Abstract

Smart contracts are self-executing programs deployed on blockchain platforms that automate transactions and enforce digital agreements without intermediaries. Despite their transformative potential for decentralized finance, legal automation, and supply chain governance, their adoption is constrained by three interrelated barriers: the programming expertise required to write correct Solidity code excludes non-technical users; the immutable nature of deployed contracts makes undetected vulnerabilities permanently exploitable; and the absence of accessible comprehension tools prevents stakeholders from auditing the code that governs their assets. This thesis addresses these limitations through six contributions organized in three complementary research angles: vulnerability correction, automated contract generation, and visual comprehension.

Along the first angle, the thesis presents an empirical evaluation of ChatGPT (gpt-3.5-turbo) to correct security vulnerabilities identified by the Slither static analyzer in 143 Solidity smart contracts from the SmartBugs-Curated benchmark. The model returned compilable code in 89.8% of cases and corrected for at least one vulnerability in 57.1% of contracts, with a modest improvement of 1.4 percentage points when structured bug descriptions were provided. These results establish that LLMs can function as first-pass patch generators within automated pipelines while simultaneously revealing the ceiling imposed by purely automated evaluation.

Towards the second angle, three studies of increasing scope investigate LLM-based smart contract generation from legal agreements. A GPT-4-build generation pipeline and the CO-STAR prompt engineering methodology achieve 98.1% compilability across 17 prompt templates applied to a property rental agreement. A subsequent expert manual evaluation of 68 generated contracts reveals five categories of critical failure invisible to automated tools, including permanently frozen funds, absent deposit logic, and universal currency mismanagement. A multi-model

comparative study then extends the analysis to four frontier LLMs (GPT-4-Turbo, Claude 3.5, Mistral, and Gemini) across five legal agreements of varying complexity, totaling 480 API calls. Results establish a performance hierarchy (Claude $>$ GPT-4-Turbo $>$ Gemini $>$ Mistral), demonstrate non-linear performance degradation with agreement complexity, and show that Chain-of-Thought prompting yields substantial improvements. The collective conclusion is that current LLMs cannot autonomously generate production-ready smart contracts and that expert review is an indispensable complement to automated metrics.

Along the third angle, the thesis contributes a systematic mapping of 40 peer-reviewed works on smart contract visualization, revealing BPMN as the dominant methodology and identifying a critical gap in formal usability evaluation: only 20% of the tools surveyed report user studies. To address this gap, the thesis presents a UML-based smart contract comprehension tool that integrates automated Slither auditing and AI-powered annotations, evaluated through a controlled study with 36 participants using the NASA-TLX instrument. The tool achieves a composite workload score of 52.62 versus 71.74 for Remix IDE, with particularly pronounced reductions in frustration (49.17 versus 81.39) and mental demand (61.67 versus 78.89).

Collectively, these contributions demonstrate that LLMs and visual tools can meaningfully reduce barriers to smart contract development while establishing that human oversight remains essential for production-quality outcomes.