Associate Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

# Evaluating Smart Contract Bytecode Similarity

## Master Thesis

Jens Ernstberger, November 15, 2022

Title: "Evaluating Smart Contract Bytecode Similarity"
Supervisor: Jens Ernstberger, Liyi Zhou
Period: 24 weeks

### Context

Blockchain-based Decentralized Finance (DeFi) ecosystem has attracted a surge in popularity since the beginning of 2020. The peak total value locked (TVL) for DeFi surpassed 253 billion USD on Dec 2, 2021, with Ethereum (145 billion, 57% TVL) and BNB Smart Chain (19.8 billion, 8% TVL) sharing the majority of DeFi's activity. While DeFi certainly provides many protocols inspired by traditional finance such as cryptocurrency exchanges, lending platforms, and derivatives, novel constructs known as flash loans and atomic composable DeFi trading emerged. Unfortunately, these very intertwined DeFi systems, coupled with the already well-studied vulnerability-prone smart contracts, broadened the threat surface of DeFi protocols.
In this thesis, it is your task to detect adversarial and vulnerable contracts through a novel methodology that analyzes the Bytecode of an already deployed contract. The reasoning is as follows — the Bytecode of a smart contract is publicly available, where as the actual human-readable Solidity source code can only be obtained through a Smart Contract decompiler, or open-source version control tools such as Github. By automating attack detection, without demanding for source code files to apply static analysis or manual security audits, one may detect attacks & vulnerabilities in real-time, even before the smart contract is deployed. In a first step, you should develop and automate a pipeline to crawl DeFi incidents/attacks to obtain a ground truth dataset. In a second step, you need to develop a methodology that applies advanced machine learning techniques that are tailored to the problem of bytecode analysis at hand. As a result, you should evaluate your developed methodology for its ability to identify adversarial and vulnerable smart contracts based on their bytecode.

### Requirements

- High: Understanding of programming for Ethereum, EVM and Solidity.
- High: Conceptual thinking and willingness to dive deep into a specific topic.
- Medium: Machine Learning, Artificial Intelligence, Deep Neural Networks.

### Tasks

The thesis is separated into several working packages.

1. Familiarization with existing work & codebase
2. Development & Execution of an automated workflow for incident/attack collection
3. Proposal of a Methodology for Bytecode similarity analysis
4. Implementation of the proposed methodology
5. Experiments: Measure performance of the proposed methodology

### Preliminary Schedule