Large Language Models for Blockchain Security: A Systematic Literature Review

Zheyuan He[a], Zihao Li[b], Sen Yang[a]

[a] *University of Electronic Science and Technology of China, China*
[b] *The Hong Kong Polytechnic University, China*

Abstract

Large Language Models (LLMs) have emerged as powerful tools in various domains involving blockchain security (BS). Several recent studies are exploring LLMs applied to BS. However, there remains a gap in our understanding regarding the full scope of applications, impacts, and potential constraints of LLMs on blockchain security. To fill this gap, we conduct a literature review on LLM4BS.

As the first review of LLM's application on blockchain security, our study aims to comprehensively analyze existing research and elucidate how LLMs contribute to enhancing the security of blockchain systems. Through a thorough examination of scholarly works, we delve into the integration of LLMs into various aspects of blockchain security. We explore the mechanisms through which LLMs can bolster blockchain security, including their applications in smart contract auditing, identity verification, anomaly detection, vulnerable repair, and so on. Furthermore, we critically assess the challenges and limitations associated with leveraging LLMs for blockchain security, considering factors such as scalability, privacy concerns, and adversarial attacks. Our review sheds light on the opportunities and potential risks inherent in this convergence, providing valuable insights for researchers, practitioners, and policymakers alike.

Keywords:
Blockchain Security, Large Language Model, Survey
PACS: 0000, 1111
2000 MSC: 0000, 1111

1. Introduction

As the digital epoch progresses, the confluence of artificial intelligence with blockchain technology emerges as a groundbreaking development, particularly at the juncture where Large Language Models (LLMs) [1, 2, 3, 4] intersect with the ever-evolving domain of blockchain security [5, 6, 7, 8]. LLMs have catapulted to the forefront of natural language processing (NLP), demonstrating profound capabilities in text generation and comprehension—abilities that mirror human-like proficiency. This transformative impact is attributable to their expansive datasets, sophisticated architectures, and the deep neural networks that underpin their operational frameworks [9, 10].

The robustness of LLMs in discerning and synthesizing complex patterns within data positions them as invaluable assets in enhancing the security measures within blockchain systems [11]. The granular analysis of smart contracts, the meticulous scrutiny of transactions, and the proactive monitoring of network behavior are among the critical tasks that LLMs are adept at performing with remarkable efficacy [12, 13].

However, the path to integrating these cognitive powerhouses into blockchain security is met with an array of challenges that beckon for consideration. From navigating the intricate dynamics of ever-advancing cybersecurity threats to addressing the ethical concerns that accompany AI deployment, the trajectory is as demanding as it is promising.

Our work seeks to delve into the multifaceted role of LLMs within the realm of blockchain security, exploring the comprehensive spectrum of their applications, the inherent challenges that surface in this merger, and the prospective future directions that such an integration opens up. As we ebb further into this digital age, the collaborative dance between LLMs and blockchain security is not merely a fleeting trend but a seminal movement, promising to redefine the parameters within which we comprehend, construct, and protect our digital sanctums.

This paper makes the following contributions:

- To the best of our knowledge, and after a meticulous review of the extant literature, we can confidently assert that our work represents the inaugural systematic examination focusing on the application of Large Language Models (LLMs) to tasks within the realm of blockchain security, offering a pioneering exploration of the interplay between advanced AI and cryptographic ledger systems.

- In our comprehensive survey, we meticulously chronicle the current landscape of Large Language Model (LLM) applications in the domain of blockchain security, delving into a detailed analysis of how LLMs are employed across various scenarios, from the enhancement of smart contract reliability to the fortification of distributed ledger integrity, thereby shedding light on the multifaceted contributions of this cutting-edge technology.

- Emanating from our research, we rigorously compile and summarize a suite of practical academic accomplishments pertaining to the application of Large Language Models (LLMs) in fortifying blockchain security, in conjunction with proposing an array of promising and prospective avenues for future research, which we anticipate will catalyze substantial advancements and innovations within this burgeoning intersection of fields.

2. Overview of LLM4BS

We will provide some basic knowledge about LLM4BS tasks.

2.1. Introduction to Large Language Models

This subsection will interpret the definition, characteristics, and diverse applications of Large Language Models (LLMs)

2.1.1. Definition and Characteristics of LLMs

Large Language Models (LLMs) represent a groundbreaking advancement in artificial intelligence, particularly within the domain of natural language processing (NLP) [14]. These models are characterized by their immense size, depth, and complexity, enabling them to process and generate human-like text with remarkable fluency and coherence [15]. At the heart of LLMs lies the transformer architecture, a powerful framework for sequence modeling that has revolutionized the field of NLP [16].

The defining characteristics of LLMs include their unprecedented scale, which involves training on vast corpora of text data containing billions or even trillions of words. This extensive training data allows LLMs to capture the intricate nuances of language, including syntax, semantics, and pragmatics, thereby endowing them with a deep understanding of linguistic structures and conventions [17]. Additionally, LLMs exhibit a high degree of generative

ability, capable of producing text that is contextually relevant and coherent across a wide range of tasks and domains.

Moreover, LLMs possess a remarkable degree of adaptability, thanks to their ability to be fine-tuned or specialized for specific applications or domains through techniques such as transfer learning [18]. By leveraging pre-trained models and fine-tuning them on task-specific datasets, practitioners can tailor LLMs to address a diverse array of NLP tasks, ranging from sentiment analysis and language translation to document summarization and conversational agents [2].

Furthermore, LLMs demonstrate an advanced understanding of context within language, enabling them to generate responses or predictions that are sensitive to the surrounding textual context [19]. This contextual awareness is achieved through mechanisms such as attention mechanisms and positional encodings, which enable LLMs to attend to relevant parts of the input sequence and model long-range dependencies effectively [20].

Overall, LLMs represent a significant milestone in AI research and have unlocked new possibilities for human-computer interaction, content generation, information retrieval, and more. Their ability to understand and generate natural language at scale has led to transformative applications across various domains, shaping the future of AI-driven technologies [21].

2.1.2. Applications of LLMs in Various Domains

The versatility and efficacy of LLMs have led to their widespread adoption across diverse domains and applications, where they have demonstrated exceptional performance and utility [22]. Some notable applications of LLMs include:

Natural Language Understanding (NLU): LLMs excel in tasks such as sentiment analysis, named entity recognition, and text classification, where the comprehension of semantic meaning and context is paramount [23]. By leveraging their deep understanding of language, LLMs can accurately analyze and interpret textual data, enabling tasks such as sentiment analysis in social media monitoring or categorization of customer feedback.

Natural Language Generation (NLG): LLMs are proficient in generating human-like text for a variety of applications, including content creation, dialogue systems, and virtual assistants [24]. Their ability to produce coherent and contextually relevant responses makes them invaluable for tasks such as generating product descriptions, composing personalized messages, or facilitating natural language interactions in conversational interfaces.

Information Retrieval and Summarization: LLMs play a crucial role in extracting relevant information from large volumes of text and generating concise summaries, thereby facilitating efficient information retrieval and knowledge extraction [25]. Whether summarizing news articles, extracting key insights from research papers, or generating abstracts for documents, LLMs offer a powerful solution for distilling vast amounts of textual data into digestible and informative summaries.

Language Translation: LLMs have revolutionized machine translation by providing more accurate and fluent translations across multiple languages [26]. By leveraging their vast linguistic knowledge and contextual understanding, LLMs can produce translations that preserve the meaning, tone, and style of the original text, enabling seamless communication across language barriers in various domains, including e-commerce, international diplomacy, and multicultural communication.

Dialogue Systems: LLMs power conversational agents and chatbots, enabling natural and contextually appropriate interactions with users [27]. Whether assisting customers with product inquiries, providing personalized recommendations, or offering customer support, LLM-based dialogue systems offer a user-friendly and efficient means of communication, enhancing user experience and engagement.

Code Generation: LLMs are increasingly being used to generate code snippets and assist developers in programming tasks by understanding and generating code in various programming languages [24, 28]. By analyzing code repositories and documentation, LLMs can generate code that adheres to programming conventions, syntax rules, and best practices, thereby accelerating the development process and aiding in code maintenance and debugging [29]

Scientific Research: LLMs support scientific discovery by analyzing and summarizing research papers, generating hypotheses, and aiding in data interpretation [30, 22]. By ingesting vast amounts of scientific literature and domain-specific knowledge, LLMs can assist researchers in navigating the ever-expanding body of scientific literature, identifying relevant publications, and extracting valuable insights to inform their research endeavors [31].

These applications underscore the broad utility and transformative potential of LLMs across a wide range of domains and industries, highlighting their significance in advancing AI capabilities and enabling human-computer interaction at unprecedented levels of sophistication. As LLMs continue to evolve and improve, their impact on various fields is expected to grow, driving

innovation, efficiency, and discovery in the years to come.

2.2. Blockchain Security Fundamentals

This section will discuss the key components and common security threats of blockchain systems.

2.2.1. Key Components of Blockchain Security

Blockchain security is a multifaceted endeavor aimed at safeguarding the integrity, confidentiality, and availability of data stored and processed within a blockchain network. Key components of blockchain security include:

Cryptography: Cryptography lies at the heart of blockchain security, serving to encrypt data, authenticate participants, and ensure the integrity of transactions [32, 33]. Techniques such as hashing, digital signatures, and cryptographic keys are utilized to secure data and verify the authenticity of transactions on the blockchain [34].

Consensus Mechanisms: Consensus mechanisms are protocols that govern how transactions are validated and added to the blockchain. By achieving agreement among network participants, consensus mechanisms ensure the immutability and integrity of the distributed ledger [35, 36]. Popular consensus mechanisms include Proof of Work (PoW) [37], Proof of Stake (PoS) [38], and Delegated Proof of Stake (DPoS) [39], each with its own strengths and vulnerabilities.

Decentralization: Decentralization is a core principle of blockchain security, distributing control and decision-making authority across a network of nodes [40, 41]. By eliminating single points of failure and reducing the risk of censorship or manipulation, decentralization enhances the resilience and security of the blockchain network [42]. However, achieving true decentralization requires careful consideration of factors such as node distribution, governance structures, and network incentives [43].

Smart Contract Security: Smart contracts are self-executing contracts with predefined rules and conditions encoded on the blockchain. Ensuring the security of smart contracts is essential to prevent vulnerabilities, exploits, and unauthorized access [44, 45]. Techniques such as formal verification, code auditing, and secure development practices are employed to mitigate risks associated with smart contracts, including reentrancy attacks, integer overflow/underflow, and unchecked external calls [46].

2.2.2. Common Security Threats in Blockchain Systems

Despite the robust security measures inherent in blockchain technology, various security threats and vulnerabilities pose risks to the integrity and functionality of blockchain systems [47, 48]. Some common security threats in blockchain systems include:

Consensus-Based Attacks: Consensus-based attacks exploit vulnerabilities in the consensus mechanism to compromise the integrity or availability of the blockchain network [49]. Examples include 51% attacks, where a single entity or coalition controls the majority of the network's hash rate, enabling them to manipulate transaction confirmations or execute double spending attacks [50]. Similarly, attacks such as selfish mining, eclipse attacks, and long-range attacks target weaknesses in specific consensus protocols, undermining the security and reliability of the blockchain network [51].

Smart Contract Exploits: Smart contract vulnerabilities pose significant risks to blockchain security, as they can be exploited to execute unauthorized transactions, drain funds, or trigger unintended behavior [52, 53]. Common smart contract vulnerabilities include reentrancy attacks, where an attacker repeatedly calls a vulnerable contract's function before the previous invocation completes, enabling them to manipulate the contract's state and steal funds [54]. Other vulnerabilities, such as integer overflow/underflow, unchecked external calls, and gas limit vulnerabilities, can also be exploited to compromise the security of smart contracts and the underlying blockchain network [55].

DeFi Protocol Vulnerabilities: Decentralized finance (DeFi) protocols introduce new security challenges due to their complex interactions and composability [56, 56]. Vulnerabilities in DeFi protocols, such as flash loan attacks, oracle manipulation, and governance exploits, can result in significant financial losses for users and undermine trust in the DeFi ecosystem [57, 58]. Additionally, vulnerabilities in specific DeFi protocols can have cascading effects on other interconnected protocols, amplifying the impact of security breaches and systemic risks within the DeFi space [59].

Auxiliary Service Vulnerabilities: Auxiliary services, such as wallets, exchanges, oracles, and decentralized applications (DApps), serve as entry points for attackers to exploit vulnerabilities and compromise the security of blockchain systems [60]. Security breaches in auxiliary services, such as exchange hacks, wallet vulnerabilities, or oracle manipulation attacks, can lead to the loss of funds, unauthorized access to user data, or manipulation

of on-chain transactions [61, 62]. Furthermore, the interconnected nature of auxiliary services within the blockchain ecosystem amplifies the impact of security breaches, as vulnerabilities in one service can propagate to others, resulting in widespread disruption and financial losse.

Addressing these security threats and vulnerabilities requires a comprehensive approach that encompasses technical measures, best practices, and community collaboration to strengthen the resilience and security of blockchain systems [63]. By understanding the key components of blockchain security and mitigating common security threats, stakeholders can foster greater trust, transparency, and adoption in the decentralized ecosystem, driving innovation and value creation for users worldwide.

3. Taxonomy of LLM4BS task

In this section, we introduce a thematic taxonomy devised to systematically categorize the body of literature about tasks associated with large language models for blockchain security (LLM4BS), emphasizing the function of the LLM within these contexts. Fig. 1 depicts the five applications of LLM4BS task.

3.1. LLM as Code auditor on Smart Contracts

The application of LLM in the domain of smart contract code auditing and vulnerability detection can be succinctly encapsulated as follows: Advanced tools [64, 65, 66, 67, 68, 69, 70, 71, 72] powered by Large Language Models, such as GPTScan [65] and SMARTINV [64], signify a monumental shift from traditional, pattern-based analysis methodologies towards more contextually aware and comprehensive inspection techniques. These cutting-edge tools extend their analytical prowess beyond static patterns by knitting together disparate threads of information, including the nuanced aspects of natural language documentation that detail the intended functions and transactional constructs of smart contracts.

This marriage of code and contextual data through a multimodal lens equips such tools with the capacity to unravel complex logical oversights and identify subtle "machine un-auditable" bugs, which would otherwise evade detection. By assimilating and interpreting the richer tapestry of human language explanations paired with code, LLM-based tools delve deeper into the intricate web of smart contract interactions. The profound understanding garnered from this approach not only sheds light on hidden vulnerabilities

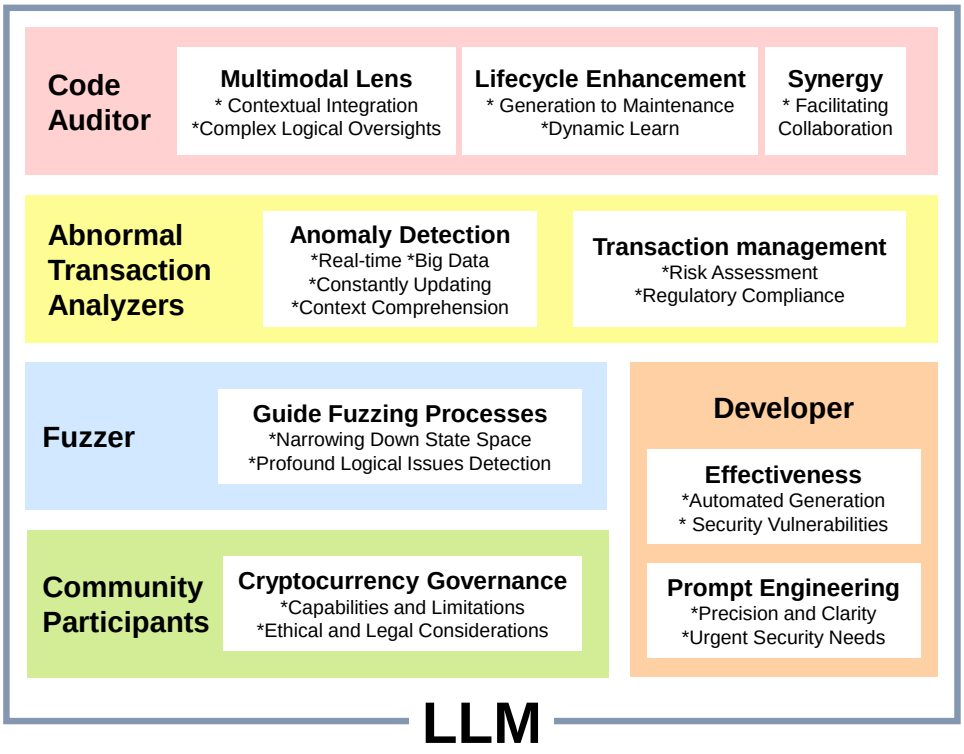

Figure 1: The applications of LLM on the task of blockchain security.

but also fortifies smart contracts against the myriad of risks that could lead to substantial financial repercussions.

In essence, the integration of Large Language Models in smart contract analysis marks a significant leap in safeguarding blockchain technology's infrastructural integrity. It underscores an evolving landscape where artificial intelligence converges with software development practices to bolster security measures. This proactive identification and remediation of weaknesses within smart contracts, facilitated by the keen insights offered by LLMs, are instrumental in cementing trust and reliability in blockchain transactions—hence mitigating potential financial liabilities and reinforcing the bedrock of digital contracts.

Expanding further on the key roles LLMs play, it's worth noting the vast potential these models have in enhancing the entire lifecycle of smart contract development. From generation to maintenance, LLMs facilitate the crafting of more secure and robust smart contracts. They do so by potentially

providing recommendations during the development phase, suggesting best practices, and even generating code snippets that align with security guidelines. Throughout the auditing process, tools like GPTScan and SMARTINV can continuously learn and adapt to new patterns of vulnerabilities emerging from the evolving landscape of blockchain technology and cyber threats. This dynamic learning process is pivotal, as it allows for the development of increasingly refined models capable of detecting even the most covert and sophisticated vulnerabilities.

Moreover, the capacity of LLMs to assimilate context and understand code as it correlates to business logic makes them particularly effective in scenarios where contractual agreements are complex and layered with intricate logic. This is especially crucial in fields such as finance, where smart contracts govern transactions involving significant sums and numerous stakeholders. The vulnerability in such a domain could have catastrophic effects, not just financially but also in terms of reputational damage for the entities involved. Hence, the stakes in accurate and effective smart contract auditing cannot be overstated.

LLMs also enhance collaborative efforts throughout the industry by facilitating a common understanding among developers, auditors, and end-users. Their ability to parse and explain code in natural language bridges communication gaps, enabling stakeholders with varying levels of technical expertise to engage in meaningful dialogue regarding the security and functionality of smart contracts. This collaborative environment fosters a culture of shared responsibility and proactive engagement in addressing and preempting security concerns.

3.2. LLM as Analyzers for abnormal transaction

The application of LLMs for blockchain transaction analysis [73, 74] underscores their crucial role in conducting real-time monitoring to detect signs of irregular or suspicious behavior. These models represent a significant advancement in the field, as they provide a more dynamic and adaptable approach to identifying potential threats within blockchain transactions.

Unlike static, rule-based systems, LLMs are capable of processing and learning from vast amounts of transaction data in real-time, which enables them to uncover not just known types of fraudulent activity, but also novel patterns that emerge as technology and attack methods evolve. By leveraging the power of machine learning, these models can constantly update their understanding of what constitutes normal transactional behavior. This

continuous learning process is essential for adapting to the ever-changing landscape of blockchain technology and the complex strategies employed by malicious actors.

Furthermore, the adaptability of LLMs is not limited to pattern recognition—they also excel in understanding the context of transactions. This includes the analysis of smart contract interactions, execution traces, gas prices, and other transaction metadata that could provide hints about the legitimacy of a transaction. Contextual analysis allows LLMs to differentiate between legitimate, though unusual, transactional behavior and genuine anomalies that could indicate fraudulent activities, such as money laundering, phishing, or exploitation of contract vulnerabilities.

In addition to identifying potentially fraudulent transactions, LLMs also contribute to risk assessment and regulatory compliance. By analyzing the transaction data against current compliance standards and risk models, LLMs can assist financial institutions in managing their risk exposure and adhering to anti-money laundering (AML) and know your customer (KYC) regulations. Their sophisticated analysis capabilities can provide valuable insights to compliance officers and regulatory bodies, allowing for a more proactive approach to detecting and preventing financial crimes.

In summary, the application of LLMs in blockchain transaction analysis reflects a commitment to enhancing the security measures of digital financial systems. By combining deep learning algorithms with extensive transaction datasets, LLMs stand as a formidable line of defense, capable of not only identifying anomalous activities in real-time but also evolving with the advancing threats, ensuring a resilient and secure framework for managing blockchain-based transactions.

3.3. LLM as Fuzzer for Smart Contract

Large Language Models (LLMs) have been increasingly employed to elevate the process of fuzzing, particularly in the realm of smart contract security analysis [75, 76]. This methodology involves utilizing LLMs to accurately assess the complexity and vulnerability likelihood of specific code regions within a smart contract. Consequently, these metrics serve to guide the direction and focus of fuzzers, steering them towards code segments that are more likely to harbor potential security threats.

The application of LLMs to fuzzing exercises significantly elevates the efficiency of these operations by narrowing down the vast state space that fuzzers typically navigate. This precision-targeted fuzzing approach contributes to

higher coverage and reveals more vulnerabilities than conventional tools, especially those pertaining to the intricate nature of smart contract code that traditional methods may overlook.

Moreover, this refined fuzzing technique allows for the integration of user-defined invariants—manually inserted assertions to monitor and manage the state during fuzzing. By doing so, it reduces the exploration overhead and improves the detection of more profound logical issues that regular fuzzing routines might miss. Evaluations of this LLM-enhanced fuzzing method within real-world decentralized finance (DeFi) projects have demonstrated its effectiveness, outperforming baseline fuzzing parameters and uncovering significant vulnerabilities. These vulnerabilities, if left undetected and exploited, could potentially result in substantial financial losses.

In summary, the fusion of LLMs into the fuzzing workflow offers a promising and intelligent solution to the challenges faced in automated security analysis of smart contracts, underscoring their potential for increasing the robustness of blockchain-based platforms.

3.4. LLM as Developer for Smart Contract

Recent studies [77, 67, 78, 79, 80] have begun to scrutinize the efficacy and reliability of Large Language Models (LLMs) like ChatGPT and Google Palm2 in the automated generation of smart contracts. These smart contracts are integral to the blockchain ecosystem, executing agreements without the need for intermediaries, and their accuracy and security are paramount. The research primarily constructs a testing framework that assesses smart contracts on multiple fronts — validity, correctness, efficiency, security, and maintainability.

These results have demonstrated that LLMs, despite showing proficiency in understanding contractual terms and generating syntactically correct Solidity code, often produce contracts with considerable security vulnerabilities. This finding signals a critical issue in the code's operational quality. The evaluations suggest that while LLMs can streamline the contract creation process, there's an underlying risk of generating code that could be exploited if used without a thorough review.

Importantly, the studies underscore the role of effective prompt engineering. It emerged that the LLMs' outputs are significantly influenced by the specificity and clarity of the prompts, which must be meticulously designed to minimize the risk of ambiguous or flawed code generation. This is particularly challenging because generating smart contracts requires precision,

and the semantics of legal terms must be correctly interpreted and applied by the models.

These works point to the necessity for comprehensive analysis and improvement in the methodologies employed by LLMs. There is optimism that future iterations of LLMs, with better training and prompt-design considerations, could enhance the quality and security of AI-generated smart contracts. It also hints at the potential for these tools to revolutionize contract generation by reducing the time and effort required, while flagging the urgent need for more robust security measures and testing methods.

Such research analysis provides an overarching view of the current state of LLM applications in smart contract generation. The discoveries made serve as a cautionary note about over-reliance on AI without adequate checks but also lay out a roadmap for future advancements that could harness AI's full potential responsibly.

3.5. LLM as participants for Cryptocurrency community

Large Language Models (LLMs) such as GPT-3.5 and ChatGPT are emerging as powerful tools in the cryptocurrency community [81, 82, 83, 84**?**], albeit with their respective strengths and weaknesses. Related works collectively depict a landscape where LLMs are being explored for their potential to revolutionize governance and legal processes within the high-stakes, highly volatile realm of cryptocurrency.

Governance emerges as a major theme, as LLMs could contribute significantly to the structuring and transparency of this largely unregulated space. The first document outlines the broader governance challenges faced by AI systems, suggesting blockchain as a viable solution to introduce verifiability and accountability. On the other hand, limitations of LLMs in capturing the complexities of legal reasoning are highlighted, a concern that is echoed across the three studies to varying degrees.

The practical applications of these models in legal settings, specifically detailed in the second and third documents, emphasize their innovative role in drafting legal complaints. This development is promising for the future of legal work related to cryptocurrency regulations and litigation, as it suggests that LLMs could alleviate some of the workload from human experts, although the need for human oversight remains.

While governance and legal assistance dominate the discourse, there's a tone of cautious optimism throughout the texts. There is recognition of the transformative potential of LLMs in the cryptocurrency sector, but also a

clear acknowledgment of the need for further advancement in AI technology to fully integrate into complex decision-making processes where legal and ethical considerations are paramount.

In essence, the collective narrative from the three documents converges on the premise that LLMs hold transformative potential for the cryptocurrency community's governance and legal sectors but must overcome challenges in understanding and application before they can be fully trusted in autonomous roles.

3.6. Miscellaneous

In addition, LLM is also used in blockchain security fields, involving smart contract compilers [85], zero-knowledge proofs [86], model training [87], NFT generation [88]. We will introduce their applications in detail in future work.

4. Case study of LLM4BS

In this section, we engage in an in-depth examination through three distinct case studies, each serving to illustrate and shed light on the diverse and concrete applications of Large Language Models for Blockchain Systems (LLM4BS). These cases have been meticulously selected to encompass a broad range of scenarios.

4.1. LLM4Fuzz

LLM4FUZZ [75] emerges as an innovative technique in the cybersecurity landscape, specifically in the niche of smart contract security within blockchain networks. It intricately combines the prowess of Large Language Models (LLMs) with fuzz testing methodologies to proactively unearth vulnerabilities that could potentially compromise the integrity of smart contracts.

LLMs are highly sophisticated AI models that have made significant strides in understanding and generating human-like text, and more recently, they have proven to be adept at comprehending programming languages and code structure. LLM4FUZZ exploits this capacity by deploying LLMs to guide fuzzing processes intelligently. This results in a more incisive and nuanced exploration of smart contracts, focusing testing efforts on areas that LLMs determine to be most likely to contain security flaws. By doing so, LLM4FUZZ succeeds in not only streamlining the anomaly detection process but also in enhancing its accuracy and depth.

In the world of blockchain technology, where smart contracts serve as immutable agreements that execute automatically based on coded conditions, the potential negative impact of a security breach is heightened. Smart contracts control significant digital assets and are essential to the functioning of distributed applications (dApps). The immutable nature of blockchain adds a layer of complexity as deployed smart contracts, once committed to the blockchain, cannot be altered. Therefore, preemptive security assurances become crucial to ensuring their reliability and safeguarding the assets and processes they govern.

LLM4FUZZ provides a novel layer of security analysis by identifying and prioritizing potential problem areas within smart contract code. This prioritization is achieved through the LLM's learned understanding of code patterns that are historically or commonly associated with vulnerabilities. The methodology enhances traditional fuzzing strategies, which typically adopt a more scattergun approach by bombarding the code with random data inputs. LLM4FUZZ's targeted testing is not just more efficient but also more effective in discovering complex vulnerabilities that might otherwise be missed.

Following implementation, LLM4FUZZ has been benchmarked against existing fuzzing techniques and has consistently demonstrated superior performance. It expedites the vulnerability detection process and increases the breadth of security flaws that can be detected, thereby reinforcing the overall security posture.

The case of LLM4FUZZ is emblematic of the foresight in AI integration into cybersecurity regimes. It encapsulates the transformative effects of AI on improving and redefining existing technological processes, particularly in areas critical to the burgeoning digital economy. Through its lens, we catch a glimpse of the future of smart contract security – a future where AI-driven tools not only anticipate but actively engage in the continuous battle against cyber threats.

4.2. SMARTINV

Proposed with the intention of enhancing the reliability and security of blockchain smart contracts, SMARTINV [64] represents a significant breakthrough in the field. Its primary function is to infer invariants within smart contracts which can be integral in automating the process of identifying elusive bugs that typically elude conventional machine-auditing methods.

The unique aspect of SMARTINV lies in its multimodal learning strategy, which acknowledges that truly understanding the operational behavior

of smart contracts requires a multifaceted approach—one that combines and analyzes different types of information, or modalities. SMARTINV specifically leverages both the static code within a smart contract and dynamic transaction data. By correlating code patterns with transaction behaviors, SMARTINV is poised to uncover invariant conditions that point to a smart contract's expected and intended state throughout its lifecycle. This holistic approach ensures a more thorough examination and superior detection rate of potential security weaknesses that could lead to future vulnerabilities and exploits.

The framework operates on the premise that no singular mode of information can fully articulate a smart contract's intricate logic and potential edge cases. Hence, by fusing multiple data sources, SMARTINV captures a more accurate depiction of a smart contract's functionality, leading to a significant reduction in false positives and more precise bug detection. Such an integrated approach to smart contract analysis promotes greater assurance in their deployment and operation, which is a critical concern in blockchain applications where security and trust are paramount.

In deploying SMARTINV, the researchers demonstrate its efficacy by testing on a collection of smart contracts, where it shows not only a high degree of accuracy but also an impressive capability in scalability. SMARTINV emerges as an invaluable asset in the realm of smart contract development and auditing, setting a precedent for future methodologies to build upon its multimodal analysis framework for enhanced security measures in the ever-evolving domain of blockchain technology.

4.3. BLOCKGPT

BLOCKGPT [73] serves as a paradigm shift in the domain of blockchain security, acting as a state-of-the-art Intrusion Detection System (IDS) specifically engineered to counteract and identify potentially malicious transactions within blockchain networks. The system is underpinned by a highly sophisticated large language model that has been meticulously trained with a significant corpus of transactional data from the Ethereum blockchain, one of the most widely utilized platforms in the industry.

The innovation expressed by BLOCKGPT is its departure from traditional detection methodologies that largely depend on predetermined rules or known patterns. Instead, BLOCKGPT adopts a proactive and learning-based approach that enables it to recognize a spectrum of anomalies, includ-

ing sophisticated and previously unseen threats that could bypass conventional rule-based systems.

Demonstrating the prowess of its detection capabilities, BLOCKGPT has proven remarkably successful in testing scenarios. It proficiently identified and appropriately ranked 49 out of 124 verified attack transactions among the most abnormal three transactions that have occurred within their respective victim contracts. This high level of precision points to the system's refined anomaly recognition algorithms, indicating substantial progress in the field of IDS for blockchain.

Beyond its detection accuracy, the efficiency of BLOCKGPT is exemplified by its processing speed, handling transactions at an average rate of 2,284 per second, with relatively minimal deviation. This capability is not merely theoretical but is indicative of the system's readiness for deployment in real-world blockchain environments where real-time monitoring and response are critical.

The adaptability of BLOCKGPT extends to various blockchain architectures and applications, from finance to smart contracts. This versatility, combined with its real-time processing faculties, provides a robust and scalable solution that can be integrated seamlessly into existing blockchain infrastructures to fortify their resilience against a wide array of security threats.

As blockchain technology continues its integration into the fabric of digital transactions and smart contract deployment, systems such as BLOCKGPT represent vital components in the ongoing effort to safeguard these platforms. With the adoption of machine learning models like the one upon which BLOCKGPT is built, the future of blockchain IDS appears increasingly secure, paving the way for safer and more reliable blockchain operations.

5. Future Direction and Challenge of LLM4BS

In delving into the future of Large Language Models for Blockchain Security (LLM4BS), the academic community contends with a series of pivotal focus areas that necessitate concerted scholarly efforts to address inherent challenges and extend LLM's utility in blockchain systems. The following focal points are elaborated to reflect the nuances and complexity inherent in this field of study:

Interdisciplinary Relationships: The essence of the next stage in LLM4BS is undeniably grounded in a harmonized interplay among the domains of artificial intelligence, cyber protection mechanisms, and distributed

ledger technologies [89, 90]. This interdisciplinary collaboration is not merely additive but synergistic, as it draws upon the strengths and insights of each discipline to forge a formidable shield against cyber animosities. There is a clarion call within the academic and industrial spheres for a robust alliance, emphasizing that the amalgamation of cognitive computing with cryptographic resilience and decentralized architectures can lead to a paradigm shift in securing blockchain networks.

Regulatory and Compliance Challenges: The shifting sands of regulatory frameworks demand not only compliance but a proactive engagement with regulatory bodies by scholars and practitioners in the LLM4BS field [7, 91]. This relationship is reciprocal; as regulatory agencies develop deeper understandings of the implications of integrating AI in blockchain, it is incumbent upon the actors within this space to advocate for regulations that encourage innovation while maintaining robust security measures. The dynamic interplay between cutting-edge technology and regulation is a delicate balance to strike, fostering a stable yet flexible platform for growth and adaptation in blockchain security solutions.

Dynamic Security Threats: The cyber threat horizon is akin to a chimeric beast—constantly mutating and presenting unforeseen challenges [10, 92]. Security models like LLM4BS must be engineered with inherent plasticity, allowing them to evolve alongside the threats they are designed to counteract. The integration of LLMs in blockchain security is not a static solution but a continually adapting safeguard, necessitating an expansive approach to cybersecurity that accounts for the proliferation of sophisticated cyberattacks as well as the subtleties of targeted breaches. Sustaining the integrity of blockchain transactions hinges on the preemptive identification and neutralization of these mercurial threats.

Ethical Governance and Bias Mitigation: The ethical tapestry within which LLM4BS operates is rich and complex, mandating a conscientious approach towards the examination and resolution of security practices that may inadvertently propagate bias or unfair outcomes [93, 94]. The quest for equitable algorithms expands beyond the technical realm, engaging with sociocultural dynamics and the moral dimensions of technological deployments. Therefore, a concerted effort in research that transcends statistical bias mitigation, touching upon philosophy, sociology, and ethics, is essential for fostering a climate where AI not only fortifies security but does so with an underlying commitment to justice and fairness.

Energy Considerations and AI Sustainability: In addressing the

carbon footprint of blockchain operations, there is also a pressing need to confront the energy-intensive nature of training and deploying Large Language Models [95, 96, 97]. The ecological impact of these AI systems necessitates a dual strategy: enhancing algorithmic efficiency to reduce computational load and exploring alternative energy sources that can power these activities sustainably. This pursuit of ecological harmonization in the application of LLM4BS must be reflective of a broader commitment to sustainability across all aspects of blockchain technology, ensuring that the acceleration of security capabilities does not come at an unsustainable environmental cost.

Ethical Considerations in AI: The role of ethics cannot be overstated in the trajectory of LLM4BS implementation, as it undergirds every facet of AI application—from the source of data to the transparency of algorithms and the accountability for decisions made by or with the aid of AI. Implementing a robust ethical framework for LLM4BS entails a deep interrogation of the principles guiding AI development, encouraging scrutiny that permeates every layer of model design, deployment, and monitoring. Thus, creating an environment where trust in AI-fueled security measures is not merely assumed but carefully cultivated through responsible practices.

Data Quality and Access: At the heart of robust LLM4BS deployments lies the foundational element of data—its caliber, its scope, and the accessibility afforded to it. Herein lies the challenge: constructing and maintaining databases that are not only comprehensive and representative but are also curated with an eye towards enhancing the efficacy of Large Language Models in detecting anomalies and reinforcing security parameters in blockchain transactions. The task extends to crafting protocols that ensure data integrity and sourcing that conforms to ethical standards, thereby upholding the sanctity and reliability of these AI systems.

Navigating these considerations requires a strategic, methodological approach to utilize the full promise of LLM4BS. This involves a commitment to ongoing research, rigorous ethical scrutiny, and a concerted effort to evolve in tandem with the technological and regulatory landscape. With a fundamental understanding of these points, the community is better equipped to pave the way for LLM4BS to enhance the resilience and efficiency of blockchain security measures.

6. Conclusion

In summing up this review on the incorporation of Large Language Models (LLMs) into blockchain security, we have traversed through the technological advancements and intricate challenges presented by this union. The potential of LLMs to augment security protocols in blockchain is clear, offering innovative solutions for smart contract auditing, identity verification, and anomaly detection. Yet, this potential comes with the necessity for vigilance regarding scalability, privacy, advancing cyber threats, and ethical implications of AI. As we look ahead, the success of LLMs in blockchain security depends not only on continuous technological refinement but also on ethical practices, regulatory alignment, and informed community engagement. The integration of LLM into blockchain heralds a transformative era in digital security that demands a collaborative approach, balancing innovation with prudent oversight to forge a resilient and equitable digital future.

References

[1] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, et al., Chatgpt for good? on opportunities and challenges of large language models for education, Learning and individual differences 103 (2023) 102274.

[2] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, M. Fritz, Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection, in: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, 2023, pp. 79–90.

[3] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, Y. Zhuang, Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face, Advances in Neural Information Processing Systems 36 (2024).

[4] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, Y. Zhang, A survey on large language model (llm) security and privacy: The good, the bad, and the ugly, High-Confidence Computing (2024) 100211.

[5] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, ACM Transactions on Intelligent Systems and Technology (2023).

[6] J. Liu, C. S. Xia, Y. Wang, L. Zhang, Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, Advances in Neural Information Processing Systems 36 (2024).

[7] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, Y. Tang, A brief overview of chatgpt: The history, status quo and potential future development, IEEE/CAA Journal of Automatica Sinica 10 (2023) 1122–1136.

[8] W. Ma, S. Liu, W. Wang, Q. Hu, Y. Liu, C. Zhang, L. Nie, Y. Liu, The scope of chatgpt in software engineering: A thorough investigation, arXiv preprint arXiv:2305.12138 (2023).

[9] S. Hu, T. Huang, F. İlhan, S. F. Tekin, L. Liu, Large language model-powered smart contract vulnerability detection: New perspectives, arXiv preprint arXiv:2310.01152 (2023).

[10] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, et al., Summary of chatgpt-related research and perspective towards the future of large language models, Meta-Radiology (2023) 100017.

[11] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future generation computer systems 107 (2020) 841–853.

[12] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. C. Grundy, H. Wang, Large language models for software engineering: A systematic literature review, ArXiv abs/2308.10620 (2023). URL: https://api.semanticscholar.org/CorpusID:261048648.

[13] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, J. rong Wen, A survey of large language models, ArXiv abs/2303.18223 (2023). URL: https://api.semanticscholar.org/CorpusID:257900969.

[14] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, H. Wang, Large language models for software engineering: A systematic literature review, arXiv preprint arXiv:2308.10620 (2023).

[15] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, Y. Su, Llm-planner: Few-shot grounded planning for embodied agents with large language models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 2998–3009.

[16] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, Q. Yang, Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023, pp. 1–21.

[17] S. Kang, J. Yoon, S. Yoo, Large language models are few-shot testers: Exploring llm-based general bug reproduction, in: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), IEEE, 2023, pp. 2312–2323.

[18] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 328–339.

[19] P. Yin, G. Neubig, W.-t. Yih, S. Riedel, Tabert: Pretraining for joint understanding of textual and tabular data, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8413–8426.

[20] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, R. Salakhutdinov, Transformer-xl: Attentive language models beyond a fixed-length context, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019.

[21] M. Cheng, T. Piccardi, D. Yang, Compost: Characterizing and evaluating caricature in llm simulations, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 10853–10875.

[22] S.-C. Dai, A. Xiong, L.-W. Ku, Llm-in-the-loop: Leveraging large language model for thematic analysis, arXiv preprint arXiv:2310.15100 (2023).

[23] G. Kim, H. Lee, D. Kim, H. Jung, S. Park, Y. Kim, S. Yun, T. Kil, B. Lee, S. Park, Visually-situated natural language understanding with contrastive reading model and frozen large language models, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.

[24] A. Ni, S. Iyer, D. Radev, V. Stoyanov, W.-t. Yih, S. Wang, X. V. Lin, Lever: Learning to verify language-to-code generation with execution, in: International Conference on Machine Learning, PMLR, 2023, pp. 26106–26128.

[25] N. Mishra, G. Sahu, I. Calixto, A. Abu-Hanna, I. H. Laradji, Llm aided semi-supervision for efficient extractive dialog summarization, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.

[26] F. Liu, J. M. Eisenschlos, F. Piccinno, S. Krichene, C. Pang, K. Lee, M. Joshi, W. Chen, N. Collier, Y. Altun, Deplot: One-shot visual language reasoning by plot-to-table translation, arXiv preprint arXiv:2212.10505 (2022).

[27] X. L. Dong, S. Moon, Y. E. Xu, K. Malik, Z. Yu, Towards next-generation intelligent assistants leveraging llm techniques, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5792–5793.

[28] Q. Gu, Llm-based code generation method for golang compiler testing, in: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023, pp. 2201–2203.

[29] P. Vaithilingam, T. Zhang, E. L. Glassman, Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models, in: Chi conference on human factors in computing systems extended abstracts, 2022, pp. 1–7.

[30] A. Agossah, F. Krupa, M. Perreira Da Silva, P. Le Callet, Llm-based interaction for content generation: A case study on the perception of employees in an it department, in: Proceedings of the 2023 ACM International Conference on Interactive Media Experiences, 2023, pp. 237–241.

[31] N. Sultanum, A. Srinivasan, Datatales: Investigating the use of large language models for authoring data-driven articles, in: 2023 IEEE Visualization and Visual Analytics (VIS), IEEE, 2023, pp. 231–235.

[32] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, in: 2016 IEEE symposium on security and privacy (SP), IEEE, 2016, pp. 839–858.

[33] L. Tan, K. Yu, C. Yang, A. K. Bashir, A blockchain-based shamir's threshold cryptography for data protection in industrial internet of things of smart city, in: Proceedings of the 1st Workshop on Artificial Intelligence and Blockchain Technologies for Smart Cities with 6G, 2021, pp. 13–18.

[34] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, J. Wang, Untangling blockchain: A data processing view of blockchain systems, IEEE transactions on knowledge and data engineering 30 (2018) 1366–1385.

[35] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, A. Rindos, Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric), in: 2017 IEEE 36th symposium on reliable distributed systems (SRDS), IEEE, 2017, pp. 253–255.

[36] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, M. A. Imran, A scalable multi-layer pbft consensus for blockchain, IEEE Transactions on Parallel and Distributed Systems 32 (2020) 1146–1160.

[37] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, S. Capkun, On the security and performance of proof of work blockchains, in: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 3–16.

[38] P. Gaži, A. Kiayias, D. Zindros, Proof-of-stake sidechains, in: 2019 IEEE Symposium on Security and Privacy (SP), IEEE, 2019, pp. 139–156.

[39] W. Li, S. Andreina, J.-M. Bohli, G. Karame, Securing proof-of-stake blockchain protocols, in: Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2017 International Workshops,

DPM 2017 and CBT 2017, Oslo, Norway, September 14-15, 2017, Proceedings, Springer, 2017, pp. 297–315.

[40] M. Conoscenti, A. Vetro, J. C. De Martin, Peer to peer for privacy and decentralization in the internet of things, in: 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), IEEE, 2017, pp. 288–290.

[41] G. D. Monte, D. Pennino, M. Pizzonia, Scaling blockchains without giving up decentralization and security: A solution to the blockchain scalability trilemma, in: Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems, 2020, pp. 71–76.

[42] G. Zyskind, O. Nathan, et al., Decentralizing privacy: Using blockchain to protect personal data, in: 2015 IEEE security and privacy workshops, IEEE, 2015, pp. 180–184.

[43] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, R. H. Deng, Crowdbc: A blockchain-based decentralized framework for crowdsourcing, IEEE transactions on parallel and distributed systems 30 (2018) 1251–1266.

[44] W. Zou, D. Lo, P. S. Kochhar, X.-B. D. Le, X. Xia, Y. Feng, Z. Chen, B. Xu, Smart contract development: Challenges and opportunities, IEEE Transactions on Software Engineering 47 (2019) 2084–2106.

[45] K. Hu, J. Zhu, Y. Ding, X. Bai, J. Huang, Smart contract engineering, Electronics 9 (2020) 2042.

[46] L. Zhou, X. Xiong, J. Ernstberger, S. Chaliasos, Z. Wang, Y. Wang, K. Qin, R. Wattenhofer, D. Song, A. Gervais, Sok: Decentralized finance (defi) attacks, in: 2023 IEEE Symposium on Security and Privacy (SP), IEEE, 2023, pp. 2444–2461.

[47] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, Y. Bian, Blockchain security: A survey of techniques and research directions, IEEE Transactions on Services Computing 15 (2020) 2490–2510.

[48] D. Berdik, S. Otoum, N. Schmidt, D. Porter, Y. Jararweh, A survey on blockchain for information systems management and security, Information Processing & Management 58 (2021) 102397.

[49] Z. Ma, L. Liu, W. Meng, Towards multiple-mix-attack detection via consensus-based trust management in iot networks, Computers & Security 96 (2020) 101898.

[50] G. Xu, H. Bai, J. Xing, T. Luo, N. N. Xiong, X. Cheng, S. Liu, X. Zheng, Sg-pbft: A secure and highly efficient distributed blockchain pbft consensus algorithm for intelligent internet of vehicles, Journal of Parallel and Distributed Computing 164 (2022) 1–11.

[51] Y. Xiao, N. Zhang, W. Lou, Y. T. Hou, Modeling the impact of network connectivity on consensus security of proof-of-work blockchain, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1648–1657.

[52] L. Brent, N. Grech, S. Lagouvardos, B. Scholz, Y. Smaragdakis, Ethainter: a smart contract security analyzer for composite vulnerabilities, in: Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation, 2020, pp. 454–469.

[53] Z. Wan, X. Xia, D. Lo, J. Chen, X. Luo, X. Yang, Smart contract security: A practitioners' perspective, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE, 2021, pp. 1410–1422.

[54] T. Sharma, Z. Zhou, A. Miller, Y. Wang, A {Mixed-Methods} study of security practices of smart contract developers, in: 32nd USENIX Security Symposium (USENIX Security 23), 2023, pp. 2545–2562.

[55] M. Coblenz, J. Sunshine, J. Aldrich, B. A. Myers, Smarter smart contract development tools, in: 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), IEEE, 2019, pp. 48–51.

[56] S. Chaliasos, M. A. Charalambous, L. Zhou, R. Galanopoulou, A. Gervais, D. Mitropoulos, B. Livshits, Smart contract and defi security tools: Do they meet the needs of practitioners?, in: Proceedings of the 46th IEEE/ACM International Conference on Software Engineering, 2024, pp. 1–13.

[57] L. Zhou, K. Qin, A. Cully, B. Livshits, A. Gervais, On the just-in-time discovery of profit-generating transactions in defi protocols, in:

2021 IEEE Symposium on Security and Privacy (SP), IEEE, 2021, pp. 919–936.

[58] Y. Wang, P. Zuest, Y. Yao, Z. Lu, R. Wattenhofer, Impact and user perception of sandwich attacks in the defi ecosystem, in: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, 2022, pp. 1–15.

[59] Q. Kong, J. Chen, Y. Wang, Z. Jiang, Z. Zheng, Defitainter: Detecting price manipulation vulnerabilities in defi protocols, in: Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis, 2023, pp. 1144–1156.

[60] K. Li, J. Chen, X. Liu, Y. R. Tang, X. Wang, X. Luo, As strong as its weakest link: How to break blockchain dapps at rpc service., in: NDSS, 2021.

[61] S. Kim, S. Hwang, Etherdiffer: Differential testing on rpc services of ethereum nodes, in: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023, pp. 1333–1344.

[62] K. Li, Y. Wang, Y. Tang, Deter: Denial of ethereum txpool services, in: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 1645–1667.

[63] Y. Ma, Y. Sun, Y. Lei, N. Qin, J. Lu, A survey of blockchain technology on security, privacy, and trust in crowdsourcing services, World Wide Web 23 (2020) 393–419.

[64] S. J. Wang, K. Pei, J. Yang, Smartinv: Multimodal learning for smart contract invariant inference, in: 2024 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, 2024, pp. 126–126.

[65] Y. Sun, D. Wu, Y. Xue, H. Liu, H. Wang, Z. Xu, X. Xie, Y. Liu, Gptscan: Detecting logic vulnerabilities in smart contracts by combining gpt with program analysis, Proc. IEEE/ACM ICSE (2024).

[66] I. David, L. Zhou, K. Qin, D. Song, L. Cavallaro, A. Gervais, Do you still need a manual smart contract audit?, arXiv preprint arXiv:2306.12338 (2023).

[67] R. Karanjai, E. Li, L. Xu, W. Shi, Who is smarter? an empirical study of ai-based smart contract creation, in: 2023 5th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), IEEE, 2023, pp. 1–8.

[68] F. Ö. Sönmez, W. J. Knottenbelt, Contractarmor: Attack surface generator for smart contracts, Procedia Computer Science 231 (2024) 8–15.

[69] M. ORTU, G. Ibba, C. Conversano, R. Tonelli, G. Destefanis, Identifying and fixing vulnerable patterns in ethereum smart contracts: A comparative study of fine-tuning and prompt engineering using large language models, Available at SSRN 4530467 (????).

[70] X. Sun, L. Tu, J. Zhang, J. Cai, B. Li, Y. Wang, Assbert: Active and semi-supervised bert for smart contract vulnerability detection, Journal of Information Security and Applications 73 (2023) 103423.

[71] L. Yu, J. Lu, X. Liu, L. Yang, F. Zhang, J. Ma, Pscvfinder: A prompt-tuning based framework for smart contract vulnerability detection, in: 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2023, pp. 556–567.

[72] Y. Sun, D. Wu, Y. Xue, H. Liu, W. Ma, L. Zhang, M. Shi, Y. Liu, Llm4vuln: A unified evaluation framework for decoupling and enhancing llms' vulnerability reasoning, arXiv preprint arXiv:2401.16185 (2024).

[73] Y. Gai, L. Zhou, K. Qin, D. Song, A. Gervais, Blockchain large language models, arXiv preprint arXiv:2304.12749 (2023).

[74] J. Nicholls, A. Kuppa, N.-A. Le-Khac, Enhancing illicit activity detection using xai: A multimodal graph-llm framework, arXiv preprint arXiv:2310.13787 (2023).

[75] C. Shou, J. Liu, D. Lu, K. Sen, Llm4fuzz: Guided fuzzing of smart contracts with large language models, arXiv preprint arXiv:2401.11108 (2024).

[76] L. Zhang, K. Li, K. Sun, D. Wu, Y. Liu, H. Tian, Y. Liu, Acfix: Guiding llms with mined common rbac practices for context-aware repair of access control vulnerabilities in smart contracts, arXiv preprint arXiv:2403.06838 (2024).

[77] A. Storhaug, J. Li, T. Hu, Efficient avoidance of vulnerabilities in auto-completed smart contract code using vulnerability-constrained decoding, in: 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2023, pp. 683–693.

[78] N. O. O. Dade, M. Lartey-Quaye, E. T.-K. Odonkor, P. Ammah, Optimizing large language models to expedite the development of smart contracts, arXiv preprint arXiv:2310.05178 (2023).

[79] Y. Du, X. Tang, Evaluation of chatgpt's smart contract auditing capabilities based on chain of thought, arXiv preprint arXiv:2402.12023 (2024).

[80] E. Chen, R. Huang, J. Liang, D. Chen, P. Hung, Gptutor: an open-source ai pair programming tool alternative to copilot, arXiv preprint arXiv:2310.13896 (2023).

[81] A. Trozze, T. Davies, B. Kleinberg, Large language models in cryptocurrency securities cases: Can chatgpt replace lawyers?, arXiv preprint arXiv:2308.06032 (2023).

[82] H. Axelsen, S. Axelsen, V. Licht, J. Potts, Scaling culture in blockchain gaming: Generative ai and pseudonymous engagement, arXiv preprint arXiv:2312.07693 (2023).

[83] Y. Liu, Q. Lu, L. Zhu, H.-Y. Paik, Decentralised governance for foundation model based ai systems: Exploring the role of blockchain in responsible ai, IEEE Software (2024).

[84] C. Ziegler, M. Miranda, G. Cao, G. Arentoft, D. W. Nam, Classifying proposals of decentralized autonomous organizations using large language models, arXiv preprint arXiv:2401.07059 (2024).

[85] R. Karanjai, L. Xu, W. Shi, Teaching machines to code: Smart contract translation with llms, arXiv preprint arXiv:2403.09740 (2024).

[86] S. Wellington, Basedai: A decentralized p2p network for zero knowledge large language models (zk-llms), arXiv preprint arXiv:2403.01008 (2024).

[87] H. Luo, J. Luo, A. V. Vasilakos, Bc4llm: Trusted artificial intelligence when blockchain meets large language models, arXiv preprint arXiv:2310.06278 (2023).

[88] H. He, T. Wang, H. Yang, J. Fu, N. J. Yuan, J. Yin, H. Chao, Q. Zhang, Learning profitable nft image diffusions via multiple visual-policy guided reinforcement learning, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 6831–6840.

[89] J. K. Kim, M. Chua, M. Rickard, A. Lorenzo, Chatgpt and large language model (llm) chatbots: The current state of acceptability and a proposal for guidelines on utilization in academic medicine, Journal of Pediatric Urology (2023).

[90] J. G. Meyer, R. J. Urbanowicz, P. C. Martin, K. O'Connor, R. Li, P.-C. Peng, T. J. Bright, N. Tatonetti, K. J. Won, G. Gonzalez-Hernandez, et al., Chatgpt and large language models in academia: opportunities and challenges, BioData Mining 16 (2023) 20.

[91] T. Teubner, C. M. Flath, C. Weinhardt, W. van der Aalst, O. Hinz, Welcome to the era of chatgpt et al. the prospects of large language models, Business & Information Systems Engineering 65 (2023) 95–101.

[92] Y. Tan, D. Min, Y. Li, W. Li, N. Hu, Y. Chen, G. Qi, Can chatgpt replace traditional kbqa models? an in-depth analysis of the question answering performance of the gpt llm family, in: International Semantic Web Conference, Springer, 2023, pp. 348–367.

[93] Ö. Aydin, E. Karaarslan, Is chatgpt leading generative ai? what is beyond expectations?, Academic Platform Journal of Engineering and Smart Systems 11 (2023) 118–134.

[94] P. P. Ray, Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope, Internet of Things and Cyber-Physical Systems (2023).

[95] K. I. Roumeliotis, N. D. Tselikas, Chatgpt and open-ai models: A preliminary review, Future Internet 15 (2023) 192.

[96] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, et al., Toolllm: Facilitating large language models to master 16000+ real-world apis, arXiv preprint arXiv:2307.16789 (2023).

[97] Q. Miao, W. Zheng, Y. Lv, M. Huang, W. Ding, F.-Y. Wang, Dao to hanoi via desci: Ai paradigm shifts from alphago to chatgpt, IEEE/CAA Journal of Automatica Sinica 10 (2023) 877–897.