Abstract:

Smart contracts are self-executing contracts with the terms of the agreement directly written into code, running on blockchain platforms like Ethereum. While they offer a decentralized, secure, and transparent way to enforce agreements, they are also vulnerable to various types of security flaws and attacks. As Ethereum smart contracts gain widespread adoption, ensuring their reliability and security has become crucial, as vulnerabilities can result in financial losses, reputation damage, and legal consequences. Manual code auditing is time-consuming and error-prone, which raises the need for automated detection methods.

This paper proposes the use of **machine learning (ML) techniques** to detect vulnerabilities in Ethereum smart contracts. By leveraging large datasets of known vulnerable and secure smart contract code, the proposed approach utilizes supervised learning models to identify patterns and anomalies indicative of potential vulnerabilities. Key ML methods, such as **decision trees, random forests, support vector machines (SVMs), and deep learning algorithms**, are explored and evaluated for their ability to classify and predict vulnerabilities like reentrancy attacks, integer overflows, and gas limit issues.

The results demonstrate that machine learning models can effectively detect a wide range of vulnerabilities with high accuracy and efficiency compared to traditional static analysis methods. By training the models on labeled datasets containing both secure and vulnerable contract code, the models learn to identify code features and structures associated with known vulnerabilities. Additionally, the paper discusses the importance of feature selection and the challenges involved in collecting and preparing datasets for training.

The findings indicate that machine learning-based detection can significantly improve the scalability, accuracy, and automation of vulnerability identification, enabling developers to enhance the security of Ethereum smart contracts before deployment. This approach has the potential to serve as a foundational tool for securing decentralized applications and mitigating the risks posed by smart contract vulnerabilities.