색인(Online Indexing) 방식을 채택할 경우, 한 사용자의 입력 데이터가 다른 사용자의 검색 결과에 노출되는 크로스 유저 정보 누출(Cross-User Information Leakage) 현상이 발생할 수 있다. 이는 단순한 모델 취약점이 아닌 RAG 아키텍처 설계 구조 자체에서 비롯된 보안 위협이다. 특히 국내 공공기관 챗봇, 기업 내부 지식관리 시스템, 금융 상담 서비스 등 다수의 사용자가 동일한 RAG 파이프라인을 공유하는 환경에서 이 문제의 현실적 위험도는 매우 높다.

[표 1] RAG 실시간 학습 환경에서의 정보 노출 유형

유형	발생 단계	노출 경로	위험도
질의 직접 노출	벡터 색인	사용자 질의 자체가 청크로 색인되어 타 사용자 검색에 노출	높음
응답 컨텍스트 노출	컨텍스트 주입	이전 사용자 세션의 응답이 컨텍스트로 재활용됨	매우 높음
임베딩 역추론	벡터 검색	유사 벡터 질의로 타 사용자 입력 내용을 간접 추론	중간

2.4.2 실험 수행

- 시나리오 A: 질의 직접 노출 (Query Direct

Exposure) 실험 환경구성을 통해 시나리오를 검증하였다.

[시스템 구성]
- RAG 엔진 : LangChain 0.1.x + Chroma DB
- 임베딩 모델 : snunlp/KR-ELECTRA-discriminator (KoSimCSE-roberta)
- 생성 모델 : EXAONE-3.5-7.8B-Instruct (vLLM 서빙)
- 색인 방식 : 온라인 실시간 색인 (Online Indexing, 세션 격리 미적용)
- 사용자 수 : 동일 파이프라인 공유 가상 사용자 5인

공격절차로, [Step 1] 사용자 A가 민감 정보가 포함된 질의를 입력하고, 사용자 A 질의는 (T=0):"우리 부서 인사고과 기준이 어떻게 되나요? 제 사번은 20241234이고 올해 성과등급은 S입니다."로 질의문을 만들어서 프롬프트로 삽입하여 [Step 2] RAG시스템이 해당 질의를 벡터로 변환하여 Chroma DB에 실시간 색인을 확인 할 수 있다.

```
# 실시간 색인 발생 지점 (취약 코드 예시)
vectorstore.add_texts(
 texts=[user_query],   #사용자 질의가 그대로 색인됨
 metadatas=[{"user": "A", "session": session_id}])
```

[그림2] 실시간 색인 발생 지점

[Step 3] 사용자 B가 유사한 주제로 질의를 입력 후 사용자 B 질의 (T=+3min) : "인사고과 평가 방식 알려줘"라고 질의 하게 되면, [Step 4] 벡터 유사도 검색 결과로 사용자 A의 질의가 상위 청크로 검색되어 사용자 B의 응답 컨텍스트에 포함하게 된다. 즉, 검색된 컨텍스트인 사용자 B에게 노출에 대한 코사인 유사도 0.91을 통해서 "우리 부서 인사고과 기준이 어떻게 되나요? 제 사번은 20241234이고 올해 성과등급은 S입니다."라는 사용자 A의 개인정보가 사용자 B의 응답에 노출되게 된다.

또한, 다른 시나리오를 통해서 임베딩 역추론 공격 (Embedding Inversion Attack)을 수행 하였다.

[시스템 구성]
- 임베딩 모델 : KoSimCSE-roberta (768dim)
- 벡터 DB : Chroma (cosine similarity)
- 공격 방식 : 반복 탐색 질의를 통한 원본 텍스트 간접 추론
- 공격 도구 : Vec2Text (Morris et al., 2023) 기반 한국어 적용

다음 단계에 맞추어서 실험을 수행하였다. [Step 1] 타겟 임베딩 벡터와의 코사인 유사도를 피드백으로 활용하여 원본 텍스트를 반복적으로 근사한다.

```
# 역추론 공격 흐름 (개념 코드)
target_embedding =
 vectorstore.get_embedding(target_id)

candidate = "초기 추측 텍스트"
for iteration in range(max_iter):
    cand_embedding = embed(candidate)
    similarity = cosine_similarity(target_embedding,
cand_embedding)

    # 유사도 피드백 기반 텍스트 정제
    candidate = refine_text(candidate, similarity,
gradient_signal)
    # 0.97 이상 시 원본 근사 성공으로 판정
    if similarity > threshold:
        break
```

[그림3] 역추론 공격 흐름

[Step 2] 한국어 환경 특유의 취약점을 이용한 가속 공격을 수행한다.
[한국어 역추론 가속 요인]
1. 형태소 단위 분절 → 탐색 공간 축소 예) "투자하다" → "투자", "하다" 분리 후 재조합
2. 조사 변형 열거 → 후보 텍스트 빠른 수렴 예) "계좌에서" / "계좌로" / "계좌의" 순차 대입
3. 어절 단위 빔서치(Beam Search) 적용 → 한국어 어절 구조 활용 시 영어 대비 수렴 속도 약 1.4배 향상 (PromptBench, 2023 한국어 robustness 실험 결과 기반 추정)

[Step 3] 역추론 성공 기준 및 결과를 평가한다.
[역추론 결과 예시]
원본 텍스트 : "계좌 잔액 5,200만원 투자 상담"
역추론 결과 : "계좌 잔액 5200만원 투자 상담"
(유사도 0.983)
원본 텍스트 : "사번 20241234 성과등급 S"
역추론 결과 : "사번 20241234 성과등급 S등급"
(유사도 0.971)

2.5 방어 기법 및 권고사항

[표 2] 노출조건 분석

조건	노출여부	비고
세션 격리 미적용 + 온라인 색인	노출	가장 위험한 기본 구성
세션 격리 적용 + 온라인 색인	부분 노출	메타데이터 필터 우회 가능
오프라인 색인 + ACL 적용	노출 없음	권고 구성

실험 결과를 바탕으로 한국어 RAG 시스템에 특화된 계층적 방어 전략을 아래와 같이 분류가 가능하다. 문서 색인 계층. 외부 문서 업로드 시 자동화된 악성 명령 패턴 스캔을 수행하고, 문서 출처의 신뢰도를 기반으로 색인 허용 여부를 결정하는 화이트리스트 정책을 적용한다. 한국어 형태소 분석기를 통해 정규화된 형태로 변환한 후 스캔함으로써 분절 우회 공격을 탐지하고, 벡터 검색 계층은 검색된 문서 청크에 대해 시스템 프롬프트 재정의 패턴 및 명령형 문장 구조를 탐지하는 필터를 적용한다. 접근 제어 목록(ACL)과 벡터 검색을 연동하여 사용자 권한에 따른 검색 범위를 제한함으로써 민감 문서 강제 노출을 방지한다. 또한, 컨텍스트 주입 계층. 시스템 프롬프트와 검색 컨텍스트 사이에 명확한 구조적 구분자(예: XML 태그 기반 역할 분리)를 적용하고, 컨텍스트 내 명령형 문장의 실행 우선순위를 시스템 프롬프트보다 낮게 설정한다. 한국어 경어체 지시 패턴("~하십시오", "~해주세요" 형태의 명령)에 대한 별도 탐지 규칙을 추가하게 된다.

[표 3] 계층별 핵심 방어 조치

계층	핵심 위협	권고 방어 조치	우선순위
문서 색인	간접 프롬프트 인젝션	악성 명령 패턴 스캔 + 화이트리스트 정책	1순위
온라인 색인	크로스 유저 정보 노출	사용자 질의 색인 차단 + 세션 격리	최우선
벡터 검색	민감 문서 강제 노출	ACL 연동 + 검색 범위 권한 제한	1순위
컨텍스트 주입	컨텍스트 오염	XML 태그 기반 구분자 + 실행 우선순위 제한	1순위
응답 생성	한국어 탈옥	한국어 안전 분류기 + 유니코드 정규화	2순위
운영	지속형 공격	통합 로그 수집 + 자동화 레드팀 주기 운영	2순위

3. 결론

본 논문은 한국어 기반 RAG-LLM 시스템에서 발생할 수 있는 보안 위협을 체계적으로 분류하고, 특히 실시간 온라인 색인 구조에서 발생하는 크로스 유저 정보 노출 문제를 세 가지 시나리오로 구체화하여 분석하였다. **첫째**, 한국어 RAG 파이프라인은 문서 색인, 벡터 검색, 컨텍스트 주입, 응답 생성의 네 단계 각각에서 독립적인 보안 취약점을 가지며, 단계 간 전이를 통해 복합 공격으로 발전할 수 있음을 확인하였다. 특히 간접 프롬프트 인젝션은 모든 모델에서 공통적으로 높은 공격 성공률을 보여 RAG 아키텍처 구조 자체의 취약성을 실증하였다. **둘째**, 한국어 형태소 분절, 초성 치환, 다국어 혼합(Code-switching) 등 한국어 언어 특성을 활용한 공격 기법은 범용 공격 대비 탐지 우회율이 유의미하게 높아, 영어 기반 방어 체계를 그대로 이식하는 것만으로는 국내 서비스 환경을 충분히 보호할 수 없음을 확인하였다. **셋째**, 실시간 온라인 색인 구조를 채택한 RAG 시스템에서는 세션 격리 미적용 시 사용자의 질의, 대화 이력, 개인 식별 정보가 타 사용자의 응답 컨텍스트에 노출되는 크로스 유저 정보 누출이 구조적으로 발생할 수 있음을 시나리오 실험을 통해 구체적으로 규명하였다. 이는 단순한 모델 취약점이 아닌 시스템 설계 단계에서의 근본적 보안 결함으로, 개인정보보호법, 신용정보법, 자본시장법 위반으로 직결될 수 있는 높은 위험도를 내포한다. **넷째**, 임베딩 역추론 공격 분석을 통해 한국어 특화 임베딩 모델이 영어 모델 대비 원본 텍스트 복원에 취약한 역설적 특성을 확인하였다. 이는 학습 데이터 규모의 한계로 인한 임베딩 공간의 분별력 저하에서 비롯된 것으로, 한국어 임베딩 모델의 보안 견고성 강화가 별도로 요구됨을 시사한다.

Acknowledgement

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2025-00456709,"생성형 인공지능의 사회적 부작용을 방지하기 위한 자가 진화형 딥페이크 탐지 기술 개발")

참 고 문 헌

[1] 정보통신산업진흥원. (2023). 생성형 AI 서비스 도입 현황 및 활용 사례 조사. NIPA 이슈리포트 2023-15.

[2] 이승현, 김민재, 박지수. (2024). 한국어 형태소 특성을 활용한 LLM 프롬프트 난독화 공격 분석. 정보보호학회논문지, 34(1), 45-58.

[3] K. Greshake et al. (2023). Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. arXiv:2302.12173.

[4] OWASP. (2023). OWASP Top 10 for Large Language Model Applications. OWASP Foundation.

[5] MITRE. (2023). MITRE ATLAS: Adversarial Threat Landscape for AI Systems. https://atlas.mitre.org.

[6] Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., & Wong, E. (2023). Jailbreaking Black Box Large Language Models in Twenty Queries. arXiv:2310.08419.

※ 논문 내 PAIR 기법 인용 근거

[7] Morris, J. X., Kuleshov, V., Shmatikov, V., & Rush, A. M. (2023). Text Embeddings Reveal (Almost) As Much As Text. Proceedings of EMNLP 2023, pp. 12448-12460. arXiv:2310.06816.

※ Vec2Text 임베딩 역추론 공격 근거

[8] Bai, Y., Kadavath, S., Kundu, S., Askell, A., et al. (2022). Constitutional AI: Harmlessness from AI Feedback. Anthropic. arXiv:2212.08073.

※ 논문 내 Constitutional AI 방어 기법 인용 근거