CICI: UCSS: Secure Machine Learning Inference in IoT-driven Analytical Scientific Infrastructure

PI: Ruimin Sun (Florida International University) Co-PI: Jason Liu (Florida International University), Yuede Ji (University of Texas Arlington)

Problem:

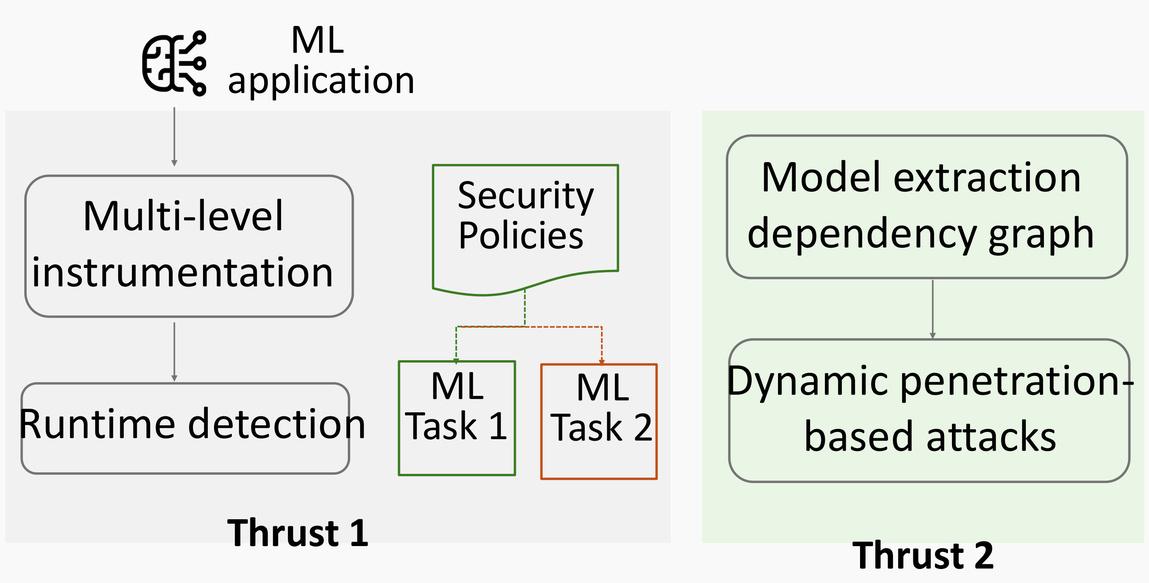
Goal: Designing secured on-device Machine Learning (ML) in scientific Cyberinfrastructure (CI).

Why On-Device LLMs? Modern edge devices can run small-to-medium LLMs locally using AI

Existing solutions are insecure:

- On-Device ML is insecure. E.g., a recent study on 1,468 ML applications shows that 41% of them do not protect the ML models, and 37% of the unprotected models are security-critical.
- A lack of security assessment. Existing studies fail to capture ML-related control flow.

Project Overview:



Research Thrusts:

- *Thrust 1: Detecting and preventing model extraction attacks.* It contains a multi-level instrumentation method, an efficient runtime detection framework, and re-defined memory regions and security policies for ML tasks.
- *Thrust 2: Accurately assessing the security of on-device models.* It contains a newly designed model extraction dependency graph (MDG), and a penetration-based technique using the MDG results and confirm the existence of model extraction attacks.

Intellectual Merit:

- This project raises awareness of ML model extraction attacks in scientific CIs, and significantly reduce the attack surfaces of ML models in CIs.
- This project results in a runtime detection and prevention mechanism for model extraction attacks, and a comprehensive assessment to improve ML model security.
- This project enables new functionalities in CIs and allows more mission-critical ML models safely and securely deployed.

Preliminary Results: SecureInfer: Heterogenous TEE-GPU Architecture for Privacy-Critical Tensors in LLM Deployment

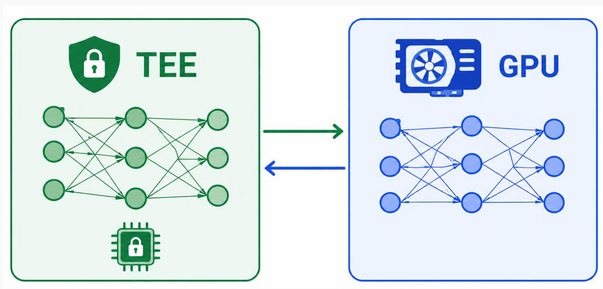
Goal Defend - model parameters, weights or functionality by these attacks

Existing Defenses: Homomorphic Encryption, or Differential Privacy — +

Employing: Trusted Execution Environments (TEEs)

Challenge: Putting the whole LLM in the TEE introduces large overhead (x50+)

Solution: Model Partitioning –



Results: Latency & Throughput

- Is **4.7× faster than TEE-only**
- **2.06x overhead vs GPU-only baseline**
- **Layer 0 dominates latency** (emd + KV-init = ~55% time)

Defense Performance on LLama2

- **TEE-Based Partitioning + Logits Perturbation**

Attack Simulated: Model-Based Attack

- Black-box model extraction (KnockoffNets-style).

Without Defense: BLEU ≈ 0.45–0.62, Token Match ≈ 94%.

SecureInfer: BLEU ↓ to 0.12, Token Match ↓ to 56%.

Contributions

- Present Hybrid TEE–GPU architecture,
- Protects privacy-critical transformer only
- Preserves GPU-accelerated performance
- Strongly mitigates MEA's

Paper URL