Final Project in NT522 Course – Machine Learning in Cybersecurity

SCALM: Detecting Bad Practices in Smart Contracts Through LLMs

Members: Nguyễn Đặng Quỳnh Như (22521050), Trần Văn Chiến (22520561), Lê Minh Quân (22521181), Đặng Đức Tài (22521270)

Group: G08 ***Project ID: CK14***

I. Introduction:

The rapid development in the use of economically incentive smart contracts is hampered by persistent security issues, which frequently result in significant money losses due to the contracts' immutable nature after deployment. This study describes a novel architecture that combines fine-tuned large language models (LLMs) with Retrieval-Augmented Generation (RAG) to increase the accuracy and interpretability of vulnerability detection in smart contracts. Tested on two public benchmarks—SolidiFI-Benchmark and SmartBugs Curated—our approach outperforms zero-shot LLMs by up to 34% in accuracy. This result shows the strength of combining RAG with fine-tuning to deliver more precise and understandable smart contract security assessments.

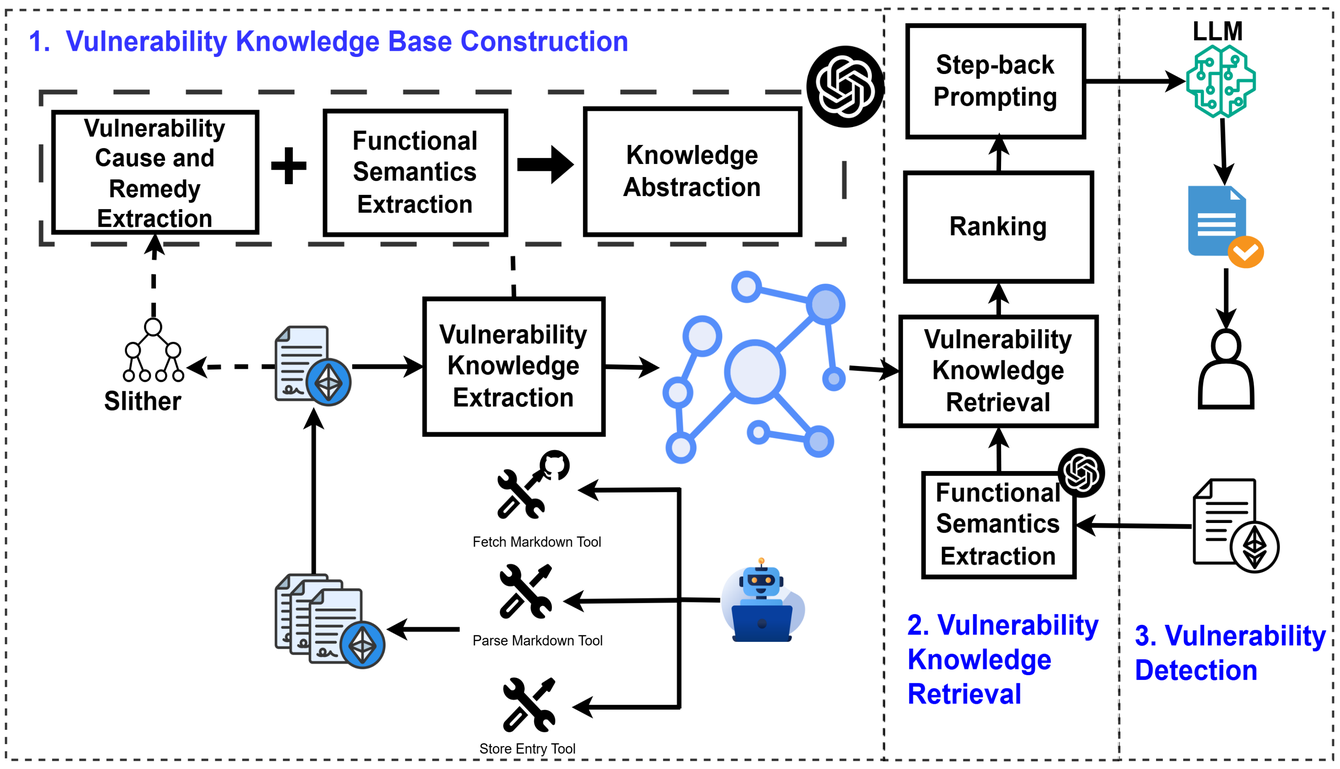

Fig.1 *Overall Architecture of SCALM*

II. Methodology:

Stage 1: Our approach begins by constructing a vulnerability knowledge graph based on the SWC Registry. An autonomous agent is used to collect smart contracts from GitHub repositories, which are then used to build the graph for RAG

Stage 2: Given a code snippet, we extract its functional semantics and trace related vulnerability information within the knowledge graph, enriching the context for LLM inference.

Stage 3: The system retrieves contextually relevant documents from the graph and employs fine-tuned open-source LLMs—Qwen-Coder-2.5-14B and DeepSeek-R1-Distill-Llama-8B—to analyze the smart contract. This can reduce the likelihood of hallucinations by the LLMs.

III.Experiments and Result:

On SmartBugs Curated:

On the SmartBug task, Fine-tuned Qwen-coder-2.5 with RAG achieves the best performance with 0.865 accuracy and a strong F1-score of 0.699. It also has the highest recall at 0.685. Fine-tuned Deepseek with RAG follows, with 0.731 accuracy and an F1-score of 0.651. Both fine-tuned models outperform GPT-3.5 variants. Qwen-coder-2.5 stands out as the most effective model overall.

On SolidiFI:

Figure 3 compares the performance of various models on accuracy, precision, recall, and F1-score. Among them, the Fine-tuned Qwen-coder-2.5 with RAG achieves the highest accuracy of 0.832, significantly outperforming all other models. It also leads in precision at 0.713, though its recall remains moderate at 0.523. Fine-tuned Deepseek with RAG follows, showing strong precision at 0.667 but lower recall at 0.481, leading to a slightly higher F1-score than GPT-3.5 with RAG. Overall, both fine-tuned models with RAG show improved performance over GPT-3.5 variants.

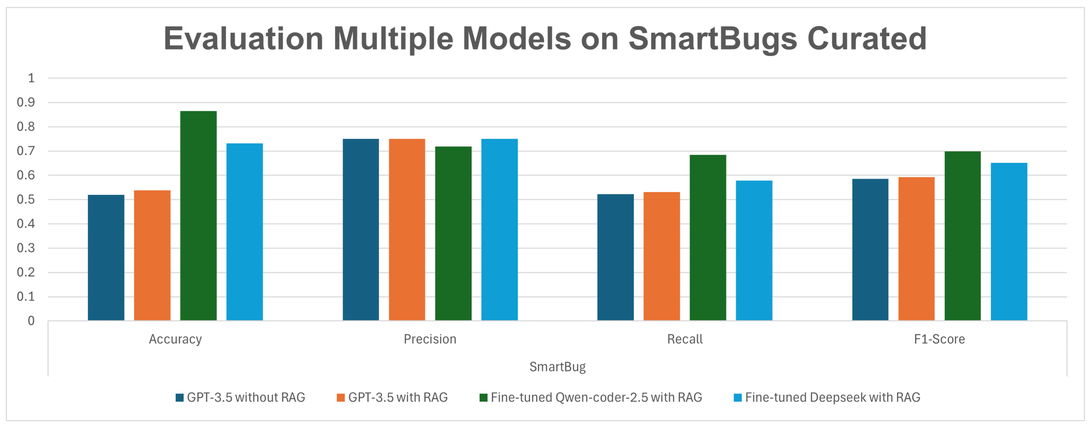

Fig.2 *Evaluation Multiple Models on SmartBug*

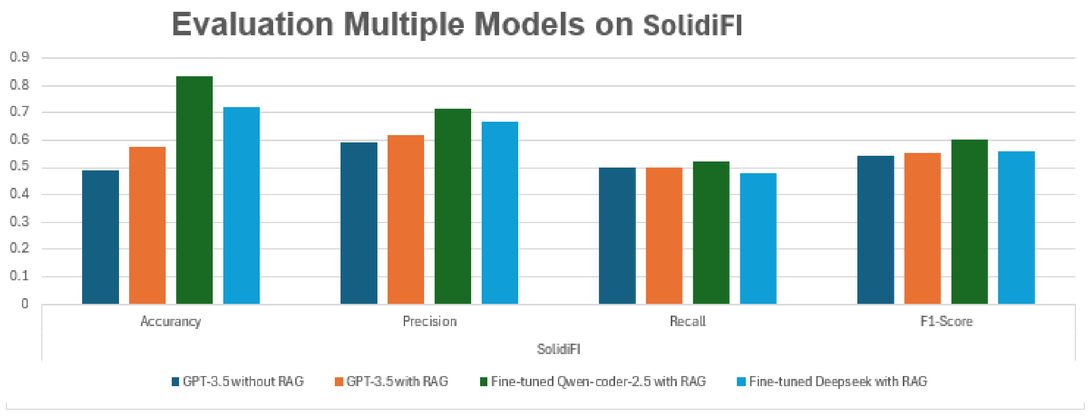

Fig.3 *Evaluation Multiple Models on SolidiFI*