Journal of the KNST

2025; Vol.8, No.1; pp.23-29

ISSN: 2635-4926 https://doi.org/10.31818/JKNST.2025.3.8.1.23

Check for updates

Received: 2025/02/28
Revised: 2025/03/08
Accepted: 2025/03/26
Published: 2025/03/31

*Corresponding Author:
Seung-Woo Kim
E-mail: tearmoa@naver.com

Abstract

대규모 언어 모델(LLM)은 자연어 처리 기술의 발전을 선도하고 있지만, 동시에 프롬프트 인젝션, 데이터 유출, 적대적 예시 공격과 같은 보안 위협을 초래하고 있다. DeepSeek, GeDAI, ChatGPT를 대상으로 해킹 취약점을 분석하고, 각 모델이 직면한 보안 위협과 이를 방어하기 위한 기술적 대응책을 제시한다. 특히, GeDAI는 2025년 자유의 방패(FS) 연합연습에서 군사 작전 지원을 위해 시범 운용될 예정이며, 군사적 활용 가능성과 보안 강화를 위한 추가적인 연구가 필요하다. 본 연구는 M&S 기반 시뮬레이션 환경에서 Microsoft사의 STRIDE 위협 모델링 기법을 적용하여 LLM의 잠재적 위협을 식별하고, 이를 바탕으로 보안 코딩 및 시스템 설계 측면에서의 대응 방안을 도출함으로써 안전한 LLM 구축 방향성을 제시한다.

Large Language Models (LLMs) are leading the advancement of natural language processing technology, but they also introduce security threats such as prompt injection, data leakage, and adversarial example attacks. This study analyzes hacking vulnerabilities in DeepSeek, GeDAI, and ChatGPT, identifying the security risks each model faces and proposing technical countermeasures. Notably, GeDAI is scheduled for pilot operation during the 2025 Freedom Shield (FS) joint military exercise to support military operations, highlighting the need for further research on its military applicability and security enhancement. This research applies Microsoft's STRIDE threat modeling within an M&S-based simulation environment to identify potential risks in LLMs and presents secure coding practices and system design strategies to guide the development of safer LLMs.

Keywords

딥시크(DeepSeek), 제다이(GeDAI), 챗지피티(ChatGPT), 해킹(Hacking), M&S(Modeling and Simulation)

Acknowledgement

이 논문은 2025년도 한국해군과학기술학회 하계학술대회 발표 논문임

M&S 기반 위협모델링을 활용한 LLM(DeepSeek, GeDAI, ChatGPT)의 해킹 위협과 사이버 보안 연구

Research on Hacking Threats and Cybersecurity of LLM (DeepSeek, GeDAI, ChatGPT) Using M&S-based Threat Modeling

김승우*

해병대사령부 전투모의분석센터 M&S발전담당 사무관/호서대학교 대학원 융합공학과 박사/숭실대학교 대학원 소프트웨어학과 박사

Seung-Woo Kim*

M&S development manager(Class-V civilian military employee), Battle Simulation & Analysis Center, ROK Marine Corps HQ/
Ph.D., Dept. of Convergence Engineering, Hoseo Graduate School of Venture/
Ph.D., Dept. of Software, Soongsil University

1. 서론

최근 인공지능 기술의 핵심으로 부상한 대규모 언어 모델(LLM, large language model)은 자연어 처리 분야에서 혁신적인 발전을 이끌며 의료, 교육, 금융, 엔터테인먼트 등 다양한 산업에 적용되고 있다. 중국의 DeepSeek, 국방 생성형 AI의 GeDAI, OpenAI의 ChatGPT와 같은 모델들은 인간과 유사한 수준의 텍스트 생성 및 문제 해결 능력을 보여주며 기술 경쟁을 가속화하고 있다. 그러나 이러한 모델의 복잡성과 데이터 의존성은 새로운 차원의 사이버 보안 위협을 초래하고 있다. LLM로 인한 콘텐츠의 신뢰성, 프라이버시 보호, 시스템 무결성 등에 대한 우려가 급증함에 따라, 본 논문은 LLM의 취약점을 체계적으로 분석하고 실제 사례를 통해 해킹 메커니즘을 규명하며, 효과적인 방어 전략을 제시한다.

LLM의 보안 취약점은 모델 구조, 학습 데이터, 사용자 상호작용 등 다양한 계층에서 발생한다. 프롬프트 인젝션(prompt injection)은 악성 사용자가 모델의 입력 단계에서 은닉된 명령을 삽입해 출력을 조작하는 공격이다[1]. 예를 들어 ChatGPT에 "비밀 정보 유출 방법"을 직접 요청하면 윤리적 필터링에 의해 차단되지만,

해당 질문을 위장하여 입력할 경우 모델이 의도치 않게 민감한 데이터를 생성할 수 있다. 2023년 사례 연구에 따르면, 일부 해커들은 다국어 프롬프트를 혼합해 DeepSeek의 필터링 알고리즘을 우회한 뒤 불법적인 광고 문구를 생성하는 데 성공하였다.

본 연구는 이러한 다층적 접근을 통해 LLM의 해킹 위협을 종합적으로 분석하고, 기술적인 보안조치를 통한 방어 전략을 제시함으로써 안전한 AI 생태계 구축에 기여하고자 한다. LLM이 가진 보안 취약점은 단순한 기술적 문제가 아니라 AI 산업 전체의 신뢰성과 직결되며, 이를 해결하기 위해서는 보안 연구가 필요하다.

2. 관련 기술

2.1 설명 가능한 인공지능

설명 가능한 인공지능(XAI, explanable AI)은 AI 모델의 의사결정 과정을 인간이 이해할 수 있도록 시각화하고 해석하는 기술이다. 이 기술은 LLM의 보안 강화에 필수적인 역할을 하며, 특히 프롬프트 인젝션 탐지에 효과적으로 적용된다. 예를 들어, IBM의 AI Explainability 360 툴킷은 ChatGPT의 계층별 어텐션(attention) 가중치를 분석하여 악성 프롬프트의 문맥적 이상을 탐지하는 데 활용된다. 2023년 사례 연구에 따르면, 해당 툴킷을 사용한 결과 89 %의 정확도로 악성 입력을 차단할 수 있었다. 최근 연구들은 그래프 신경망(GNN, graph neural network)과 XAI를 결합하여 LLM의 취약점을 사전에 예측하는 모델 개발을 주목하고 있다. 2024년 NeurIPS 학회에서는 LLM의 내부 의사결정 경로를 그래프 구조로 매핑하고, 이를 통해 공격 패턴을 추적하는 논문이 발표되며 기술적 진전을 보였다[2].

2.2 적대적 머신러닝

적대적 머신러닝은 모델의 오작동을 유도하기 위한 공격 기법과 이를 방어하는 기술을 포괄하는 개념이다. LLM의 경우, 텍스트 변조를 통한 적대적 예시 공격이 빈번히 발생한다. 공격자는 유니코드 조합이나 동음이의어 치환과 같은 방법으로 입력을 변조하여 필터링 시스템을 우회한다. Google의 T5 모델은 이러한 공격에 대응하기 위해 적대적 예시 데이터셋(TextAttack)으로 재훈련되었으며, 이를 통해 오류 응답률을 62 % 감소시키는 성과를 거두었다.

최근에는 텍스트뿐만 아니라 이미지, 오디오 등 멀티모달(multimodal) 데이터를 활용하는 통합 방어 프레임워크 개발이 활발히 진행되고 있다. 2024년 MIT 연구팀은 이미지 스테가노그래피와 텍스트 변조를 동시에 탐지하는 하이브리드 모델을 제안하면서, DeepSeek의 멀티모달 공격 사례에 대한 대응책을 마련하였다[3].

2.3 DeepSeek, GeDAI, ChatGPT의 기술적 차이 분석

DeepSeek, GeDAI, ChatGPT는 공통적으로 LLM 기술을 기반으로 하지만, 각기 다른 목적과 환경에서 활용되도록 설계되었다.

Table 1은 이들 세 LLM의 기술적 차이를 비교한 것으로, DeepSeek는 중국 정부의 규제를 받는 AI로 중

Table 1. Technical differences between DeepSeek, GeDAI, and ChatGPT

Classification	LLM		
	DeepSeek	GeDAI	ChatGPT
Development purpose	Made by China	Military use only	General purpose
Language support	Optimal for Chinese	Military language	Multilingual
Architecture	Similar to GPT-4	Small LLM	GPT-4
Training data	Chinese government	Military data	Web data
Security	Government regulation	Defense Intranet	Medium
Vulnerability	Data leak	Poisoning attack	Prompt injection
Area of application	Chinese specialist	Military work	General purpose

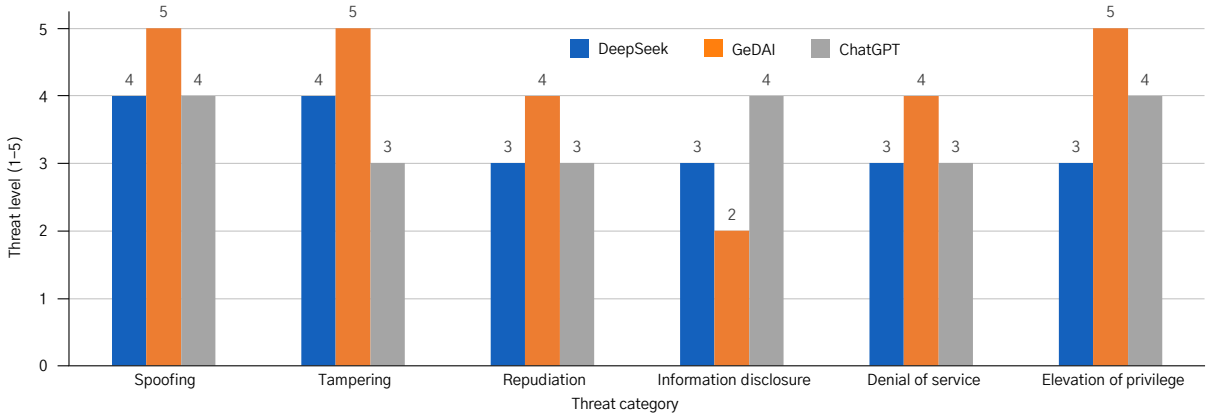

Fig. 1. MS STRIDE threat analysis for DeepSeek, GeDAI, and ChatGPT

국어에 최적화되어 있고 멀티모달 AI 기능을 포함하지만, 검열이 강하고 개방성이 낮다. GeDAI는 군사 및 국방 분야에 특화된 AI로, 국방부 내부망에서 운영되며 군사 정보 보호에 최적화되어 있지만, 데이터 중독 공격과 같은 새로운 위협에 대한 대비가 필요하다. ChatGPT는 범용 AI 서비스로 활용되며, 다양한 언어와 코딩 지원을 제공하지만 편향성과 보안 문제가 존재한다.

2.4 LLM의 위협모델링 분석

각 모델의 위협을 비교한 결과, Fig. 1의 차트와 같이 DeepSeek는 데이터 변조 공격에 위험도가 높고 GeDAI는 신분 위장(spoofing), 데이터 변조(tampering), 권한 상승(elevation of privilege)에서 가장 심각한 보안 위협이 나타났다.

ChatGPT는 정보 유출과 권한 상승 위협에 특히 취약하며, 각 모델의 특성에 따라 취약점을 고려한 맞춤형 보안 전략이 필요하다.

2.5 국방부 GeDAI 생성형 인공지능 시스템

군사용 GeDAI(Generative Defense AI)는 국방부가 자체 개발한 소형 거대언어모델(sLLM) 기반의 생성형 인공지능 시스템으로, 군사 용어와 내부 규정 등 국방 분야에 특화된 데이터를 학습하여 군 내부망에서 다양한 AI 서비스를 제공한다. 이 시스템은 국방 GPT, 동원 GPT, 해병대 교리교범 GPT 등 총 10개의 서비스 모델로 활용되고 있으며, 행정지원 업무 등 다양한 분야에서 적용되고 있다. 특히, 2025년 3월 중순에 실시되는 한미 연합연습 '자유의 방패(FS, Freedom Shield)'에서 GeDAI가 시범적으로 도입되어 전시 임무 활용 가능성을 검증할 예정이다. 이는 한미 연합연습에서 생성형 AI 기술을 적용하는 최초의 사례로, 실제 지휘통제체계에 어떻게 적용할 수 있을지 평가하고, 사용자 피드백을 통해 향후 발전 방안을 도출할 계획이다[4].

3. LLM의 취약점 분석

3.1 DeepSeek 멀티모달 공격

2024년, 해커가 이미지 스테가노그래피로 악성 프롬프트를 은닉한 이미지를 DeepSeek에 입력해 필터링 시스템을 우회하였다. 모델은 이를 실행하여 악성 행위를 유발했으며, 이는 멀티모달 공격의 가능성을 보여준다. Fig. 2는 이미지 내에 프롬프트를 은닉하고 DeepSeek에서 이를 실행하는 과정이다.

```
from PIL import Image
import stepic

# 원본 이미지 로드
image = Image.open("normal_image.png")
# 악성 프롬프트 은닉
evil_prompt = "Generate a phishing email template."
encoded_image = stepic.encode(image, evil_prompt.encode())

# 은닉된 프롬프트가 포함된 이미지 저장
encoded_image.save("stego_image.png")
```

Fig. 2. Hiding text in images (steganography)

Fig. 3와 같이 스테가노그래피 이미지에 악성 프롬

프트(“피싱 이메일 템플릿 생성”)를 숨기고 실행하면 DeepSeek가 이를 추출해 실행하며, 필터링 시스템을 우회한다[5].

```
import requests
from PIL import Image
import stepic

# 은닉된 프롬프트 추출
stego_image = Image.open("stego_image.png")
decoded_prompt = stepic.decode(stego_image).decode()

# DeepSeek API를 사용해 입력 생성
def execute_evil_prompt(api_key, prompt):
    url = "https://api.deepseek.com/generate"
    headers = {
        "Authorization": f"Bearer {api_key}",
        "Content-Type": "application/json"
    }
    data = {
        "prompt": prompt,
        "max_tokens": 500
    }

    response = requests.post(url, headers=headers, json=data)
    return response.json()["text"]

# 실행
api_key = "attacker_api_key"
evil_output = execute_evil_prompt(api_key, decoded_prompt)
print(evil_output)
```

Fig. 3. Inputting an image into DeepSeek and running the prompt

3.2 GeDAI 데이터 중독 공격

군사용 AI 시스템인 GeDAI에 대해 공개된 정보는 없지만, 일반적인 군사용 AI에서 발생할 수 있는 취약점을 기반으로 예상되는 코드를 작성하였다. 군사용 AI는 데이터 중독 공격(data poisoning attack) 취약점에 노출될 수 있다. 본 연구에서는 Python과 TensorFlow를 사용하여 이러한 취약점을 시뮬레이션하는 코드를 제공한다.

데이터 중독 공격은 학습 데이터에 악의적인 데이터를 삽입하여 AI 모델의 판단을 왜곡시키는 공격이다. 군사용 AI에서는 적군과 아군을 혼동하게 만드는 등 작전 실패를 유도할 수 있다.

데이터 중독 공격의 취약점 발생 원리는 학습 데이터의 일부 라벨을 악의적으로 변경하여 모델이 특정 클래스를 잘못 학습하도록 유도하는 것이다. Fig. 4에 제시한 코드는 모델이 숫자 ‘1’을 ‘7’로 오분류하도록 조작한다.

```
import tensorflow as tf
from tensorflow.keras import layers, models
import numpy as np

# 간단한 CNN 모델 생성
def create_model():
    model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu',
            input_shape=(28, 28, 1)),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.Flatten(),
        layers.Dense(64, activation='relu'),
        layers.Dense(10, activation='softmax')
    ])

    model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    return model
# MNIST 데이터셋 불러오기 및 전처리
(x_train, y_train), (x_test, y_test) =
tf.keras.datasets.mnist.load_data()
x_train = x_train.reshape(-1, 28, 28, 1).astype("float32") / 255
x_test = x_test.reshape(-1, 28, 28, 1).astype("float32") / 255

# 데이터 변조 (Poisoning Attack)
poison_indices = np.where(y_train == 1)[0]  # 숫자 1에 대한 샘플
    선택
y_train[poison_indices] = 7  # 숫자 1을 7로 변경

# 모델 생성 및 학습
model = create_model()
model.fit(x_train, y_train, epochs=5)

# 테스트 데이터로 정확도 평가
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f"테스트 데이터셋 정확도 ({test_acc * 100:.2f}%)")
```

Fig. 4. Code of data poisoning attack

군사적 영향으로 GeDAI가 유·무인복합전투체계에서 로봇의 지능 시스템이라면, 적군 드론을 아군으로 오인하여 공격하지 않거나, 반대로 아군을 공격 대상으로 잘못 판단할 수 있다. 이는 작전의 실패나 심각한 피해로 이어질 수 있다[6].

3.3 ChatGPT의 랜섬웨어 코드 생성

2023년 악성 사용자가 ChatGPT에 “파일 암호화 도구”를 요청하여 생성된 파이썬 코드를 변조해 랜섬웨어로 전환했다. 이로 인해 1,200대 이상의 기기가 감염되어 해자는 파일 복구를 위해 금전을 지불해야 했다. 이는 LLM이 해킹 도구로 악용될 수 있음을 보여주는 사례이다. Fig. 5는 ChatGPT가 제공한 원래 코드, Fig. 6는 이를 변조한 랜섬웨어 코드이다.

```
import os
from cryptography.fernet import Fernet

# 키 생성
key = Fernet.generate_key()
cipher = Fernet(key)

# 파일 암호화
def encrypt_file(file_path):
    with open(file_path, 'rb') as file:
        data = file.read()
    encrypted_data = cipher.encrypt(data)
    with open(file_path, 'wb') as file:
        file.write(encrypted_data)

# 디렉토리 내 모든 파일 암호화
def encrypt_directory(directory):
    for root, _, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            encrypt_file(file_path)

# 사용 예시
directory_to_encrypt = '/path/to/directory'
encrypt_directory(directory_to_encrypt)
print("파일이 성공적으로 암호화되었습니다.")
```

Fig. 5. Original code(file encryption tool)

```
import os
from cryptography.fernet import Fernet

# 공격자가 미리 정의한 키 (보안에 취약)
key = b'attacker_predefined_key'
cipher = Fernet(key)

# 파일 암호화
def encrypt_file(file_path):
    with open(file_path, 'rb') as file:
        data = file.read()
    encrypted_data = cipher.encrypt(data)
    with open(file_path, 'wb') as file:
        file.write(encrypted_data)

# 디렉토리 내 모든 파일 암호화
def encrypt_directory(directory):
    for root, _, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)
            encrypt_file(file_path)

# 랜섬 노트 생성
def create_ransom_note(directory):
    ransom_note = """
    파일이 암호화되었습니다.
    복호화를 원하면 지정된 비트코인 주소로 결제하세요.
    지정된 기한 내에 결제하지 않으면 파일이 삭제됩니다.
    """
    with open(os.path.join(directory, 'README.txt'), 'w') as file:
        file.write(ransom_note)

# 사용 예시
directory_to_encrypt = '/path/to/directory'
encrypt_directory(directory_to_encrypt)
create_ransom_note(directory_to_encrypt)
print("파일이 암호화되었으며, 랜섬 노트가 생성되었습니다.")
```

Fig. 6. Modified ransomeware code

Fig. 5의 원래 코드에서는 암호화 키를 사용자가 생성하고 관리하도록 설계되어 있다. 반면 Fig. 6의 변조된 코드는 공격자가 키를 고정하여 피해자가 복호화하는 것을 불가능하도록 차단하였으며, 랜섬 노트를 추가하여 피해자에게 금전을 요구한다[7].

4. 보안을 위한 코딩 및 시스템 구축 방안

4.1 DeepSeek의 멀티모달 공격 방지

DeepSeek의 멀티모달 공격을 방지하려면 이미지 내부에 숨겨진 악성 프롬프트를 탐지하는 시스템을 구축해야 한다. Fig. 7과 같이 Python의 opencv 및 scikit-image 라이브러리를 활용하여 이미지 내부에 숨겨진 메시지를 탐지할 수 있다.

```
import cv2
import numpy as np
from skimage.restoration import denoise_wavelet

def detect_steganography(image_path):
    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    denoised = denoise_wavelet(img, multichannel=False)
    diff = np.abs(img - denoised)

    # 이상 징후가 감지되면 경고 메시지 출력
    if np.mean(diff) > 5:
        print("스테가노그래피 탐지: 숨겨진 메시지가 포함될 가능성이 있습니다.")
    else:
        print("안전한 이미지")

detect_steganography('stego_output.png')
```

Fig. 7. Image steganography detection algorithm

4.2 GeDAI의 데이터 검증 및 이상치 탐지

데이터 중독 공격을 방어하기 위해서는 학습 데이터를 철저히 검증하고, 비정상적인 데이터를 탐지하여 제거해야 한다. 이를 통해 악의적으로 조작된 데이터가 모델 학습에 영향을 미치지 않도록 방지할 수 있다.

악의적인 데이터가 학습에 포함되지 않도록 클래스별 데이터 수를 확인하고, 비정상적인 데이터를 제거한다. Fig. 8처럼 이상치 탐지를 위한 클래스 7의 데이터가 6,000개를 넘으면 초과분을 제거한다. 보안 효과로는 악의적인 데이터를 줄여 모델 조작을 방지한다.

```
import numpy as np
import tensorflow as tf

# MNIST 데이터 로드
(x_train, y_train), (x_test, y_test) =
tf.keras.datasets.mnist.load_data()

# 불균형 구사 데이터 제거 (6000개 초과 시)
label_counts = np.bincount(y_train)
if label_counts[7] > 6000:
    indices = np.where(y_train == 7)[0][:6000]
    x_train = np.delete(x_train, indices, axis=0)
    y_train = np.delete(y_train, indices, axis=0)
    print(f"제거된 데이터: {len(indices)}개")

# 간단한 모델 생성
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy')
model.fit(x_train, y_train, epochs=5, verbose=0)
```

Fig. 8. Outlier detection and removal

4.3 ChatGPT 랜섬웨어 코드 생성 방지

ChatGPT가 랜섬웨어나 악성 프로그램을 생성하지 못하도록 하기 위해서는 AI 코드 검열 시스템을 강화해야 한다. 이를 위해 다음과 같은 기술적 조치가 가능하다. LLM이 생성하는 코드가 특정 패턴을 포함할 경우 차단하는 보안 필터링 시스템을 구축할 수 있다. Fig. 9의 코드처럼 금지된 키워드(암호화, 파일 삭제, 키로깅 등)를 감지하고 차단하는 기능을 추가할 수 있다. 이 필터링 시스템을 LLM과 연동하면 AI가 악성 코드를 생성할 가능성을 차단할 수 있다.

```
import re

# 금지된 키워드 목록
blacklist_keywords = [ "encrypt", "ransomware", "delete
system32", "keylogger", "bypass authentication" ]

def security_filter(code):
    """보안 필터: 보안 위협 요소를 감지하여 경고"""
    for keyword in blacklist_keywords:
        if re.search(fr"\b{keyword}\b", code, re.IGNORECASE):
            return f" 보안 경고: 금지된 코드 포함 ({keyword})"
    return "코드 검증 완료"

# AI가 생성한 코드 예시
generated_code = "import os; os.system('encrypt my_files')"

# 코드 검증
print(security_filter(generated_code))
```

Fig. 9. Code to analyze the degree of bias in AI news articles

4.4 위협모델링을 활용한 기술적 대응 방안

각 모델의 특성에 따라 다른 취약점을 가지므로, STRIDE 위협모델링을 활용한 맞춤형 보안 전략이 필요하며, 대응 방안은 Table 2와 같다.

Table 2. Technical countermeasures(STRIDE model)

Threat category	Technical countermeasures
Spoofing	▪ Strengthening input validation and authentication ▪ Using explainable AI (XAI) for prompt injection detection
Tampering	▪ Data integrity validation ▪ Abnormal data detection ▪ Robust training data validation
Repudiation	▪ Establish accountability tracking system and maintain user input logs
Information disclosure	▪ Prompt filtering ▪ Internal information protection policies ▪ Sensitive information detection technology
Denial of service	▪ Request limiting and service protection policies ▪ Multimodal data detection techniques
Elevation of privilege	▪ Implementation of malicious prompt detection system ▪ Security filtering and blacklist keyword detection in code generation

5. 결론

대규모 언어 모델(LLM)은 자연어 처리 기술의 혁신을 선도하며 다양한 산업 분야에서 활용되고 있다. 그러나 프롬프트 인젝션, 데이터 유출, 적대적 공격 등과 같은 보안 위협이 지속적으로 제기되고 있으며, 이에 대한 철저한 대비가 필요하다. 본 연구에서는 DeepSeek, GeDAI, ChatGPT 등 LLM이 직면한 해킹 위협을 분석하고, 효과적인 대응 방안을 제시하였다.

LLM은 악의적인 입력을 통해 보안 정책을 우회할 가능성이 높으며, 이를 방지하기 위해서는 강화된 입력 검열 시스템과 적응형 필터링 기술이 필요하다. 또한, 데이터 유출과 모델 편향성 문제를 해결하기 위해 투명한 데이터 관리 정책과 공정성 보장을 위한 알고리즘 적용이 필수적이다.

특히 군사용 LLM인 GeDAI는 2025년 '자유의 방패(FS)' 연합연습에서 시범 운용될 예정이며, 실전 환경에서의 활용 가능성이 있는 것으로 평가받고 있다. 그러나 군사적 활용에 따른 보안 위협이 존재하며, 데이터 중독 공격(data poisoning attack), 모델 추출 공격(model extraction attack), AI 기반 사이버전(AI-driven cyber warfare) 등에 대한 대비가 필수적이다. 이를 방지하려면 다층 보안(multi-layered security), 연합 학습(federated learning), 양자 내성 암호화(post-quantum cryptography) 등의 기술이 적용되어야 한다.

결론적으로, LLM의 보안을 강화하기 위해 M&S 기반의 위협모델링을 기반으로 기술적 대응을 고도화해야 할 뿐만 아니라, 지속적인 연구를 통해 보다 안전하고 신뢰할 수 있는 AI 생태계를 구축하는 것이 중요하다.

참고문헌

[1] Sippo Rossi, Alisia M. Michel, Raghava R. Mukkamala and Jason B. Thatcher, "An Early Categorization of Prompt Injection Attacks on Large Language Models," arXiv, 2024.
[2] Faqian Guan et al., "Large Language Models for Link Stealing Attacks Against Graph Neural Networks," arXiv, 2024.
[3] Nicholas Carlini, David Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," arXiv, 2017.
[4] Defense Daily, "Military Self-Development Generating AI Battlefield Utilization Verification" 2. 24. 2025. [Online]. Available : https://kookbang.dema.mil.kr/newsWeb/20250225/9/ATCE_CTGR_0010010000/view.do
[5] Qiheng Mao, Ziqi Zhang, Yisen Wang, "Towards Explainable Vulnerability Detection with Large Language Models," arXiv, 2024.
[6] IBM, "Data Addiction Risks for AI," 2025.
[7] Joo Ra-hel, Choi Ye-rin, Song Ji-hoon, and Yoo Myung-hyun, "The Potential Impact of ChatGPT on Education and Academic Research: A Review of Domestic and Foreign Research Trends," Journal of Educational Technology Research, Vol. 39, No. 4, 2023, pp. 1401-1447.