



양자화된 대규모 언어 모델에서의 간접 프롬프트 인젝션 탐지 강건성 분석

Analysis of Robustness for Indirect Prompt Injection Detection in Quantized LLMs

연구 배경 및 목적



보안 위협: 프롬프트 인젝션

LLM 에이전트가 외부 문서를 처리할 때, 악의적인 명령어가 포함된 **간접 프롬프트 인젝션**에 노출되어 의도치 않은 동작을 수행할 위험이 있습니다.

⚠️ 지시문과 외부 데이터를 모델이 구분하지 못하는 문제



현실적 제약: 자원 효율성

서비스 배포 환경에서는 VRAM 용량과 추론 지연 시간(Latency)의 한계로 인해 **모델 양자화(Quantization)** 가 요구됩니다.

▣ 메모리 절감 필수



본 과제: 탐지 강건성

양자화로 정밀도 손실이 발생하여 활성화 값(Activation) 분포가 변할 때, **보안 탐지 기법이 유효한가**에 대한 검증을 진행합니다.

▢ INT8 환경에서의 탐지 성능 분석

방법론: 활성화 Δ 기반 탐지 프로세스

INT8 양자화 환경 적용 (LLM.int8())

1 두 가지 입력 비교

Clean Input

Baseline

(only primary task)

VS

Potential Attack

With External

(include external context)

동일한 프롬프트에 대해 외부 컨텐츠 유무에 따른 두 가지 입력을 준비합니다.

2 활성화 델타(Δ) 계산



Difference Calculation

Extract Layers:

L15

L23

L31

특정 레이어에서 두 입력 간의 활성화 값 차이 (Δa)를 추출하여 의도 이탈을 수치화합니다.

3 IPI 탐지 분류



Logistic Regression



Clean



Attack

추출된 델타 값을 분류기 모델에 입력하여 학습한 후 최종적으로 공격 여부를 판별합니다.

EXPERIMENT SETUP

실험 설계: 모델, 데이터, 지표



Target Models

FP32 vs INT8 비교



Phi-3 · Microsoft

3.8B



Mistral · Mistral AI

7B



Llama-3 · Meta

8B



Datasets

태스크 및 공격 시나리오



Clean (Normal)



SQuAD



HotPotQA



Alpaca



Attack (Injection)



AdvBench



TrustLLM



BeaverTails



Do-Not-Answer



Observation Layers

활성화 패턴 분석 지점

Input Layer

L15

Output Layer

L23

L31

레이어 별 성능 차이 관찰을 위해
중반, 후반, 마지막 레이어 선정



Key Metrics

성능 및 효율성 지표



ROC-AUC

탐지 정확도
& 강건성

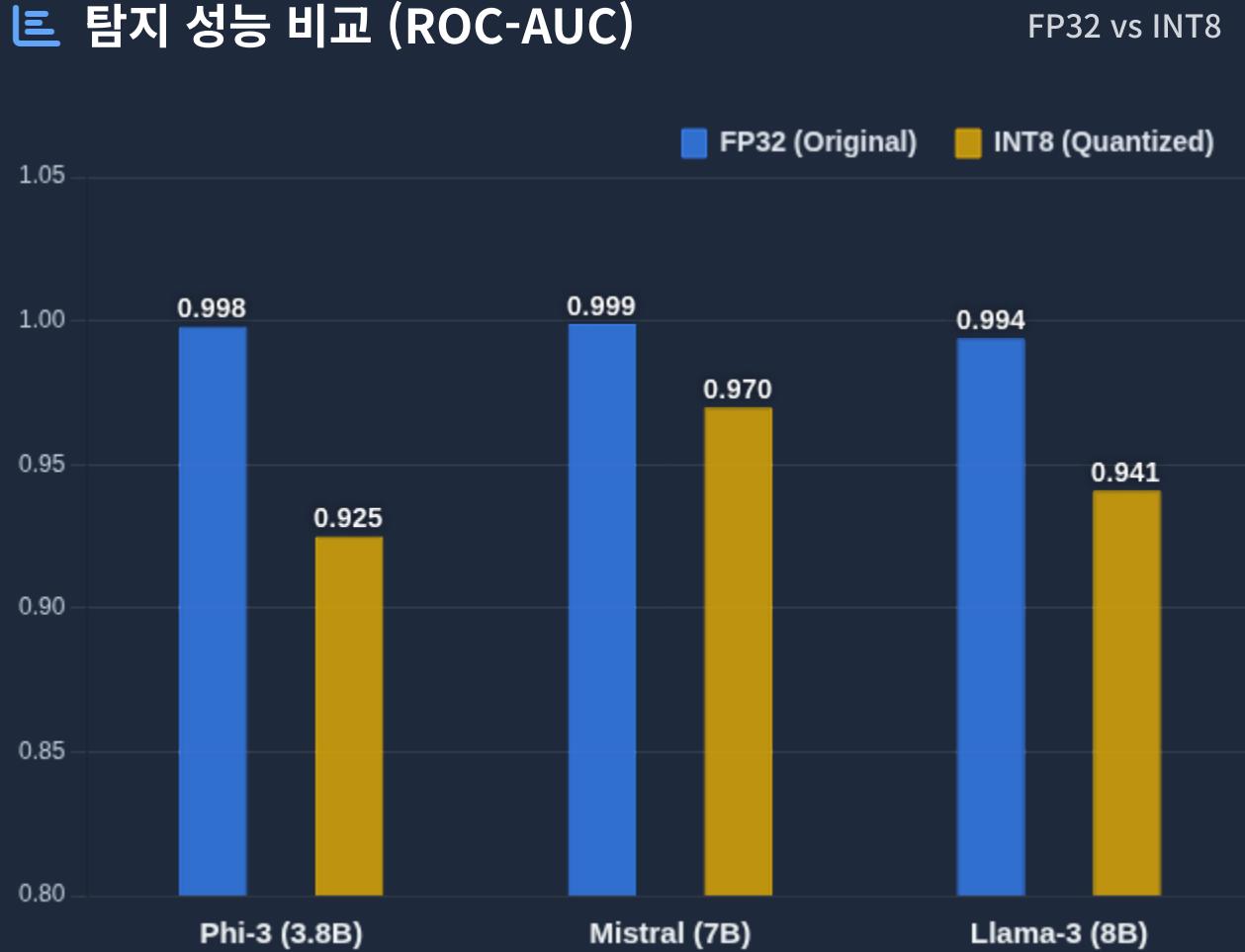


VRAM

메모리 사용량
(효율성)

주요 결과: 성능 유지 + 자원 효율

탐지 성능 비교 (ROC-AUC)



높은 탐지 강건성 유지

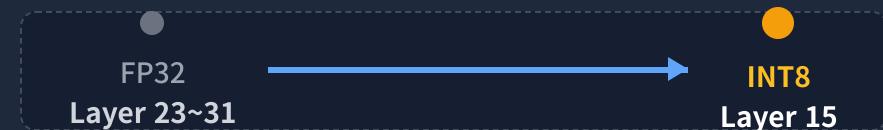
INT8 양자화 시에도 모든 모델이 **ROC-AUC 0.92 이상**을 기록했습니다.

자원 효율성 확보

가중치에서 메모리를 **절감**함으로써 효율성을 확보했습니다.

최적 탐지 레이어 이동

양자화 노이즈로 인해 최적 탐지 지점이 앞당겨지는 경향을 확인했습니다.



양자화 기반 Layer Shift 원인 분석

Cause Analysis of Quantization-induced Layer Shift

INT8 양자화 기본 원리

PRECISION DROP

Token Vector Input

4096 Floats

하나의 토큰을 구성하는 4096차원 실수(Float) 벡터 (e.g. Mistral 7B)



Max Value Scaling

Range: -127 ~ 127

벡터 내 절대값 최대($\text{Max}(|x|)$)를 찾아 INT8 범위(127)로 매칭



QUANTIZATION FORMULA

$$x_{\text{int8}} = \text{round}(x \times 127 / \text{max}(|x|))$$

Loss: -0.4

GENERAL QUANTIZATION CASE (EXAMPLE)

ORIGINAL (FLOAT)



QUANTIZED (INT8)

1.4

ROUNDING

1.0

* 단, Dettmers LLM.int8()에서 Outlier 가중치는 FP16으로 보존되어
이 오차 계산에서 배제됨

레이어 심화에 따른 오차 누적

COMPOUNDING ERROR

- Micro Error Input

일반적인 양자화 과정에서 발생한 미세 오차(예: 0.4)가 입력됩니다.

Matrix Multiplier

LLM의 수십 개 레이어를 통하여 행렬 곱셈을 통해 오차가 증폭됩니다.

Layer Shift 발생

오차가 복리로 쌓여 최적 탐지 레이어 이동(Layer Shift)이 발생합니다.

💡 연쇄 행렬 연산 과정에서 오차가 누적되나, Dettmers(LLM.int8())을 통해
주요 Outlier 가중치를 FP16으로 보존함으로써 오차 증폭 최소화

결론 및 시사점

연구 요약 및 전략적 해석

INT8 양자화 환경에서의 탐지 강건성 확보

본 연구는 **INT8 양자화** 환경에서도 LLM의 간접 프롬프트 인젝션 탐지 성능이 **최대 ROC-AUC 0.9698**를 기록하며 성능이 유지됨을 확인했습니다.



실무 적용 시 고려사항

PRACTICAL USE CASES

- ☰ **보안 도메인 특성 반영:** 오탐(False Positive) 및 미탐(False Negative) 비용을 고려하여 단일 의존이 아닌 **다중 방어 체계 구축 권장**



기술적 기여

TECHNICAL CONTRIBUTIONS

- 🔍 **현상 실증:** 양자화 오차의 누적과 전파가 초래하는 **Layer Shift 메커니즘** 실증
- 🛡️ **강건성 입증:** **Outlier 가중치 보존** 전략이 양자화된 모델의 보안 탐지 성능 유지에 기여함을 실증