A Study of Synthetic-Data-Enhanced Analysis for Smart Contracts: Function-Level Detection and Explanation

2310409 NGUYEN, Minh Ngoc

Smart contracts are self-executing programs that run on blockchain platforms, most notably Ethereum. They automate transactions and enforce agreements without intermediaries, forming the foundation of decentralized finance (DeFi), non-fungible tokens (NFTs), and decentralized applications (dApps). Despite their growing importance, smart contracts remain prone to security vulnerabilities. Exploited bugs can lead to irreversible financial losses, service disruptions, and systemic failures. Although machine learning-based tools have emerged to aid vulnerability detection, two critical challenges remain: (1) limited fault localization at the function level, and (2) a lack of interpretable, human-readable explanations that enable developers to understand and fix issues effectively.

This thesis addresses both challenges by proposing a unified framework that combines graph-based neural network modeling with explainable language model techniques. Specifically, the contributions consist of: (1) a function-level vulnerability detection system using Sub-Graph Neural Networks (Sub-GNNs), and (2) an explanation generation mechanism based on synthetic data and Chain-of-Thought (CoT) prompting using large language models (LLMs). These two components aim to improve both the technical granularity and practical usability of smart contract security analysis.

The first part of the thesis introduces a novel function-level detection method that decomposes smart contracts into subgraphs centered around individual functions. While prior approaches using Graph Neural Networks (GNNs) operate at the contract level, they fail to pinpoint specific sources of vulnerabilities, limiting their value for debugging and remediation. To overcome this, we construct function-level subgraphs that incorporate control-flow and data-flow dependencies, preserving the semantic and structural context of each function. We then apply a Sub-GNN model to perform vulnerability classification at this finer granularity. Empirical evaluation on a curated synthetic dataset demonstrates that the proposed method achieves high precision in localizing faulty functions. Although it trades off a small margin of global classification accuracy compared to full-graph models, the localized predictions are significantly more actionable for developers. A benchmark comparison quantifies this trade-off and validates the effectiveness of subgraph-based analysis in practical settings.

To facilitate this line of work, we develop a synthetic dataset of smart contracts with function-level vulnerability labels. The dataset includes diverse

vulnerability types such as reentrancy, integer overflows, access control flaws, and unhandled exceptions. Each function is annotated with corresponding vulnerability types and contains metadata for constructing control and data flow graphs. This dataset fills a gap in the current landscape, which largely lacks fine-grained, labeled corpora for training and evaluating function-level detectors.

The second component of the thesis tackles the issue of explanation. While detecting a vulnerability is important, understanding why it occurs and how to resolve it is crucial for real-world usability. Most existing detection tools output low-level indicators such as line numbers or vulnerability labels without offering semantic explanations. To address this gap, we propose an explanation generation system that produces structured, human-readable justifications for detected vulnerabilities. We construct another synthetic dataset where each entry consists of a vulnerable function, its formal label, and a professionally formatted explanation describing the issue, its cause, and suggested remediation steps. These explanations are derived from real-world audit patterns and follow a consistent template.

Together, these two components form a comprehensive framework for smart contract vulnerability analysis. The Sub-GNN-based detector provides precise localization of faulty functions, while the CoT-guided explanation generator delivers semantic insight into the causes and consequences of the vulnerabilities. This dual capability bridges the gap between vulnerability detection and developer comprehension.

The thesis concludes with a discussion of future directions. On the detection side, extending the Sub-GNN architecture to support inter-function and inter-contract reasoning could enable the modeling of call chains and complex compositional vulnerabilities. On the explanation side, integrating user feedback to iteratively refine generated explanations could support interactive auditing tools. Furthermore, we propose exploring multimodal models that combine graph-based embeddings with textual features to enhance both detection and explanation tasks.

Keywords: Smart Contracts, Vulnerability Detection, Graph Neural Networks, Subgraph Analysis, Function-Level Classification, Chain-of-Thought Prompting, Large Language Models, Security Analysis, Solidity, Synthetic Dataset, Blockchain Security.