DECENTRALISED GOVERNANCE-DRIVEN ARCHITECTURE FOR DESIGNING FOUNDATION MODEL BASED SYSTEMS: EXPLORING THE ROLE OF BLOCKCHAIN IN RESPONSIBLE AI

arXiv:2308.05962v3 [cs.SE] 21 Feb 2024

Yue Liu[1,2], **Qinghua Lu**[1,2], **Liming Zhu**[1,2], **Hye-Young Paik**[2]
[1]Data61, CSIRO, Australia
[2]University of New South Wales, Australia

February 23, 2024

ABSTRACT

Foundation models including large language models (LLMs) are increasingly attracting interest worldwide for their distinguished capabilities and potential to perform a wide variety of tasks. Nevertheless, people are concerned about whether foundation model based AI systems are properly governed to ensure the trustworthiness and to prevent misuse that could harm humans, society and the environment. In this paper, we identify eight governance challenges of foundation model based AI systems regarding the three fundamental dimensions of governance: decision rights, incentives, and accountability. Furthermore, we explore the potential of blockchain as an architectural solution to address the challenges by providing a distributed ledger to facilitate decentralised governance. We present an architecture that demonstrates how blockchain can be leveraged to realise governance in foundation model based AI systems.

Key terms - Governance, Foundation model, Large language model, LLM, Blockchain, Accountability, Responsible AI

1 Introduction

The year of 2023 has witnessed the emergence of large language models, one type of foundation models. Unlike previous AI models which are trained using data from a particular domain and aim to resolve problems in that domain, foundation models are trained with an extensive range of data for comprehensive capabilities, and eventually to accomplish various tasks [1]. OpenAI released a conversational foundation model named "ChatGPT" based on the GPT-3.5 model in 2022. ChatGPT draws widespread attention from diverse areas that it reached over 100 million users in two months after release [2], resulting in competition that many other IT companies also released their foundation models: Google introduced "Bard", and Meta released "LLaMA", to name a few.

Foundation models can provide a wide variety of services based on the massive AI models and vast amounts of broad training data [1]. For instance, ChatGPT has shown its outperformance in natural language processing, code programming and analysis. Currently there are numerous projects exploring the potential use of foundation model based systems (e.g. agents) in diverse human-AI teaming scenarios, such as climate [3], medicine [4], gaming [5], etc., Figure 1 is a high-level graphical representation of foundation model based system ecosystem. Users can sift out the appropriate foundation model based system(s) considering the performance, cost, etc. to achieve certain goals. When multiple systems are employed, they may cooperate with each other while one of them needs to act as a coordinator. The systems can generate strategies and tasks, which may require subtle tools to orchestrate chores such as interactions and specific problem resolution.

However, we notice that currently there is a lack of consideration of how to realise governance in foundation model based AI systems across different architecture layers. Specifically, foundation model based systems consist of diverse AI and non-AI components, plugins, and stakeholders behind these products. Lacking proper governance solutions may

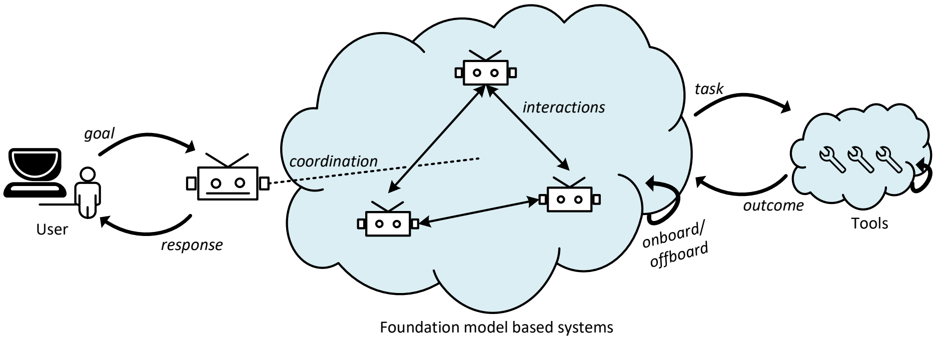


Figure 1: Human-AI teaming in job market.

result in disordered decision rights, complicated accountability processes, and eventually harm to humans, society and the environment. Considering the diversity of involved stakeholders, blockchain can serve as an infrastructure to realise decentralised governance in foundation model based AI systems.

Blockchain was firstly known as the underlying infrastructure of cryptocurrencies [6]. It is then generalised to distributed data storage and computing platforms, where a large network of untrusted participants need to reach agreements on transactional data states [7]. Blockchain technology is proven to preserve certain software attributes and bring its distinctive features like transparency, on-chain autonomy to existing software applications. Blockchain can provide two core elements for realising decentralisation in existing software systems: (i) a distributed ledger, and (ii) a decentralised "compute" infrastructure. A blockchain is essentially a distributed ledger for transaction storage and verification without relying on any central trusted authority [7]. The on-chain programmability via smart contracts enables blockchain as a "compute" infrastructures [8]. Smart contracts can be deployed on-chain to realise complex on-chain business logic such as triggers, conditions, etc.

In this paper, blockchain is leveraged as a means to realise governance in foundation model based systems. We first analyse the challenges in the foundation model systems regarding the three governance dimensions, namely, decision rights, incentives, and accountability. We then discuss how blockchain can address each identified challenge, and present a blockchain-based architecture design for governance-driven foundation model systems, where blockchain is leveraged for identity management, response recording and validation, and incentive distribution.

2 Governance challenges in foundation model based systems

Foundation models are believed to become a revolutionary force in the field of artificial intelligence, whereas the development and use of foundation model based AI systems are still at an early stage. It is in doubt whether these systems can behave in a responsible and trustworthy manner. In this paper, we adhered to our previous work on designing foundation model based systems [9], and extended the scope by adopting traditional IT governance dimensions [10] and exploring European Commission's new Product Liability Directive [11, 12] to investigate the governance issues. As listed in Table 1, there is still a set of governance challenges in foundation model based AI systems.

Challenges of decision rights. Decision rights refer to stakeholders' authority, responsibility and capability for decision-making. The determination of decision rights is complicated as there are diverse roles in foundation model based systems. For instance, a system consists of a foundation model, the orchestration components for handling interactions and communication, external systems and corresponding APIs and plugins for certain tasks, and additional operational components for safeguarding the use of foundation model. All components have their respective providers. A critical governance issue is that what are the responsibilities and capabilities of these project teams in a foundation model based system, and how they control the rights and act in the system to coordinate with users.

Further, for a foundation model based system, users input prompts to the system, and receive the generated responses. Nevertheless, the generated content may cause conflicts as users believe the results are created adhering to their thoughts and instructions, whilst the system providers may also require the responses to fine-tune the foundation models and train new models. Hence, the involved parties of a foundation model based system need to decide the rights of the model-generated content and related intellectual property. A more intricate circumstance is to take the data subjects and their consent for a foundation model to perform tasks into consideration.

In addition, there may be highly-modularised systems that allow users to customise the combination of multiple foundation model based systems and plugin tools, which will derive the need of comparing and selecting different

Table 1: Governance challenges of foundation model based AI systems.

Governance dimensions		Challenges	Blockchain-based solutions
Decision rights	D1.	How to determine the decision rights of stakeholders, and how they can control and act in the system?	Blockchain provides a governance infrastructure where stakeholders' decision rights can be managed via embedded access control mechanisms (e.g., who can access the training dataset for foundation models).
	D2.	How to determine the Intellectual Property (IP) of contents generated by foundation model based systems?	IP agreement template can be deployed as smart contracts, which should be signed by the involved stakeholders (e.g. system providers, foundation model providers, users).
	D3.	How to select foundation models, agents, or external tools for certain tasks?	A marketplace of foundation model based systems or external tools can be developed and deployed as a decentralised application in a blockchain network.
Incentives	I1.	How to motivate the foundation model based systems to behave in a responsible manner?	Blockchain can be used to distribute incentives to reward stakeholders (e.g., verifiers, tool providers) for actions that are aligned with human values, accordingly, incentives will also be locked or destructed if violations are detected.
	I2.	How to compensate stakeholders who are affected by the unintended or harmful behaviours of foundation model based systems?	The impacted stakeholders can be registered in a smart contract. Compensation can be made after confirmation.
Accountability	A1.	How is identity managed in foundation model based AI systems?	Stakeholders can participate in the same blockchain network, where they can register blockchain accounts as on-chain identities for themselves and their products or virtual representatives (e.g., agents).
	A2.	How to scrutinise the operation information of foundation models?	Smart contracts can provide storage for recording users' prompts and foundation model-generated responses, and voting schemes for reaching consensus on the validation results.
	A3.	How to realise responsible resource provenance in foundation model based AI systems?	Blockchain can be leveraged to record critical runtime data of different components, such as inputs/outputs of foundation models, actions taken by the external tools, retrieved data from local data store through RAG, etc. Such information can enable traceability and auditability of the responsibilities of relevant stakeholders.

systems and downstream tools for certain tasks. System developers, procurers, and users need to determine which system to deploy or even whether to include multiple systems to collaborate. A foundation model based system marketplace is required to provide a unified and convenient source to select assorted systems and tools based on particular metrics (e.g., price, processing time, context window).

Challenges of incentives. In a foundation model based system, any misconduct during operation may result in the abnormal performance of foundation model, e.g., mistakes in model fine-tuning can lead to incorrect responses to users' prompts. Incentives are considered a significant factor to motivate and regulate the system and related stakeholders' behaviours. We also envision that the systems themselves may require more rights in future, therefore, users may need to provide actual incentives to accomplish certain tasks.

In addition, product defects may lead to the unintended or harmful behaviours of foundation model systems (e.g., property damage due to inaccurate classification outputs), which could heavily affect end users and even a larger community [11]. In this case, the system or tool providers are liable for the defects and unexpected behaviours, while the victims may demand restitution. Hence, a complete process of compensation is required for foundation model based AI systems. Such situations can be further generalised to dispute resolution in foundation model systems. Different stakeholders may have their own goals, while the authorities in a foundation model based system (e.g., system providers, governors) can resolve conflicts by providing incentives to emphasise the collective benefit of most stakeholders. However, a subsequent concern is whether pursuing incentives introduces bias in prompt execution and conflict resolution, and corresponding mechanisms should be in place to redress issues or harms.

Challenges of accountability. Accountability refers to the identifiability, answerability and traceability processes of stakeholders for their behaviours and activities in corresponding decision-making processes. A fundamental challenge of accountability would be how to manage identities in foundation model based AI systems. A foundation model system may be developed and operated across multiple organisations, and a unified identity management solution, albeit each organisation may have its own management system, can help facilitate the coordination across different organisations. Further, with the increasingly enhanced capability of foundation models, the application systems may be assigned formal identities in future to better address accountability issues.

A foundation model is developed via a large amount of corpus for training and specific datasets for fine-tuning, and the procurers will adapt the foundation model and implement the entire AI system. Nevertheless, ineffective training and fine-tuning, tooling defects, or misleading instructions (e.g., prompt injection) may introduce discrimination to the system, and result in biased responses or hallucinations to users. Consequently, responses need to be examined to understand and assess the operation status and mitigate risks. Verifiers are incorporated to validate model-generated results for iterative fine-tuning, while their activities also need proper supervision to ensure security and safety.

The process of acquiring necessary data and tooling resources (e.g. external software systems) for accomplishing certain tasks can be sophisticated as it involves problems such as data privacy and security, obtaining access to specific tools or APIs, licensing agreements, etc. Maintaining proper recordings for responsible provenance process is significant for realising accountability of the eventual responses to users.

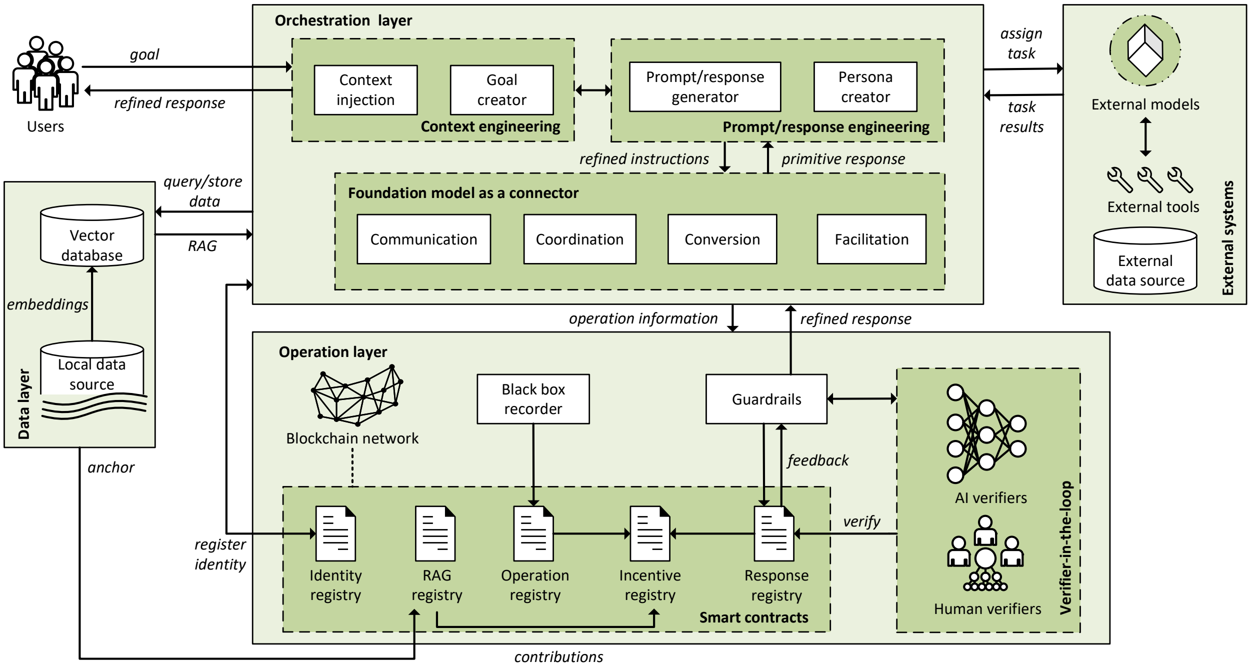


Figure 2: Blockchain-based architecture for governance-driven foundation model applications.

3 Blockchain as a Governance Solution for Foundation Model Systems.

Blockchain can be leveraged to address governance issues by providing a transparent distributed ledger maintained in each participating entity for accountability, and a programmable infrastructure to facilitate and automate incentive distribution and decision-making processes. The third column of Table 1 specifies how blockchain can be utilised to realise governance in foundation model based systems regarding the above identified challenges.

First, the system providers can deploy a permissioned blockchain network where all relevant stakeholders can collaborate to finalise different decisions. Specifically, smart contracts enable built-in access control to manage stakeholders' decision rights to certain governance issues (***D1***). For instance, only appointed verifiers are capable of viewing and validating model-generated responses, or only system providers and auditors can access the training dataset. All participating stakeholders will have their public keys as on-chain identities and private keys for authorisation. Note that the whole blockchain network can be considered a mapping to off-chain business relationships, hence participating in the network may require real-world identity verification, which can ensure more complete accountability attribution

(***A1***). Further, blockchain-based self-sovereign identity can help establish formal decentralised identities for different users, foundation models, systems and tools in business relationships.

When operating a foundation model based system, the system providers can explain the intellectual property rights of model-generated data, and obtain users' consent to the clarified terms and policies, which can be demonstrated in the form of smart contracts. The system providers, users and all involved stakeholders need to reach agreement and sign the smart contract before users can access the system services (***D2***). In addition, assorted systems can provide similar services, whilst a highly-modularised system may consist of multiple foundation models and tools, and users can choose one or several systems/models/tools for certain tasks. In this case, a marketplace for selecting foundation model systems can be developed as a decentralised application on blockchain, to provide a unified infrastructure for users to intuitively compare the systems regarding different metrics (***D3***).

Blockchain allows issuing on-chain tokens[1] as inherent incentives to motivate stakeholders' behaviours. For instance, verifiers are rewarded for their contributions of improving foundation model's response. The system providers can either issue two types of tokens to represent the positive and negative incentives respectively, or manage only one token type for both rewards and penalties (e.g., locking or destructing tokens) via clear and strict regulations. Stakeholders' possessed tokens are checked periodically and transferred into real-world currencies based on the business agreement. Further, incentives can be given to the systems or foundation models as they also have on-chain identities, to encourage or depress certain activities (***I1***). However, a concern would be that foundation models are considered "black box" to stakeholders, it would be difficult to anticipate how they would perform to gain incentives, which may require enhanced explainability to improve trustworthiness.

A broad community can benefit from the services provided by the systems, or may be affected by their unintended or harmful behaviours. In the latter case, blockchain can serve as a log for identifying the responsible entities, and hence alleviating users' burden of proof. The victims of foundation model systems may request compensation for their loss. Smart contracts provide distributed registries where all impacted entities can sign up, while the system providers and other responsible stakeholders can make compensation according to these registries after accountability processes (***I2***).

In particular, the foundation model system operation status can be revealed by the generated responses, which can be recorded in smart contracts for validation. Nevertheless, verifiers may have opposite understandings and thoughts about the responses. Such conflicts can be resolved via efficient ways such as the authorities (e.g., system providers) can make decisions in a short time, or via more democratic means like referendum. In the latter case, smart contracts support voting for validation determination, where various voting schemes can be applied to highlight different quality attributes (e.g., security, flexibility, preference expression) (***A2***). System providers can also participate in voting as their suggestions are significant to verifiers, and their vote(s) can be set with a different weight.

Meanwhile, all activities related to response validation are recorded by blockchain for further analysis. In addition, for each task, the critical runtime data and resource information (e.g., foundation model input/output, external tools, retrieved data from local data store through retrieval augmented generation (RAG)) can all be kept in smart contracts for auditing (***A3***). All on-chain data cannot be tampered with or discarded by malicious stakeholders without being perceived by others. Consequently, blockchain can ensure data integrity, hence resource provenance and accountability in foundation model based AI systems by providing evidence for audit trail.

4 Blockchain-based architecture for governance-driven foundation model based systems.

In this section, we propose an architecture design to present how to leverage blockchain as a software component to address certain governance issues in foundation model based AI systems. We adopt and extend existing architectures for foundation model based AI systems [9, 13–15] by integrating a blockchain network and five on-chain smart contracts. Figure 2 illustrates an overview of a blockchain-based governance-driven architecture design for foundation model based systems. Specifically, the architecture consists of three main layers: orchestration layer, data layer, and operation layer, while we also include external application systems to demonstrate the complete workflow.

Orchestration layer. The orchestration layer maintains the core services of the whole foundation model system. It is responsible for receiving, executing, and replying to users' instructions, and connecting with other layers for task completion. In particular, the system provides *context engineering* for handling users' inputs to understand the ultimate goals, which are then processed by *prompt/response engineering* components, e.g., inspect whether there are prompt injections, modify or refuse the instructions, and refine the responses. Valid prompts are transferred to the foundation model, which can be leveraged as a *software connector* [13] in the orchestration layer as follows:

[1] Programmable digital assets, different from the concept of "token" as input contents in foundation models.

- *Communication*: Foundation model transfers data between software components, e.g., sending certain data to external applications.
- *Coordination*: Foundation model coordinates the computation results, e.g., decomposing an assignment into fine-grained tasks and generating execution plans.
- *Conversion*: Foundation model transforms data format to assist communication between components, e.g., interpreting users' descriptions to other AI models in a machine-readable scheme.
- *Facilitation*: Foundation model optimises the overall workflow and interactions between components by finalising specific decisions, e.g., whether to invoke other components.

Meanwhile, completing tasks may require external systems, which include other AI models and tools for specific computation processes, and additional data sources for useful information. The task/search results are sent back to the orchestration layer for further processing by foundation model, and storage in the data layer for future queries.

Data layer. The data layer includes two main components: *local data source* (e.g., data lake) and *vector database*. The local data source is responsible for storing raw data, which then will be converted into embeddings (i.e., numerical representations of data) and recorded in the vector database. Specifically, the embeddings can represent the semantic meaning of data, enabling efficient similarity search. As there is usually a context window in foundation models, applying a vector database can help achieve better accuracy for foundation models to understand users' input and generate responses through retrieval augmented generation.

Operation layer. The operation layer consists of a set of on-chain smart contracts and the related off-chain components: *black box recorder*, *guardrails*, *verifier-in-the-loop*, a *blockchain network* and on-chain smart contracts: *identity registry, RAG registry, operation registry, response registry*, and *incentive registry*.

First, blockchain provides a decentralised public key infrastructure that can assign blockchain accounts to all stakeholders. Moreover, the *identity registry* can enable formal on-chain identity management (e.g., self-sovereign identity) to establish and maintain certain business relationships.

Secondly, considering off-chain data repositories may be compromised and tampered with, which will result in inaccurate model responses, the *RAG registry* can be utilised for periodically anchoring off-chain data from the data layer. Similarly, *black box recorder* saves the runtime data in *operation registry*, including the input/output, and intermediate data of other layers and components (e.g., external tools). In addition, user consent can also be included in the *operation registry* as it is regarded as the starting point for a task. Storing all the above information on-chain comprises a complete and traceable workflow to provide audit trails if the system has unexpected behaviours. Any attempt to revise on-chain stored data will leave traces while modifying the related block requires altering all subsequent blocks, hence data integrity is preserved via smart contracts.

Thirdly, *response registry* facilitates the *guardrail* functionalities. Guardrails can verify whether users' prompts are compliant with responsible AI requirements and also refine foundation model's responses. Any response violating system provider's predefined rules will be recorded in *response registry*. Subsequently, verifiers can assess the quality (e.g., correctness, relevance, and appropriateness) of model outputs and guardrails operation via this registry and provide feedback (e.g., voting whether the output is appropriate and useful), which is then used to improve the guardrails and fine-tune the foundation model to better process users' instructions.

Finally, the *incentive registry* records the contributions of different stakeholders. In particular, *RAG registry* and *operation registry* transfer data contributors and providers of the employed tools respectively, while *response registry* delivers the involved verifiers to *incentive registry*. System providers can either issue and distribute on-chain tokens via this registry, or examine the contribution records and provide rewards to the relevant entities through off-chain channels.

Discussion. In the proposed architecture, blockchain is exploited to achieve governance in foundation model based systems. First, blockchain assists in decision-making on response validation via various voting schemes, which can also be applied to other conflict resolutions in the system. Moreover, the providers of foundation model and external applications can register identities for their products to enable accountability and incentive distribution processes. Responsibility attribution is facilitated when the foundation model generates incorrect or even malicious responses to users, since blockchain records the used data, runtime information, and prompt execution outcome, along with the registered on-chain identities of involved stakeholders. In addition, incentives are allocated according to the contributions regarding data, tooling, task completion and validation.

We remark that the proposed architecture can serve as a reference while the practitioners need to finalise the design decisions further. For instance, a consortium blockchain network can be deployed with resource/energy-efficient solutions. The prompts and responses can be encrypted or apply certain access control mechanisms to be examined by only authorised verifiers. The votes can be held with different schemes, such as one verifier one vote, one token one vote, etc. It should also be noted that the incentive distribution to foundation models is dependent on model explainability, whilst the marketplace for foundation model systems also requires clear taxonomy based on assorted metrics.

5 Conclusion and Future Work

In this paper, we identify a series of governance challenges in foundation model based AI systems in terms of decision rights, incentives, and accountability, and discuss the role of blockchain in addressing each identified challenge. In addition, we propose a blockchain-based architecture for designing governance-driven foundation model systems. The architecture focuses on the governance issues during the operation of foundation model systems, including identity management, response validation, incentive provision, and accountability of system operation. The proposed architecture leverages five on-chain smart contracts to realise governance in the foundation model based system. In our future work, we plan to implement a proof-of-concept prototype with fine-grained design decisions, evaluate its performance, and further explore decentralised governance in foundation model based systems.

References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[2] T. Teubner, C. M. Flath, C. Weinhardt, W. van der Aalst, and O. Hinz, "Welcome to the era of chatgpt et al. the prospects of large language models," *Business & Information Systems Engineering*, pp. 1–7, 2023.

[3] M. Leippold, "Thus spoke gpt-3: Interviewing a large-language model on climate finance," *Finance Research Letters*, vol. 53, p. 103617, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1544612322007930

[4] T. H. Kung, M. Cheatham, A. Medenilla, C. Sillos, L. De Leon, C. Elepaño, M. Madriaga, R. Aggabao, G. Diaz-Candido, J. Maningo *et al.*, "Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models," *PLoS digital health*, vol. 2, no. 2, p. e0000198, 2023.

[5] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3586183.3606763

[6] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," https://bitcoin.org/bitcoin.pdf, 2008, accessed 26-May-2022.

[7] F. Tschorsch and B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, p. 464, 2016.

[8] S. Omohundro, "Cryptocurrencies, Smart Contracts, and Artificial Intelligence," *AI Matters*, vol. 1, no. 2, pp. 19–21, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2685328.2685334

[9] Q. Lu, L. Zhu, X. Xu, Z. Xing, S. Harrer, and J. Whittle, "Towards responsible generative ai: A reference architecture for designing foundation model based agents," *arXiv preprint arXiv:2311.13148*, 2023.

[10] P. Weill, "Don't just lead, govern: How top-performing firms govern it," *MIS Quarterly Executive*, vol. 3, pp. 1–17, 03 2004.

[11] S. D. Luca, "New product liability directive," https://www.europarl.europa.eu/RegData/etudes/BRIE/2023/739341/EPRS_BRI(2023)739341_EN.pdf, 2023, accessed 21-February-2024.

[12] T. Rodríguez de las Heras Ballell, "The revision of the product liability directive: a key piece in the artificial intelligence liability puzzle," in *ERA Forum*. Springer, 2023, pp. 1–13.

[13] Q. Lu, L. Zhu, X. Xu, Y. Liu, Z. Xing, and J. Whittle, "A taxonomy of foundation model based systems through the lens of software architecture," *arXiv preprint arXiv:2305.05352*, 2023.

[14] Q. Lu, L. Zhu, X. Xu, Z. Xing, and J. Whittle, "Towards responsible ai in the era of chatgpt: A reference architecture for designing foundation model-based ai systems," *arXiv preprint arXiv:2304.11090*, 2023.

[15] M. Bornstein and R. Radovanovic, "Emerging Architectures for LLM Applications," https://a16z.com/2023/06/20/emerging-architectures-for-llm-applications/?utm_source=tldrnewsletter, 2023, accessed 21-February-2024.

arXiv:2403.14280v4 [cs.CR] 11 May 2024

Large Language Models for Blockchain Security: A Systematic Literature Review

Zheyuan He[a], Zihao Li[b], Sen Yang[a], Ao Qiao[a], Xiaosong Zhang[a], Xiapu Luo[b], Ting Chen[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 across various domains within cyber security. Notably, recent studies are increasingly exploring LLMs applied to the context of blockchain security (BS). However, there remains a gap in a comprehensive understanding regarding the full scope of applications, impacts, and potential constraints of LLMs on blockchain security. To fill this gap, we undertake a literature review focusing on the studies that apply LLMs in blockchain security (LLM4BS).

Our study aims to comprehensively analyze and understand existing research, and elucidate how LLMs contribute to enhancing the security of blockchain systems. Through a thorough examination of existing literature, 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, transaction anomaly detection, vulnerability repair, program analysis of smart contracts, and serving as participants in the cryptocurrency community. Furthermore, we assess the challenges and limitations associated with leveraging LLMs for enhancing blockchain security, considering factors such as scalability, privacy concerns, and ethical concerns. Our thorough review sheds light on the opportunities and potential risks of tasks on LLM4BS, providing valuable insights for researchers, practitioners, and policymakers alike.

Keywords:
Blockchain Security, Large Language Model, Literature Review

Preliminary manuscript

1. Introduction

As the digital era advances, 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, 5, 6] intersect with the ever-evolving domain of blockchain security [7, 8, 5, 6, 9, 10]. LLMs have risen to the forefront of blockchain security [11, 12], showcasing profound capabilities in text generation and comprehension [13, 14, 6, 15]), especially in source code analysis. These abilities mirror human-like proficiency [16, 17]. This transformative impact is attributable to their expansive datasets, sophisticated architectures, and the deep neural networks that underpin their operational frameworks [1, 18, 17].

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 [19, 20, 3, 4, 21, 22, 23, 24, 25]. Concretely, the granular analysis of smart contracts [13], the meticulous scrutiny of transactions [26], and automatic code (resp. text) generation [27] are among the critical tasks that LLMs are adept at performing with remarkable efficacy [14, 28, 9].

However, integrating these cognitive powerhouses into blockchain security is met with an array of challenges that beckon for consideration. Navigating the intricate dynamics of ever-advancing cybersecurity threats and addressing the ethical concerns that accompany AI deployment make this trajectory as demanding as it is promising. Despite the progress, there is still a lack of comprehensive work depicting the current application status and future development prospects of LLM in blockchain security (BS).

To fill the gap, we seek to delve into the multifaceted role of LLMs within the realm of blockchain security, exploring the comprehensive spectrum of LLMs on blockchain security (LLM4BS) tasks. We commence by delineating the contemporary landscape of Large Language Model (LLM) applications across diverse domains (§2.1), as well as the myriad of security threats implicated by the blockchain technology (§2.2). Then, as illustrated in Table.1, we elaborate on the incorporation and progression of LLM4BS tasks, involving smart contract auditing, block transaction detection, contract dynamic analysis, smart contract development, and cryptocurrency community contributors (§3). Thereafter, we meticulously select three quintessential cases of LLM4BS tasks to elucidate the state-of-the-art LLM4BS tasks (§4), comprising LLM4FUZZ [44], SMARTINV [29], BLOCKGPT [26]. Finally, we

Table 1: Table of LLM4BS studies

Domains	Amounts	Publications
Smart Contract Auditing	15	SMARTINV [29], GPTScan [13], David et al. [30], Karanjai et al. [31], ContractArmor [32], Ortu et al. [33], ASSBert [34], PSCVFinder [35], LLM4Vuln [36], TrustLLMf [37], AuditGPT [38], PropertyGPT [39] Chen et al. [40], Jain et al. [41] and SolGPT [42]
Block Transaction Detection	2	BLOCKGPT [26] and Nicholls et al. [43]
Contract Dynamic Analysis	2	LLM4FUZZ [44] and ACFIX [45].
Smart Contract Development	8	Storhaug et al. [27], karanjai et al. [31], MazzumaGPT [46], Du et al. [47], GPTutor [48], Petrovic et al. [49], Zhao et al. [50] and Haque et al. [51]
Cryptocurrency Community Contributors	5	Trozze et al. [52], Axelsen et al. [53], Liu et al. [54], ziegler et al. [55] and GPTutor [56]
Miscellaneous	5	compilers [57], zero-knowledge proofs [58], model training [20, 59] and NFT generation [60]

present an insightful discourse on the challenges presently faced within the ambit of LLM4BS, and proffer prospective trajectories for future research and development in this emergent field (§5).

This paper makes the following contributions:

- To the best of our knowledge, after a meticulous review of the existing literature, we conduct the first systematic examination focusing on the application of Large Language Models to tasks within the realm of blockchain security, offering a pioneering exploration of the interplay between advanced AI and distributed ledger systems.

- In our comprehensive survey, we meticulously chronicle the current landscape of applications of Large Language Models in the domain of blockchain security. We delve into a detailed analysis of how Large Language Models are employed across various scenarios, from enhancing the reliability of smart contracts to fortifying the integrity of distributed ledger systems. This sheds light on the multifaceted contributions of this cutting-edge technology.

- Based on our study, we rigorously compile and summarize a range of practical academic achievements related to the application of Large Language Models (LLMs) in strengthening blockchain security. We also propose several promising avenues for future research, anticipating that these will catalyze substantial advancements and innovations within this burgeoning intersection of fields.

2. Overview of LLM4BS

We provide basic knowledge about LLM4BS tasks in this section, including LLM applications in §2.1 and threats of blockchain security in §2.2.

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) [61]. 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 [62]. At the heart of LLMs lies the transformer architecture, a powerful framework for sequence modeling that has revolutionized the field of NLP [63].

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 [64]. 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 [65]. 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 the context within language, enabling them to generate responses or predictions that are sensitive to the surrounding textual context [66]. This contextual

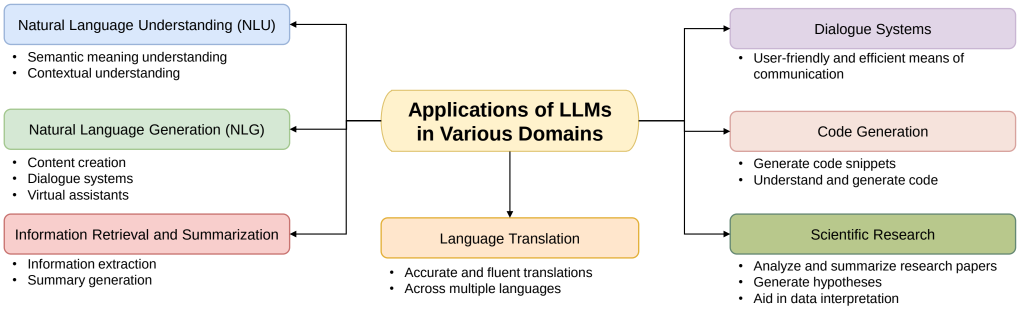

Natural Language Understanding (NLU)
• Semantic meaning understanding
• Contextual understanding
Natural Language Generation (NLG)
• Content creation
• Dialogue systems
• Virtual assistants
Information Retrieval and Summarization
• Information extraction
• Summary generation
Applications of LLMs in Various Domains
Language Translation
• Accurate and fluent translations
• Across multiple languages
Dialogue Systems
• User-friendly and efficient means of communication
Code Generation
• Generate code snippets
• Understand and generate code
Scientific Research
• Analyze and summarize research papers
• Generate hypotheses
• Aid in data interpretation

Figure 1: The various applications of LLM.

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 [67].

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 [68].

2.1.2. Applications of LLMs in Various Domains

As depicted in Fig. 1, 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 [69]. 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 [70]. 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 [71]. 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 [72]. 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 [73]. 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 [74]. 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 [71, 75]. 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 [76]

Scientific Research: LLMs support scientific discovery by analyzing and summarizing research papers, generating hypotheses, and aiding in data interpretation [77, 69]. 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 [78].

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 [79]. 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 [80, 81]. Techniques such as hashing, digital signatures, and cryptographic keys are utilized to secure data and verify the authenticity of transactions on the blockchain [82].

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 [83, 84]. Popular consensus mechanisms include Proof of Work (PoW) [85], Proof of Stake (PoS) [86], and Delegated Proof of Stake (DPoS) [87], 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 [88, 89]. By eliminating single points of failure and reducing the risk of censorship or manipulation, decentralization enhances the resilience and security of the blockchain network [90]. However, achieving true decentralization requires careful consideration of factors such as node distribution, governance structures, and network incentives [91].

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 [92, 93, 94]. 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 [95, 96].

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

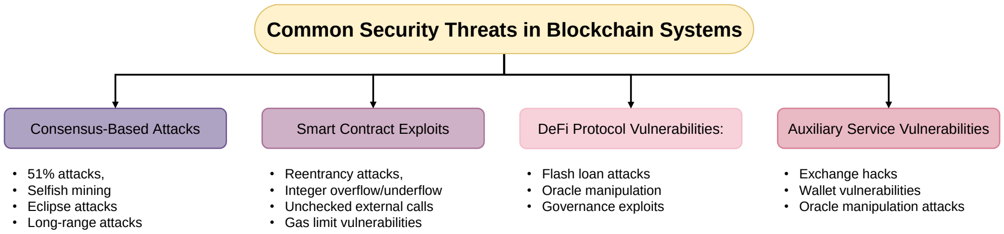


Figure 2: The threats in blockchain systems.

functionality of blockchain systems [97, 98]. We illustrate those threats in Fig. 2. 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 [99]. 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 [100]. 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 [101].

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 [102, 103]. 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 [104, 105, 106]. 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 [107, 108].

DeFi Protocol Vulnerabilities: Decentralized finance (DeFi) protocols introduce new security challenges due to their complex interactions and composability [109, 95, 110]. 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 [111, 112, 113]. 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 [114, 115].

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 [116, 117]. 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 [118, 119]. 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 losses.

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 [120]. 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 tasks

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. 3 depicts the five applications of LLM4BS task, involving code audit of smart contracts §3.1, analysis of abnormal transactions §3.2, dynamic analysis of smart contracts §3.3, development of smart contracts §3.4, participants of cryptocurrency community §3.5, and other potential directions §3.6.

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, such as SMARTINV [29], GPTScan [13], David et al. [30], Karanjai et al. [31], ContractArmor [32], Ortu et al. [33], ASSBert [34], PSCVFinder [35], LLM4Vuln [36], TrustLLMf [37], AuditGPT [38], Chen et al. [40], Jain et al. [41], PropertyGPT [39] and SolGPT [42]. As shown

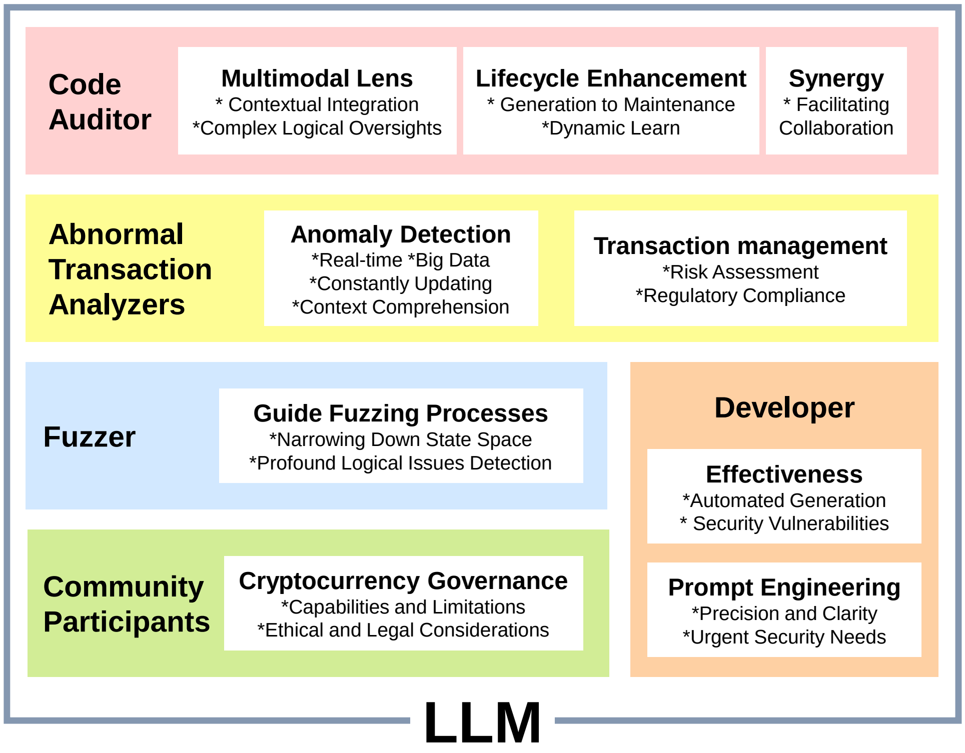


Figure 3: The applications of LLM on the task of blockchain security.

in Table.2, these tools powered by Large Language Models 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.

The integration 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 but also fortifies smart contracts against the myriad of risks that could lead

Table 2: Table of tools and models for code auditor

Research	Functionality	LLM(s)	Publication
SMARTINV [29]	Model finetuning approach	Alpaca etc.	SP,2024
GPTScan [13]	Logic vulnerability detection	GPT-3.5 etc.	ICSE,2024
David et al. [30]	LLM audit feasibility	GPT-4 etc.	arXiv,2023
Karanjai et al. [31]	LLM code evaluation	ChatGPT etc.	BRAINS,2023
ContractArmor [32]	Rule-based code analysis	ChatGPT	EUSPN,2024
Ortu et al. [33]	Contract automatic repair	ChatGPT etc.	arXiv,2023
Jain et al. [41]	Contract automatic repair	GPT-3.5 etc.	ICI,2023
ASSBert [34]	Contract vulnerability detection	BERT	JISA,2023
PSCVFinder [35]	Contract vulnerability detection	CodeT5	ISSRE,2023
Chen et al. [40]	Contract vulnerability detection	ChatGPT etc.	arXiv,2023
LLM4Vuln [36]	Vulnerability reasoning enhancement	GPT-4 etc.	arXiv,2024
TrustLLMf [37]	Smart contracts audit	CodeLlama-13b	arXiv,2024
AuditGPT [38]	ERC token audit	ChatGPT etc.	arXiv,2024
PropertyGPT [39]	Formal verification automation	GPT-4	arXiv,2024
SolGPT [42]	Contract vulnerability detection	GPT-2	ICA3PP,2023

to substantial financial repercussions.

In essence, the integration of Large Language Models in smart contract analysis marks a significant leap in safeguarding the infrastructural integrity of blockchain technology. 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 [121]. 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

Table 3: Table of tools and models for abnormal transaction analyzers

work	function	LLM(s)	Publication
BLOCKGPT [26]	Transaction anomaly detection	Transformer	arXiv,2023
Nicholls et al. [43]	Logic vulnerability detection	BERT etc.	arXiv,2023

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, such as BLOCKGPT [26] and Nicholls et al. [43], underscores their crucial role in conducting real-time monitoring to detect signs of irregular or suspicious behavior. These tools in Table 3 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

Table 4: Table of tools and models for smart contract fuzzer

work	function	LLM(s)	Publication
LLM4FUZZ [44]	Fuzzing optimization tool	Llama 2	arXiv,2024
ACFIX [45]	AC vulnerability repair	GPT-4	arXiv,2024

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, such as LLM4FUZZ [44] and ACFIX [45]. This methodology in Table 4 involves utilizing LLMs to accurately assess the complexity

Table 5: Table of tools and models for smart contract developer

work	**function**	**LLM(s)**	**Publication**
Storhaug et al. [27]	Vulnerability-constrained decoding	GPT-J-6B	ISSRE,2023
karanjai et al. [31]	Smart contract generation	ChatGPT etc.	BRAINS,2023
MazzumaGPT [46]	Smart contract generation	Davinci	arXiv,2023
Du et al. [47]	Audit capacity evaluation	GPT-4	arXiv,2024
GPTutor [48]	AI programming assistant	GPT-3.5 etc.	arXiv,2023
Petrovic et al. [49]	Smart contract generation	ChatGPT	ICSEng,2023
Zhao et al. [50]	AI programming assistant	GPT-3.5	arXiv,2024
Haque et al. [51]	Norm extraction	ChatGPT	arXiv,2024

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 toward 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 [122].

Moreover, this refined fuzzing technique allows for the integration of user-defined invariants and manually inserted assertions to monitor and manage the state during fuzzing. This approach can reduce the exploration overhead and improve 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 in Table 5, such as Storhaug et al. [27], karanjai et al. [31], Petrovic et al. [49], Zhao et al. [50], Haque et al. [51], MazzumaGPT [46], Du et al. [47] and GPTutor [48], 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, i.e., 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, such as Trozze

Table 6: Table of tools and models for Cryptocurrency community participants

work	function	LLM(s)	Publication
Trozze et al. [52]	Legal support tools	GPT-3.5 etc.	arXiv,2023
Axelsen et al. [53]	Community moderation support	ChatGPT	arXiv,2023
Liu et al. [54]	Blockchain-based Governance Framework	-	IEEE Software,2024
ziegler et al. [55]	Automating Contextual Classification	GPT-4	arXiv,2024
GPTutor [56]	Blockchain Revolutionizes Finance	DistilBert etc.	IEEE Access,2024

et al. [52], Axelsen et al. [53], Liu et al. [54], Ziegler et al. [55] and GPTutor [56], albeit with their respective strengths and weaknesses. Related works in Table 6 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, the 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 before they can be fully trusted in autonomous roles.

Table 7: Table of other potential work

work	**function**	**LLM(s)**	**Publication**
SolMover [57]	Language Translation Framework	Alpaca	arXiv,2024
BasedAI [58]	Privacy-Preserving Computation	GPT-4 etc.	arXiv,2024
BC4LLM [20]	Secure Learning Path	ChatGPT	arXiv,2023
DLLM [59]	Dynamic Language Modeling	-	arXiv,2023
Diffusion-MVP [60]	NFT creation platform	Stable-Diffusion	MM,2023

3.6. Miscellaneous

As displayed in Table 7, LLM is also used in other blockchain security fields, involving smart contract compilers [57], zero-knowledge proofs [58], model training [20, 59], NFT generation [60]. We will introduce their applications in detail in the future.

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 in Table.8 have been meticulously selected to encompass a broad range of scenarios, comprising LLM4FUZZ [44] §4.1, SMARTINV [29] §4.2, BLOCKGPT [26] §4.3.

4.1. LLM4Fuzz

As depicted in Fig.4, LLM4FUZZ [44] 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.

Table 8: The table of the three cases on LLM4BS

Research	Domain	Publications	Date	faculty
LLM4FUZZ [44]	Fuzz	arXiv	2024	UC Berkeley
SMARTINV [29]	Program analysis	IEEE S&P	2024	Columbia University
BLOCKGPT [26]	Transaction analysis	arXiv	2023	University of California

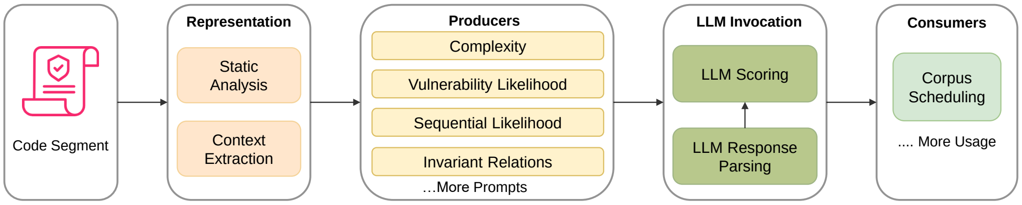


Figure 4: The architecture of LLM4FUZZ.

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

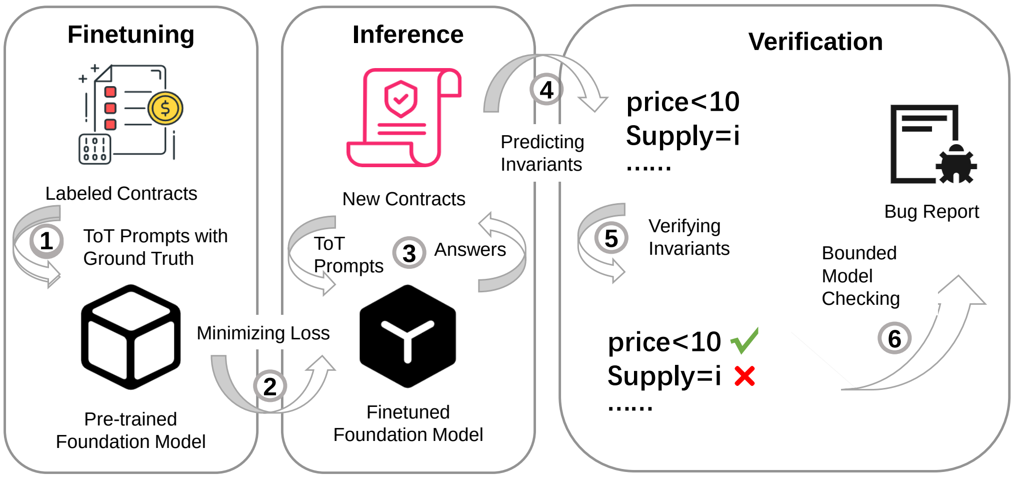


Figure 5: The architecture of SMARTINV.

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 [29] 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. Fig.5 displays the architecture of SMARTINV.

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

As shown in Fig.6, BLOCKGPT [26] 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 IDS 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, including 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

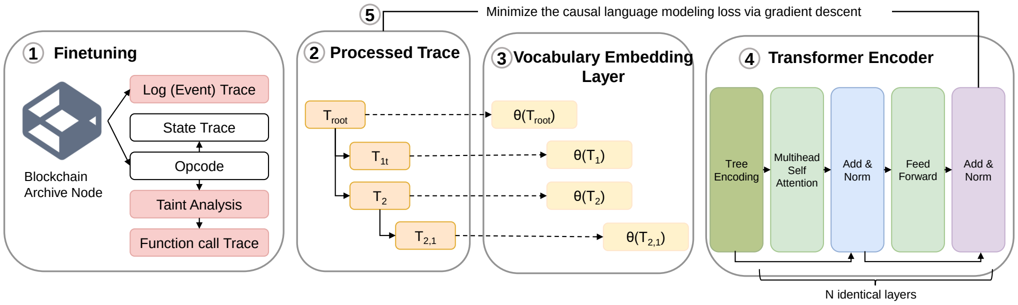


Figure 6: The architecture of BLOCKGPT.

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 tasks

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 [123, 124, 12]. 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 [5, 125]. This relationship is reciprocal; as regulatory agencies develop a deeper understanding 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 [17, 126]. 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 [127, 128]. 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 [129, 130, 131]. 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 toward 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 conclusion, our review of the integration of Large Language Models (LLMs) into blockchain security highlights the technological advancements and intricate challenges presented by this combination of LLM4BS. The potential of LLMs to enhance security protocols in the blockchain is evident, offering innovative solutions for smart contracts, abnormal transaction detection, and cryptocurrency community development. However, realizing this potential requires vigilance regarding scalability, privacy, evolving cyber threats, and the ethical implications of AI. The success of LLMs in blockchain security hinges not only on continuous technological refinement but also on ethical practices, regulatory alignment, and informed community engagement. The integration of LLMs into blockchain security marks a transformative era that necessitates a collaborative approach, balancing innovation with prudent oversight to forge a resilient and equitable security 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] 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.

[6] 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).

[7] 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).

[8] 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).

[9] L. Zhang, Machine learning for blockchain: Literature review and open research questions, in: NeurIPS 2023 AI for Science Workshop, 2023.

[10] H. Wang, J. Zheng, I. E. Carvajal-Roca, L. Chen, M. Bai, Financial fraud detection based on deep learning: Towards large-scale pretraining transformer models, in: China Conference on Knowledge Graph and Semantic Computing, Springer, 2023, pp. 163–177.

[11] R. Zhang, R. Xue, L. Liu, Security and privacy on blockchain, ACM Computing Surveys (CSUR) 52 (2019) 1–34.

[12] F. N. Motlagh, M. Hajizadeh, M. Majd, P. Najafi, F. Cheng, C. Meinel, Large language models in cybersecurity: State-of-the-art, arXiv preprint arXiv:2402.00891 (2024).

[13] 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).

[14] 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.

[15] M. Jin, S. Shahriar, M. Tufano, X. Shi, S. Lu, N. Sundaresan, A. Svyatkovskiy, Inferfix: End-to-end program repair with llms, in: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2023, pp. 1646–1656.

[16] J. Ji, M. Liu, J. Dai, X. Pan, C. Zhang, C. Bian, B. Chen, R. Sun, Y. Wang, Y. Yang, Beavertails: Towards improved safety alignment of llm via a human-preference dataset, Advances in Neural Information Processing Systems 36 (2024).

[17] 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.

[18] 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).

[19] 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.

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

[21] P. Azad, C. G. Akcora, A. Khan, Machine learning for blockchain data analysis: Progress and opportunities, arXiv preprint arXiv:2404.18251 (2024).

[22] H. Xu, S. Wang, N. Li, Y. Zhao, K. Chen, K. Wang, Y. Liu, T. Yu, H. Wang, Large language models for cyber security: A systematic literature review, arXiv preprint arXiv:2405.04760 (2024).

[23] C. T. Nguyen, Y. Liu, H. Du, D. T. Hoang, D. Niyato, D. N. Nguyen, S. Mao, Generative ai-enabled blockchain networks: Fundamentals, applications, and case study, arXiv preprint arXiv:2401.15625 (2024).

[24] J. G. M. Mboma, O. T. Tshipata, W. V. Kambale, K. Kyamakya, Assessing how large language models can be integrated with or used for blockchain technology: Overview and illustrative case study, in: 2023 27th International Conference on Circuits, Systems, Communications and Computers (CSCC), IEEE, 2023, pp. 59–70.

[25] J. G. M. Mboma, K. Lusala, M. Matalatala, O. T. Tshipata, P. S. Nzakuna, D. T. Kazumba, Integrating llm with blockchain and ipfs to enhance academic diploma integrity, in: 2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), IEEE, 2024, pp. 1–6.

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

[27] 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.

[28] 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.

[29] 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.

[30] 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).

[31] 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.

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

[33] 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 (2024).

[34] 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.

[35] 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.

[36] 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).

[37] W. Ma, D. Wu, Y. Sun, T. Wang, S. Liu, J. Zhang, Y. Xue, Y. Liu, Combining fine-tuning and llm-based agents for intuitive smart contract auditing with justifications, arXiv preprint arXiv:2403.16073 (2024).

[38] S. Xia, S. Shao, M. He, T. Yu, L. Song, Y. Zhang, Auditgpt: Auditing smart contracts with chatgpt, arXiv preprint arXiv:2404.04306 (2024).

[39] Y. Liu, Y. Xue, D. Wu, Y. Sun, Y. Li, M. Shi, Y. Liu, Propertygpt: Llm-driven formal verification of smart contracts through retrieval-augmented property generation, arXiv preprint arXiv:2405.02580 (2024).

[40] C. Chen, J. Su, J. Chen, Y. Wang, T. Bi, Y. Wang, X. Lin, T. Chen, Z. Zheng, When chatgpt meets smart contract vulnerability detection: How far are we?, arXiv preprint arXiv:2309.05520 (2023).

[41] A. Jain, E. Masud, M. Han, R. Dhillon, S. Rao, A. Joshi, S. Cheema, S. Kumar, Two timin': Repairing smart contracts with a two-layered approach, in: 2023 Second International Conference on Informatics (ICI), IEEE, 2023, pp. 1–6.

[42] S. Zeng, H. Zhang, J. Wang, K. Shi, Solgpt: A gpt-based static vulnerability detection model for enhancing smart contract security, in: International Conference on Algorithms and Architectures for Parallel Processing, Springer, 2023, pp. 42–62.

[43] 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).

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

[45] 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).

[46] 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).

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

[48] 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).

[49] N. Petrović, I. Al-Azzoni, Model-driven smart contract generation leveraging chatgpt, in: International Conference On Systems Engineering, Springer, 2023, pp. 387–396.

[50] J. Zhao, X. Chen, G. Yang, Y. Shen, Automatic smart contract comment generation via large language models and in-context learning, arXiv preprint arXiv:2311.10388 (2024).

[51] A. Haque, M. P. Singh, Extracting norms from contracts via chatgpt: Opportunities and challenges, arXiv preprint arXiv:2404.02269 (2024).

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

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

[54] 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).

[55] 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).

[56] R. S. Wahidur, I. Tashdeed, M. Kaur, H.-N. Lee, Enhancing zero-shot crypto sentiment with fine-tuned language model and prompt engineering, IEEE Access (2024).

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

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

[59] Y. Gong, Dynamic large language models on blockchains, arXiv preprint arXiv:2307.10549 (2023).

[60] 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.

[61] 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).

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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).

[70] 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.

[71] 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.

[72] 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.

[73] 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).

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] Z. He, Z. Li, A. Qiao, X. Luo, X. Zhang, T. Chen, S. Song, D. Liu, W. Niu, Nurgle: Exacerbating resource consumption in blockchain state storage via mpt manipulation, in: 2024 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, 2024, pp. 125–125.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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.

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

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

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

[94] T. Chen, Z. Li, X. Luo, X. Wang, T. Wang, Z. He, K. Fang, Y. Zhang, H. Zhu, H. Li, et al., Sigrec: Automatic recovery of function signatures in smart contracts, IEEE Transactions on Software Engineering 48 (2021) 3066–3086.

[95] 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.

[96] Z. He, Z. Liao, F. Luo, D. Liu, T. Chen, Z. Li, Tokencat: detect flaw of authentication on erc20 tokens, in: ICC 2022-IEEE International Conference on Communications, IEEE, 2022, pp. 4999–5004.

[97] 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.

[98] 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.

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

[100] 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.

[101] 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.

[102] 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.

[103] 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.

[104] 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.

[105] F. Luo, R. Luo, T. Chen, A. Qiao, Z. He, S. Song, Y. Jiang, S. Li, Scvhunter: Smart contract vulnerability detection based on heterogeneous graph attention network, in: 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE), IEEE Computer Society, 2024, pp. 954–954.

[106] Z. He, S. Song, Y. Bai, X. Luo, T. Chen, W. Zhang, P. He, H. Li, X. Lin, X. Zhang, Tokenaware: Accurate and efficient bookkeeping recognition for token smart contracts, ACM Transactions on Software Engineering and Methodology 32 (2023) 1–35.

[107] 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.

[108] Z. Liao, S. Song, H. Zhu, X. Luo, Z. He, R. Jiang, T. Chen, J. Chen, T. Zhang, X. Zhang, Large-scale empirical study of inline assembly on 7.6 million ethereum smart contracts, IEEE Transactions on Software Engineering 49 (2022) 777–801.

[109] 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.

[110] T. Chen, R. Cao, T. Li, X. Luo, G. Gu, Y. Zhang, Z. Liao, H. Zhu, G. Chen, Z. He, et al., Soda: A generic online detection framework for smart contracts., in: NDSS, 2020.

[111] 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.

[112] 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.

[113] Z. Li, J. Li, Z. He, X. Luo, T. Wang, X. Ni, W. Yang, X. Chen, T. Chen, Demystifying defi mev activities in flashbots bundle, in: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 165–179.

[114] 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.

[115] R. Gan, L. Wang, X. Ruan, X. Lin, Understanding flash-loan-based wash trading, in: Proceedings of the 4th ACM Conference on Advances in Financial Technologies, 2022, pp. 74–88.

[116] 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.

[117] R. Gan, L. Wang, X. Lin, Why trick me: The honeypot traps on decentralized exchanges, arXiv preprint arXiv:2309.13501 (2023).

[118] 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.

[119] 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.

[120] 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.

[121] K. Zhao, Z. Li, J. Li, H. Ye, X. Luo, T. Chen, Deepinfer: Deep type inference from smart contract bytecode, in: FSE, 2023.

[122] S. Wu, Z. Li, L. Yan, W. Chen, M. Jiang, C. Wang, X. Luo, H. Zhou, Are we there yet? unraveling the state-of-the-art smart contract fuzzers, ICSE (2024).

[123] 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).

[124] 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.

[125] 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.

[126] 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.

[127] 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.

[128] 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).

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

[130] 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).

[131] 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.

arXiv:2406.04076v1 [cs.CR] 6 Jun 2024

Federated TrustChain: Blockchain-Enhanced LLM Training and Unlearning

Xuhan Zuo, Minghao Wang, Tianqing Zhu*, Lefeng Zhang, Dayong Ye, Shui Yu, *Fellow, IEEE,* Wanlei Zhou, *Senior Membership, IEEE*

***Abstract*—The development of Large Language Models (LLMs) faces a significant challenge: the exhausting of publicly available fresh data. This is because training a LLM needs a large demanding of new data. Federated learning emerges as a promising solution, enabling collaborative model to contribute their private data to LLM global model. However, integrating federated learning with LLMs introduces new challenges, including the lack of transparency and the need for effective unlearning mechanisms. Transparency is essential to ensuring trust and fairness among participants, while accountability is crucial for deterring malicious behaviour and enabling corrective actions when necessary. To address these challenges, we propose a novel blockchain-based federated learning framework for LLMs that enhances transparency, accountability, and unlearning capabilities. Our framework leverages blockchain technology to create a tamper-proof record of each model's contributions and introduces an innovative unlearning function that seamlessly integrates with the federated learning mechanism. We investigate the impact of Low-Rank Adaptation (LoRA) hyperparameters on unlearning performance and integrate Hyperledger Fabric to ensure the security, transparency, and verifiability of the unlearning process. Through comprehensive experiments and analysis, we showcase the effectiveness of our proposed framework in achieving highly effective unlearning in LLMs trained using federated learning. Our findings highlight the feasibility of integrating blockchain technology into federated learning frameworks for LLMs.**

***Index Terms*—LLM, Federated Learning, Unlearning, Blockchain, Privacy.**

I. Introduction

The evolution of Large Language Models (LLMs) marks the beginning of a new era in artificial intelligence, significantly altering how we interact with and utilize machine learning [1], [2]. As these models progress, a significant challenge becomes apparent: by 2030, publicly available data sources are expected to be insufficient to support the continued growth and development of LLMs [3]. Therefore, the use of private data becomes crucial, not only to sustain development but also as an essential resource for LLMs to access.

With this demand, a significant challenge persists: data owners, aware of the value of LLMs, are hesitant to share their private data because of privacy concerns. At present, individuals have the option to download models and train them on their own private datasets. This method, however, leads to the development of isolated models. These models lack synergy and do not benefit from interconnected learning among various LLMs, underscoring the need for a more cohesive strategy to efficiently utilize private data.

Federated learning emerges as a prominent solution to address the pressing requirement for private data to enhance LLMs [4]. This method of collaborative machine learning enables the training of a model on multiple decentralized devices or servers, each of which holds a portion of the entire dataset [5]. This approach guarantees that confidential information remains on the owner's device, eliminating the need to distribute or consolidate data, thus directly addressing privacy concerns.

However, merging federated learning with LLMs presents a series of new challenges. One major concern is the lack of transparency in the federated learning process when combined with LLMs. The decentralized nature of federated learning makes it difficult to track and verify the contributions of each participating model, as well as to ensure that the collective learning process is not negatively impacted by suboptimal or compromised models. Additionally, the need for effective unlearning mechanisms becomes crucial in this context, as data owners may wish to remove their data from the training process while minimizing the impact on other participants [6].

To address these challenges and enhance the transparency and accountability of federated learning in LLM training, we propose the integration of blockchain technology. Blockchain's immutable and distributed ledger provides a secure and transparent record of all transactions and interactions within the federated learning process [7]. By leveraging blockchain, we can create a tamper-proof record of each model's contributions, facilitating the identification and removal of problematic models without disrupting the overall learning process.

Furthermore, blockchain enables the implementation of effective unlearning mechanisms, ensuring that data owners can remove their data from the training process while maintaining the integrity of the collective model. Through these dedicated efforts, we introduce an innovative solution that utilizes blockchain technology's strengths to overcome the intricate challenges of training LLMs with private data within a federated learning framework. Our approach represents a substantial step forward in achieving a secure, efficient, and transparent methodology for integrating private data into LLM development.

In addressing the challenges previously outlined, our work

*Tianqing Zhu is the corresponding author with Faculty of Data Science, City University of Macau, Macao (E-mail: tqzhu@cityu.edu.mo)

Xuhan Zuo, Dayong Ye and Shui Yu are with School of Computer Science, University of Technology Sydney, Ultimo 2007, Australia (E-mail: Xuhan.Zuo-1@student.uts.edu.au; Dayong.Ye@uts.edu.au; Shui.Yu@uts.edu.au)

Minghao Wang, Lefeng Zhang and Wanlei Zhou are with the Faculty of Data Science, City University of Macau, Macao (E-mail: sydminghao@gmail.com; lfzhang@cityu.edu.mo; wlzhou@cityu.edu.mo)

offers three significant contributions, each targeting a key aspect of merging federated learning with LLMs via blockchain technology:

- We present a blockchain-based architecture meticulously documenting every facet of the federated learning training process. This architecture is crucial for facilitating effective unlearning, as it provides a detailed and unchangeable record of all training actions, ensuring transparency and verifiability at every step.
- We introduce an unlearning function within this blockchain environment. This feature is designed to seamlessly integrate with the federated learning mechanism, enabling the targeted removal of specific models or data while preserving the integrity of the wider learning system. Its deployment is vital for upholding the federated learning framework's integrity and effectiveness, allowing it to dynamically respond to changing data privacy requirements.
- Our approach strengthens the accountability and verification process by methodically recording unlearning actions on the blockchain. This procedure is essential for evaluating the unlearning process's success.

The structure of the paper is as follows: Section II provides a comprehensive review of the existing literature on federated LLMs, unlearning with LLMs, and blockchain's role in enhancing LLMs. Section III lays out fundamental concepts crucial to our discussion, including federated learning, LLMs, LoRA Finetuning, and blockchain technology. Section IV defines the problem and outlines the system model, preparing the groundwork for Section V, which unveils our blockchain-based framework for federated learning. Section VI delves into the privacy and security evaluations of our framework, whereas Section VII measures its overall effectiveness. Finally, Section VIII wraps up the paper by summarizing our key findings and proposing avenues for future investigation.

II. Related Work

Large Language Models (LLMs) mark a significant breakthrough in natural language processing (NLP), distinguished by their capability to comprehend, interpret, and produce text that closely mimics human language [8]. Prominent examples of these models include GPT (Generative Pre-trained Transformer) [9] and BERT (Bidirectional Encoder Representations from Transformers) [10], which are trained on vast collections of textual data. This extensive training process equips LLMs with a profound understanding of linguistic subtleties, empowering them to support a broad spectrum of applications. These range from enhancing text completion functionalities to powering complex question-answering systems.

To fully grasp the current research landscape in integrating Large Language Models (LLMs) with federated learning and blockchain technology, we review three key areas: federated learning with LLMs, unlearning mechanisms in LLMs, and the application of blockchain to LLMs. Federated learning allows training LLMs on decentralized datasets while preserving privacy, but introduces challenges like ensuring model integrity and enabling efficient unlearning. Unlearning is crucial for maintaining data privacy and regulatory compliance. Blockchain technology can potentially enhance the security, transparency, and verifiability of federated learning and unlearning in LLMs. Reviewing these interconnected areas helps identify state-of-the-art approaches, limitations, and opportunities for synergistic integration.

A. Federated LLM

In addressing the exhaustion of public data resources for LLM training, federated learning emerges as a potent solution. By enabling multiple participants to collaboratively train a model without sharing their raw data, federated LLM can access a wider array of diverse and representative datasets.

Chen et al. conclude the concept of federated Large-Scale Language Models (LLMs), which includes federated pre-training, fine-tuning, and prompt engineering, and explore the unique challenges and potential engineering strategies within this framework, highlighting its advantages over traditional LLM training approaches [11]. In Gupta et al. study [12], they introduce FILM, a novel attack methodology for federated learning of language models. They demonstrate for the first time the feasibility of recovering text data from large batch sizes and evaluating various defence strategies, thereby suggesting new directions for enhancing privacy in language model training. This paper [4] proposed an industrial federated learning framework, which is designed to facilitate the efficient training of large language models. This framework addressed the dual challenges of computational resources and data privacy. The LP-FL methodology prioritizes the reduction of model parameters within the federated learning framework [13], this method employs Low-Rank Adaptation (LoRA) technology to construct compact learnable parameters, enabling effective local model fine-tuning and sustainable global model federation. FederatedScope-LLM (FS-LLM) provides a robust framework for optimizing LLM in a network. FS-LLM processes from data preparation to outcome assessment and facilitating diverse computational strategies [14]. Therefore, there are limitations among these frameworks due to the lack of transparency in the LLM training processes. [15] proposed an automated data quality control pipeline for federated fine-tuning of LLM, by utilizing data valuation algorithms, this pipeline assesses the quality of training samples across collaborative platforms, thereby enhancing model performance while preserving data privacy. There are also research concerns in the wireless field, [16] addresses significant challenges, including privacy concerns, inefficient data handling, and high communication costs, and demonstrates the effectiveness of these methods through simulations. In the Ro et al. research, they demonstrate that scale-invariant modifications to the Coupled Input Forget Gate (CIFG) and transformer models significantly enhance federated learning performance by improving convergence speeds and offering an improved privacy-utility trade-off [17].

B. Unlearning with LLM

The rapid advancements in large language models (LLMs) have led to remarkable breakthroughs in natural language processing and artificial intelligence. However, as these models are trained on vast amounts of data, they may inadvertently learn and perpetuate undesirable behaviors, biases, and harmful information. To address this issue, researchers have recently turned their attention to the concept of unlearning in LLMs.

In [18] paper, the authors explore the novel concept of unlearning in large language models (LLMs). They present a method that utilizes only negative examples to efficiently remove undesirable behaviors, demonstrating its effectiveness in alignment while significantly reducing computational resources compared to traditional reinforcement learning from human feedback (RLHF). While there is another paper introduces a data-driven unlearning approach for large language models (LLMs), utilizing a fine-tuning method informed by the importance of weights and relabeling during the pre-training phase of LLMs [19]. This method adjusts word embedding, involving identifying and neutralizing bias vectors within the embedding space to prevent biased associations. Wang et al. [20] proposed an unlearning framework called Knowledge Gap Alignment (KGA), emphasizing its capability to efficiently handle large-scale data removal requests with significant accuracy. However, the inability of KGA to guarantee the complete removal of data influences also faces the challenge of maintaining extra data sets and models. Si et al. [21] explores the technical challenges of knowledge unlearning in large language models (LLMs), specifically introducing parameter optimization, parameter merging, and in-context learning as methods to efficiently remove harmful or biased data while maintaining the integrity of the models. This approach not only advances the field of responsible AI but also opens new avenues for enhancing data privacy and model impartiality. Huang et al. claim an innovation offset unlearning framework tailored for the black box LLM [22]. This framework effectively addresses the challenge of unlearning problematic training data in LLMs without requiring access to internal model weight, thus offering a versatile solution for adapting current unlearning algorithms.

C. Blockchain with LLM

Blockchain technology and artificial intelligence (AI) have emerged as two of the most transformative technologies of our time. The integration of these technologies has the potential to revolutionize various industries and address critical challenges faced by both domains. Recent research has explored the synergistic relationship between blockchain and AI, particularly focusing on the integration of blockchain with large language models (LLMs) and generative AI (GAI) techniques.

Luo et al. [23] introduce the concept of "Blockchain for LLM" (BC4LLM), which aims to empower LLMs with the superior security features of blockchain technology, enabling reliable learning corpora, secure training processes, and identifiable generated content. This paper presents emerging solutions that showcase the effectiveness of GAI in detecting unknown blockchain attacks and smart contract vulnerabilities, designing key secret sharing schemes, and enhancing privacy. Through a case study, they demonstrate that the generative diffusion model, a GAI approach, can significantly optimize blockchain network performance metrics, outperforming traditional AI approaches in terms of convergence speed, rewards, throughput, and latency [24]. Mboma et al. propose a novel approach to combat academic document fraud by integrating Large Language Models (LLMs), specifically the Bidirectional Encoder Representations from Transformers (BERT), with blockchain and Interplanetary File System (IPFS) technologies to pre-validate academic documents before certification [25]. LLMChain, a decentralized blockchain-based reputation system, assists users and entities in identifying the most trustworthy LLM for their specific needs while providing valuable information to LLM developers for model refinement [26]. This framework demonstrated through evaluation across two benchmark datasets, making it a significant contribution to the field of trustworthy and transparent LLM assessment.

D. Conclusion

Despite progress in federated learning, unlearning, and blockchain integration with LLMs, several common limitations persist:

- Lack of comprehensive frameworks that integrate these approaches for enhanced security and transparency.
- Limited scalability and efficiency of current unlearning mechanisms in large-scale federated learning settings.
- Insufficient privacy and security guarantees in federated learning, with potential for attacks or information leakage.
- Absence of standardized frameworks and protocols for integrating these technologies, hindering interoperability and adoption.

Addressing these limitations requires developing a comprehensive framework that integrates federated learning, efficient unlearning, and blockchain technology to enable secure, transparent, and privacy-preserving LLM training on decentralized datasets.

III. Preliminary

A. Federated Learning

Federated Learning (FL) is a distributed machine learning approach that allows multiple devices or servers, each possessing its own local data samples, to collaboratively develop a model without the need to share their data directly [27]. This concept can be mathematically represented as:

$$FL = \{D_1, D_2, \ldots, D_n\} \tag{1}$$

where D_i denotes the local dataset of the i-th participant in the federation, and n represents the total number of participants. The core aim of FL is to build a comprehensive global model G that assimilates the knowledge from all local datasets, thereby improving model efficacy and ensuring data privacy.

The federated learning training protocol unfolds through several essential steps:

1) Initially, a global model G is distributed among all participants.

2) Each participant i refines this global model using their own data D_i, resulting in an updated local model M_i.
3) These updated local models M_i are then consolidated to refine the global model G, utilizing secure aggregation techniques to protect the privacy of individual updates.

The process undergoes multiple iterations, with the global model G being incrementally improved in each round. The aggregation function, often employing a form of weighted average, is pivotal in merging the local updates into a cohesive global model. This can be represented mathematically as:

$$G = Agg(M_1, M_2, \ldots, M_n) \tag{2}$$

where Agg denotes the aggregation mechanism used to combine the updates.

B. Large Language Models (LLMs)

The architecture of LLMs is fundamentally based on transformer models, characterized by a series of layers that systematically process the input text data [28]. At the heart of these models is the self-attention mechanism, a crucial feature that enables LLMs to assess the significance of each word in a sentence, thereby crafting responses that are contextually coherent [29]. The mathematical representation of an LLM's output can be succinctly expressed as:

$$O = F(I; \theta) \tag{3}$$

The equation represents the relationship where I is the input text, O the output generated by the model, F the function embodied by the LLM, and θ the set of parameters honed during training. The training process for LLMs involves fine-tuning these parameters (θ) to reduce the discrepancy between the model's output and the expected output, enhancing the model's precision in generating relevant responses. The sheer size of LLMs, with potentially billions of parameters, endows them with exceptional levels of language comprehension and production.

However, deploying LLMs is not without its hurdles. The need for extensive datasets for training and considerable computational power are significant barriers [30]. Furthermore, incorporating LLMs into federated learning environments brings extra challenges, including preserving model performance and privacy across decentralized data sources.

C. LoRA Fine-tuning

LoRA (Low-Rank Adaptation) [31] offers an innovative method for fine-tuning Large Language Models (LLMs) that balances efficiency with effectiveness, especially valuable in scenarios demanding model adaptability and computational thrift. Unlike traditional approaches that modify the original model parameters, LoRA adapts pre-trained LLMs to specific tasks through a low-rank decomposition technique. This method introduces additional trainable parameters, enabling the model to undergo task-specific adjustments without direct changes to its foundational parameters.

LoRA is particularly well-suited for federated learning environments, as it allows for efficient and targeted fine-tuning of LLMs across multiple participants without the need to share the entire model. Additionally, LoRA's low-rank decomposition approach makes it an ideal candidate for enabling effective unlearning mechanisms, as it allows for the selective modification of specific model components without affecting the overall model performance.

The core of LoRA's innovation lies in its approach to modifying the attention and feed-forward layers of transformer-based Large Language Models (LLMs) by integrating low-rank matrices. Specifically, for a weight matrix $W \in \mathbb{R}^{m \times n}$ within a transformer layer, LoRA introduces two smaller matrices, $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times n}$, where $k \ll \min(m, n)$. The adaptation of the original weight matrix W can be mathematically described as:

$$W' = W + AB \tag{4}$$

where W' denotes the adapted weight matrix, and AB is the low-rank update applied to W.

Such a strategy enables substantial customization of the model with only a modest increase in parameters, maintaining the extensive knowledge of the pre-trained LLM while introducing task-specific adjustments efficiently. During the fine-tuning phase, only the low-rank matrices A and B are updated, significantly lowering the computational demands typically seen with large-scale model training. Consequently, LoRA's approach to fine-tuning offers a scalable, resource-effective method for tailoring LLMs to diverse tasks and sectors. This is especially advantageous in federated learning settings, where computational efficiency and the flexibility to adapt to various tasks are paramount.

D. Blockchain

Blockchain technology is a decentralized ledger system that offers a secure and transparent method for recording transactions across multiple computers [32]. Its foundation relies on cryptography principles, ensuring that each entry in the ledger is immutable and verifiable [33]. While this technology is the backbone of cryptocurrencies like Bitcoin and Ethereum, it also extends its applications to secure transactional data in various sectors, including supply chain management, healthcare, and, as explored in this paper, federated learning environments.

A blockchain comprises a sequence of blocks, each containing a list of transactions. These blocks are interconnected through a cryptographic hash, linking each block to its predecessor and forming a chain. This structure is mathematically represented as $B_1 \to B_2 \to \ldots \to B_n$, where B_i symbolizes the i-th block in the chain, and n represents the total number of blocks. The integrity of the chain is preserved by consensus algorithms, which ensure that all instances of the distributed ledger are synchronized and in agreement on the transaction sequence.

IV. Problem Definition and System Model

A. Problem Definition

The integration of Large Language Models (LLMs) with federated learning, supported by a blockchain framework, introduces distinct aims that require a precise problem definition.

These aims arise from the complexities of managing private data, ensuring model integrity, and implementing efficient unlearning processes. We formalize these aims as follows:

1) **Data Privacy and Model Efficacy:** Federated learning aims to train LLMs on a collection of private datasets (D_c) across various clients (C_{id}) without breaching data privacy. The main challenge is to enhance the global LLM (LLM_g) performance while respecting privacy constraints, posing an optimization problem of maximizing LLM_g's efficacy across the federated network without direct access to D_c.
2) **Model Integrity and Security:** Within federated learning, each client boosts the global model by updating parameters using their local data. This decentralized method, however, exposes vulnerabilities like the potential for backdoor attacks or model tampering. It is crucial to secure the global model (LLM_g) and the aggregation process (A_{id}, JWT), especially when model updates come from possibly unreliable sources.
3) **Efficient Unlearning Mechanisms:** The changing dynamics of data privacy laws and data itself demand an effective mechanism for removing specific data (D_{forget}) from the trained model (LLM_g). The challenge lies in developing a process that allows LLM_g to selectively discard D_{forget} through unlearning epochs (E_u) and LoRA parameters (λ), with minimal detriment to the model's overall performance.
4) **Immutable Record Keeping and Verification:** The decentralized nature of federated learning complicates the monitoring and validation of model updates, contributions, and unlearning activities. It is vital to establish a transparent and unchangeable record-keeping system on a blockchain (SC, T_{id}) that logs all actions related to model training, updating, and unlearning. This system must support the authentication of actions ($parameters$, $D_{validate}$) to maintain integrity and accountability in the federated learning process.

Our proposed blockchain-based framework seeks to achieve these aims by employing cryptography techniques (JWT, P_k, S_k) for secure client registration, ensuring model integrity through a secure aggregation process, enabling efficient unlearning, and maintaining an immutable ledger for action verification. This strategy aims to improve the privacy, security, and effectiveness of LLM development within a federated learning framework.

B. System Model

Our system model achieve these aims by integrating Large Language Models (LLMs) with federated learning, underpinned by the security and immutability of blockchain technology. The model encompasses the processes of client registration, federated learning training, model aggregation, and the unlearning process, each facilitated by smart contracts (SC) on a blockchain network. Below, we detail the components and their interactions within the system.

1) Participants: The system includes several types of participants, each playing a pivotal role in the federated learning ecosystem:

- **Clients** (C_{id}): Entities with private datasets (D_c) looking to contribute to and benefit from the global LLM (LLM_g) without sacrificing data privacy.
- **Agents** (A_{id}): Individuals responsible for managing the aggregation of local model updates into the global model and facilitating the unlearning process. Agents operate with verification and authorization provided by JWTs, ensuring secure interactions.
- **Smart Contracts** (SC): Autonomous programs on the blockchain executing predefined operations such as client registration, model aggregation, and the execution of the unlearning process, thereby ensuring transparency, security, and trust.

2) Process Flow: The system model revolves around key processes, orchestrated through the interaction of participants:

- **Client Registration:** Clients (C_{id}) register in the system through a secure process involving the generation of a public-secret key pair (P_k, S_k) and obtaining a JSON Web Token (JWT) for secure communication. This process guarantees each client's unique identification and secure authentication within the system.
- **Federated Learning with LLM Training:** Clients engage in the federated learning process by locally training the LLM on their private datasets (D_c) and sharing the learned parameters with the global model (LLM_g), all while keeping their data confidential. This iterative process across multiple epochs aims to enhance the global model's precision and robustness.
- **Model Aggregation:** Agents (A_{id}), verified via JWTs, consolidate the parameters from clients into the global model (LLM_g). Smart contracts (SC) secure and oversee this aggregation process, ensuring only authorized updates enhance the global model.
- **Unlearning Process:** The system facilitates an efficient unlearning mechanism allowing the selective omission of data (D_{forget}) from the global model (LLM_g). Utilizing unlearning epochs (E_u) and specific parameters (λ), the model adjusts without losing learning from other data contributions.
- **Blockchain for Security and Transparency:** All activities, including client registration, model updates, and the unlearning actions, are recorded on the blockchain via smart contracts (SC). This immutable ledger elevates the system's security, transparency, and trust.

V. Proposed Framework

A. Overview

Our proposed system introduces a novel framework that seamlessly integrates Large Language Models (LLMs) with federated learning, leveraging the security and transparency provided by blockchain technology. This meticulously designed integration aims to harness the advantages of federated learning for training LLMs on decentralized private datasets while preserving data privacy, ensuring model integrity, and facilitating an efficient unlearning process.

Figure 1 presents an overview of our proposed system. To begin, all clients must complete the registration process

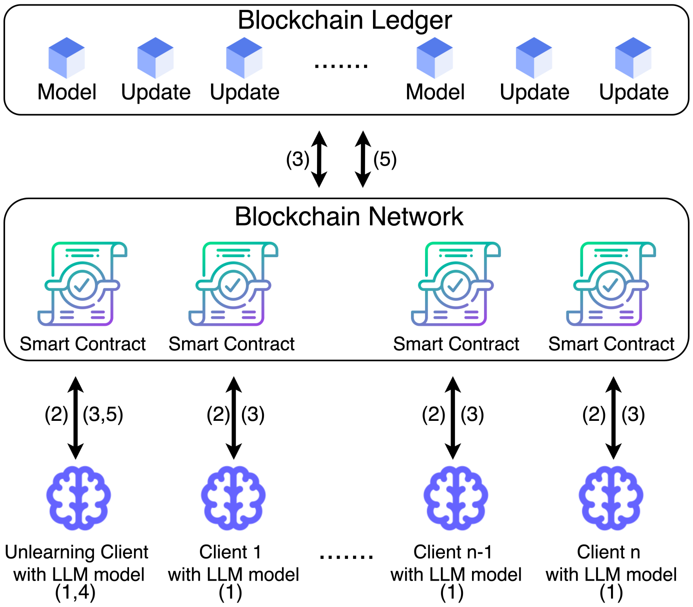


Fig. 1. Overview and process of our proposed system. (1) Client register. (2) Federated learning LLM training process. (3) Model aggregation process. (4) Unlearning process using LoRA for forgetting. (5) Unlearning verification and submitting unlearning results.

within the blockchain network. Once registration is finalized, the blockchain network initiates the federated learning training process. The global model is transferred within the blockchain network through a smart contract, followed by the aggregation of the model. In the event that a client wishes to erase their private data, the unlearning process is triggered, employing LoRA to facilitate efficient forgetting. Subsequently, a verification process is conducted to ensure the integrity of the unlearning procedure. Upon successful verification, the system seamlessly returns to the standard federated learning training process. The implementation details of our meticulously designed framework are outlined below.

B. Client Register

In our proposed framework, every *Client* need to enroll in blockchain network first. Algorithm 1 facilitates a straightforward method for registering a client using a unique identifier and securing their communication with a JSON Web Token (JWT). Initially, the process verifies if the client's unique identifier (C_{id}) is already present in the user pool (U_{pool}). If the identifier exists, the registration halts, indicating the client already exists. Otherwise, the algorithm proceeds to generate a public-secret key pair using the $keyGenerator()$ function. With these keys, it then creates a *JWT* for the client. This *JWT*, along with the client ID, is securely stored, effectively registering the client. The user pool is updated to include the new client ID, marking the registration successful. The algorithm concludes by returning a success status and the generated *JWT*, signifying the client's successful registration and their secure token for future communications. This process ensures a secure registration framework by leveraging cryptographic keys and *JWT*s, ensuring both security and simplicity in client management.

Algorithm 1 Client Register

```
Require: $C_{id}$, keyGenerator(),generateJWT()
Ensure: RegisterSucess, jwt token
RegisterSucess = False;
if $C_{id} \in U_{pool}$ then
    return $C_{id}$ already existed.
end if
$(P_k, S_k) \leftarrow$keyGenerator();
jwt = generateJWT($P_k$, $S_k$);
$C_{id} \leftarrow jwt \leftarrow$ SC;
$U_{\text{pool}} = U_{\text{pool}} \cup C_{\text{id}}$;
RegisterSuccess = True;
return RegisterSucess, jwt
```

C. Federated Learning with LLM Training Process

The federated learning process for training large language models (LLMs) involves multiple clients collaborating to improve a global model without sharing their private data directly. This process ensures data privacy, security, and decentralization. First, an agent initiates the process by sending the LLM to the smart contract (SC). The SC verifies the agent's identity using a JSON Web Token (jwt) and uploads the global model (LLM_g) to the blockchain network. The LLM_g is then distributed to the participating clients. During each training epoch, clients perform federated learning on their private datasets to improve the LLM_g. After training, the clients send the updated LLM parameters to the SC for verification and aggregation. The SC verifies each client's identity using their jwt tokens and publishes the updated parameters and client information to the blockchain network. This process is repeated for a specified number of epochs. Upon completion, the algorithm returns the status of the LLM upload and training process.

Algorithm 2 outlines the steps for training a large language model (LLM) in a federated learning environment, emphasizing security and decentralization.

The algorithm begins by initializing two boolean variables, *UploadSuccess* and *TrainingProcess*, to False. These variables track the status of the LLM upload and the training process, respectively. The required inputs include the client identifier (C_{id}), agent identifier (A_{id}), JSON Web Token (jwt) for authentication, number of training epochs, and the LLM to be trained.

The SC first verifies the agent's identity using the provided jwt. If the token is invalid or has expired, the process is terminated, and an error message is returned. Upon successful authentication, the agent sends the LLM to the SC, which then verifies and uploads the global model (LLM_g) to the blockchain network, ensuring the model's integrity and security in a decentralized environment. The LLM_g is initialized with the agent's LLM, and the *UploadSuccess* variable is set to True, indicating the successful upload of the model.

The SC then distributes the LLM_g to the participating clients for training. The training process is conducted iteratively for a specified number of epochs (n). During each epoch, clients perform federated learning on their private datasets

Algorithm 2 Client LLM Training Process

```
Require: $C_{\text{id}}, A_{\text{id}}$, jwt, $epochs, LLM$
Ensure: UploadSuccess, TrainingProcess
UploadSuccess, TrainingProcess = False;
SC check Agent's identity;
if Agent's jwt token ineligibility then
    return Agent jwt token expired
end if
Agent sends $LLM$ to SC;
SC verifies and uploads the global model $LLM_g$ to the blockchain network;
$LLM_g$ = $LLM$;
UploadSuccess = True;
SC send the $LLM_g$ to $Client$;
for $epoch = 1$ to $n$ do
    $Clients$ do the federated learning training process according to their different private dataset $D_c$ for $LLM_g$
    $Clients$ send the parameters of $LLM$ to SC
    SC verify the $Client$ identity
    if Client's jwt token ineligibility then
        return Client identity check false
    else
        SC publish the $parameters$ and $Clients$ information in blockchain network
    end if
end for
TrainingProcess = True;
return UploadSuccess, TrainingProcess
```

(D_c) to improve the LLM_g. After training, the clients send the updated LLM parameters to the SC for verification and aggregation.

The SC verifies each client's identity using their jwt tokens. If a client's token is invalid, the process is terminated for that client, and an error message is returned. Otherwise, the SC publishes the updated parameters and client information to the blockchain network, ensuring transparency and security. Upon completing the specified number of training epochs, the *TrainingProcess* variable is set to True, indicating the successful completion of the federated learning process. Finally, the algorithm returns the values of *UploadSuccess* and *TrainingProcess*, providing information about the status of the LLM upload and training process.

D. Model Aggregation Process

The model aggregation process is a crucial step in updating the global language model (LLM_g) in a secure and decentralized manner. This process is initiated by an agent who requests the latest model parameters from the blockchain network. The smart contract (SC) verifies the agent's identity using a JSON Web Token (JWT). Upon successful authentication, the SC sends the parameters to the agent, who then updates the LLM and generates a new model version (LLM_n). The agent sends LLM_n back to the SC, which uploads it to the blockchain network, ensuring a secure and transparent record of the update. Finally, LLM_g is updated to reflect the changes in LLM_n, completing the model aggregation process.

Algorithm 3 outlines the procedure for aggregating updates to a large language model (LLM) in a secure and decentralized manner, leveraging a blockchain network for data integrity and transparency.

Algorithm 3 Model Aggregation Process

```
Require: $A_{id}$ JWT, $parameters$
Ensure: ModelAggregation, $LLM_g$
ModelAggregation = False;
Agent wants to get parameters from blockchain network;
SC check the Agent identity;
if Agent's jwt token ineligibility then
    return Agent identity check false
else
    SC send parameters to Agent
end if
Agent updating LLM according to parameters and generating new model $LLM_n$;
Agent send the new model $LLM_n$ to SC;
SC upload the $LLM_n$ to blockchain network;
$LLM_g \leftarrow LLM_n$ ;
ModelAggregation = True;
return ModelAggregation, $LLM_g$
```

The process begins by initializing the $ModelAggregation$ flag to False, indicating that the aggregation process has not yet started. An agent, identified by A_{id} and authenticated using a JWT, requests the latest model parameters from the blockchain network. These parameters will be used to update the LLM to a new version, LLM_n.

The SC verifies the agent's identity by checking the validity of the provided JWT. If the JWT is invalid, the process is terminated, and the agent is informed that their identity check has failed. This step ensures that only authorized agents can retrieve and update model parameters, maintaining the system's security.

If the agent's identity is successfully verified, the SC sends the requested parameters to the agent. The agent then uses these parameters to update the LLM, generating a new model version, LLM_n. This step involves applying the aggregated updates from various sources to improve the model's performance or capabilities based on newly acquired data or insights.

After generating LLM_n, the agent sends this new model version back to the SC. The SC uploads LLM_n to the blockchain network, ensuring that the update is securely and transparently recorded. The global version of the LLM, LLM_g, is then updated to reflect the changes in LLM_n, completing the model update process.

Finally, the $ModelAggregation$ flag is set to True, indicating the successful aggregation of the model updates. The algorithm returns this flag along with LLM_g, the updated global model, signifying the end of the aggregation process.

E. Unlearning Process

The unlearning process is a crucial step in selectively forgetting specific data from a large language model (LLM) due to data sensitivity or correction needs. This process begins with the initialization of a local version of the LLM (LLM_{local}) using the parameters of the global model (LLM_g). An adapter (A) is then constructed within LLM_{local} to facilitate the forgetting of the specified dataset (D_{forget}). The core of the unlearning process involves several epochs of training, where a forward pass of D_{forget} is performed through LLM_{local} to identify the features associated with the data points that need to be forgotten. Gradients are then computed for LLM_{local}, emphasizing the data to be unlearned. The Low-Rank Adaptation (LoRA) technique is applied to the adapter's gradients to focus the unlearning process on the identified features. Finally, LLM_{local}'s parameters are updated using the adjusted gradients and a specified learning rate, gradually leading to the forgetting of the specified data points. The algorithm returns the updated parameters, representing the outcome of the forgetting process.

Algorithm 4 describes the procedure for selectively forgetting specific data from a large language model (LLM).

```
**Algorithm 4** Unlearning Process using LoRA for Forgetting
**Require:** $LLM_g$, $D_{forget}$ (Dataset to forget), Learning rate
    $\eta$, Unlearning epochs $E_u$, LoRA parameters $\lambda$
**Ensure:** *parameters*
Unlearning Request due to data sensitivity or correction
  needs;
Initialize unlearning model $LLM_{local}$ with $LLM_g$
*Adapter* $A$ constructed for $LLM_{local}$ targeting forgetting
  process
**for** $epoch = 1$ to $E_u$ **do**
    Perform a forward pass with $D_{forget}$ through $LLM_{local}$
      to identify features to forget
    Compute gradients for $LLM_{local}$ emphasizing data
      points in $D_{forget}$ to be forgotten
    Apply LoRA to adjust gradients of adapter $A$ using
      parameters $\lambda$, focusing on unlearning
    Update $LLM_{local}$'s parameters using the adjusted gra-
      dients and learning rate $\eta$, facilitating forgetting
**end for**
Calculate the updating *parameters* indicative of the for-
  getting process between $LLM_{local}$ and $LLM_g$;
**return** *parameters*
```

The process begins with the need to remove certain data points from a global language learning model (LLM_g) due to their sensitivity or incorrectness. To achieve this, a local version of the LLM, denoted as LLM_{local}, is initialized with the parameters of LLM_g. An adapter, A, is then constructed within LLM_{local} specifically designed to target and facilitate the forgetting of the specified dataset, D_{forget}.

The core of the unlearning process involves several epochs of training, defined by the parameter E_u. In each epoch, the algorithm performs a forward pass of D_{forget} through LLM_{local} to identify the features associated with the data points that need to be forgotten. Following this, gradients are computed for LLM_{local} with an emphasis on the data to be unlearned, highlighting what needs to be forgotten.

The LoRA technique is applied to the adapter A's gradients using parameters λ. LoRA is instrumental in focusing the unlearning process by adjusting the gradients to specifically target the forgetting of the identified features. With these adjusted gradients, LLM_{local}'s parameters are updated using the specified learning rate η. This iterative process of adjustment and updating gradually leads to the forgetting of the specified data points from D_{forget}.

Upon completion of the unlearning epochs, the algorithm calculates the parameters that indicate the changes made to LLM_{local} in comparison to LLM_g. These parameters represent the outcome of the forgetting process, effectively capturing the essence of what has been unlearned.

The algorithm concludes by returning these updated parameters, signifying the successful exclusion of sensitive or incorrect data from the language model. Through this structured process, the algorithm ensures that the unlearning is specific, efficient, and aligned with the requirements of data sensitivity or correction, thereby maintaining the integrity and relevance of the LLM.

F. Unlearning Verification and Submitting Unlearning Results

The unlearning verification and submission process is a critical step in ensuring the integrity and transparency of the unlearning results in a large language model. The process begins with the client sending the updated parameters, resulting from an unlearning process, to the smart contract (SC). The SC validates the client's credentials through their JSON Web Token (JWT). If the client's identity is successfully verified, the SC initializes an updated version of the language learning model ($LLM_{updated}$) with the new parameters. The SC then employs a validation dataset ($D_{validate}$) to assess the efficacy of the unlearning process by calculating the training loss and accuracy of $LLM_{updated}$. If the unlearning results satisfy predefined verification criteria, the SC submits the updated parameters to a blockchain network. An agent downloads these parameters from the blockchain for weight integration into the global model. The SC records the updated model's weights on the blockchain, ensuring transparency and traceability. Additionally, the SC logs a Transaction ID (T_{id}), providing verifiable proof of submission and an integration request. The process concludes with the return of the Transaction ID, signifying the successful verification and submission of the unlearning results.

Algorithm 5 details the steps for verifying the results of an unlearning process in a large language model and subsequently submitting these results for integration and transparency.

The process commences with the client sending the updated parameters, resulting from an unlearning process, to the SC. These parameters are intended to modify a language learning model by excluding specific, potentially sensitive, or incorrect data. Initially, the client's credentials are validated through their JWT token. If the token does not pass the eligibility check, the process halts, indicating a failure in client identity verification.

Algorithm 5 Unlearning Verification and Submitting Unlearning Results

```
Require: parameters, Validation dataset $D_{validate}$, Client
Ensure: parameters
Client send the parameters to SC;
if Client's jwt token ineligibility then
    return Client identity check false
end if
SC instantiate the updated language learning model $LLM_{updated}$ with the received parameters;
SC use the validation dataset $D_{validate}$ to evaluate $LLM_{updated}$. Calculate the training loss and accuracy to measure the impact of the unlearning process.
if Verification criteria are met then
    SC send the parameters to blockchain networks
    $Agent$ downloads parameters from blockchain network for weight integration.
    SC ensuring that the updated weights are recorded on the blockchain, providing transparency and traceability
    SC record the Transaction ID $T_{id}$, which serves as proof of submission and integration request, facilitating tracking and verification in the blockchain ledger.
end if
Continue for future federated learning process;
return $T_{id}$
```

Assuming successful verification, the SC then initializes an updated version of the language learning model ($LLM_{updated}$) with the new parameters. The SC employs a validation dataset ($D_{validate}$) to assess the efficacy of the unlearning process. This assessment involves calculating the training loss and accuracy of $LLM_{updated}$ to gauge the impact of the modifications.

If the unlearning results satisfy predefined verification criteria, which indicate that the data has been effectively forgotten without compromising the model's overall performance, the SC will submit the updated parameters to a blockchain network. This submission is not merely for record-keeping; an agent then downloads these parameters from the blockchain for weight integration into the global model.

Recording the updated model's weights on the blockchain ensures that the unlearning process is transparent and traceable. Furthermore, the SC logs a Transaction ID (T_{id}), providing verifiable proof of submission and an integration request. This ID facilitates tracking and verification within the blockchain ledger, offering a transparent audit trail of the changes made to the language model.

The process culminates with the return of the Transaction ID, signifying the successful verification and submission of the unlearning results. This structured approach not only secures the integrity of the model by removing unwanted data but also enhances accountability and transparency through blockchain technology.

VI. Privacy and Security Analysis

Our proposed blockchain-based federated learning framework with unlearning capabilities for Large Language Models (LLMs) is designed to address critical privacy and security challenges. By leveraging the inherent features of federated learning, blockchain technology, and efficient unlearning mechanisms, our approach provides a comprehensive solution for secure and privacy-preserving LLM training.

A. Privacy Analysis

Federated learning, a core component of our framework, enables the training of LLMs across multiple participants without the need for direct data sharing. This decentralized approach ensures that sensitive data remains within the control of each participant, minimizing the risk of data breaches and unauthorized access.

From a theoretical perspective, federated learning can be modeled as an optimization problem that aims to minimize the global loss function while keeping the data locally [34]. This can be represented as:

$$\min_{w} F(w) = \sum_{i=1}^{k} p_i F_i(w) \tag{5}$$

where w is the global model parameters, $F(w)$ is the global loss function, $F_i(w)$ is the local loss function of the i-th participant, p_i is the weight of the i-th participant, and k is the total number of participants.

By minimizing the global loss function, federated learning enables the optimization of the global model without directly sharing raw data, leveraging the data distributed across local participants while protecting privacy and improving model performance.

Our framework further enhances privacy protection by integrating blockchain technology, which provides a secure and immutable record of all transactions and interactions within the federated learning process. The use of smart contracts in our framework automates the execution of predefined rules and conditions, ensuring that all participants adhere to agreed-upon privacy policies. This automation minimizes the potential for human error and reduces the risk of unauthorized data access or manipulation.

Moreover, blockchain technology can provide privacy protection for the federated learning process [35]. By leveraging the immutability and distributed consensus mechanisms of blockchain, it ensures that all participants follow predefined privacy policies and prevents malicious behavior. In our framework, smart contracts automatically enforce these policies, further reducing the risks of human error and unauthorized data access.

The unlearning mechanism embedded in our framework allows for the selective removal of specific data points or model updates, enabling participants to maintain control over their data and comply with evolving privacy regulations. The unlearning process can be theoretically formulated as a constrained optimization problem [36], where the objective is to minimize the impact of the removed data on the model's performance while satisfying the unlearning constraints:

$$\min_{w} F(w) = \sum_{i=1}^{k} p_i F_i(w) \quad (6)$$
$$\text{s.t.} \quad w \in W_u$$

where W_u represents the feasible set of model parameters after unlearning. The goal is to minimize the impact of the removed data on the model's performance while satisfying the unlearning constraints. By introducing the unlearning mechanism, our framework provides participants with an effective way to control their data lifecycle, enhancing privacy protection.

By integrating federated learning, blockchain technology, and efficient unlearning mechanisms, our approach creates a robust, transparent, and secure environment for collaborative LLM development while preserving the privacy of individual participants.

B. Security Analysis

The integration of blockchain technology in our framework significantly enhances the security of the federated learning process. The immutable nature of blockchain ensures that all transactions and model updates are tamper-proof and easily verifiable.

From a theoretical standpoint, the security of a blockchain network can be analyzed using game theory and consensus mechanisms [37]. In a proof-of-work (PoW) based blockchain, the security is guaranteed by the assumption that honest nodes control the majority of the computing power, making it infeasible for attackers to tamper with the blockchain. This can be formalized as a game between honest nodes and attackers, where the honest nodes aim to maximize their rewards by following the protocol, while the attackers try to maximize their profits by deviating from the protocol. The Nash equilibrium of this game represents a state where no party can benefit by unilaterally changing their strategy, ensuring the stability and security of the blockchain network.

Our framework leverages cryptographic techniques, such as digital signatures and secure hash functions, to ensure the integrity and authenticity of all transactions. The use of digital signatures allows participants to verify the origin and authenticity of the data and model updates, preventing unauthorized modifications. Secure hash functions, such as SHA-256, are used to create a unique fingerprint of the data, ensuring its integrity. By combining these cryptographic primitives, our framework establishes a secure and trustworthy environment for federated learning.

The use of smart contracts further reinforces the system's security by automatically executing predefined rules and conditions, reducing the potential for unauthorized access or manipulation. Smart contracts are self-executing programs stored on the blockchain that enforce the terms of an agreement between parties. In our framework, smart contracts govern the federated learning process, ensuring that all participants adhere to the agreed-upon rules and conditions. This automated enforcement minimizes the risk of human error and malicious behavior, enhancing the overall security of the system.

The decentralized architecture of our framework, enabled by blockchain technology, eliminates single points of failure and distributes the risk across multiple nodes. This distributed approach makes it significantly more challenging for attackers to compromise the entire system, as they would need to control a majority of the participating nodes simultaneously, which is known as a 51% attack [38]. The probability of a successful 51% attack decreases exponentially with the number of honest nodes in the network, making it practically infeasible in a large-scale federated learning setting.

Furthermore, the unlearning mechanism in our framework, facilitated by the LoRA technique, allows for the targeted removal of specific data points or model updates without affecting the overall model performance. This selective unlearning capability not only enhances privacy but also serves as a security measure, enabling the swift removal of potentially malicious or corrupted data. By promptly removing suspicious data or updates, our framework minimizes the impact of security threats and maintains the integrity of the federated learning process.

In conclusion, our blockchain-based federated learning framework with unlearning capabilities provides a comprehensive solution for addressing security concerns in LLM training. By leveraging the inherent security features of blockchain technology, cryptographic techniques, and smart contracts, our approach creates a robust and secure environment for collaborative LLM development. The decentralized architecture and the ability to swiftly remove malicious data through unlearning further enhance the system's resilience against attacks, ensuring the integrity and reliability of the federated learning process.

VII. Performance Evaluation

This section presents a detailed evaluation of our proposed federated learning and blockchain framework, specifically focusing on its application with the GPT-2 model. Our primary objective is to investigate the influence of various LoRA configurations on the effectiveness of the unlearning process within this context. By manipulating the LoRA settings, we aim to discern their impact on the model's ability to selectively forget data—a crucial capability for maintaining data privacy and compliance with evolving regulations. The effectiveness of each configuration is quantitatively assessed through changes in model accuracy, providing a clear metric for comparing the performance across different settings. This evaluation not only highlights the practical implications of our approach but also helps in identifying optimal LoRA settings that enhance unlearning performance without compromising the overall accuracy of the GPT-2 model.

A. Experimental Configuration

To evaluate the effectiveness of our proposed blockchain-based federated learning framework with unlearning capabilities for Large Language Models (LLMs), we conducted a series of experiments focusing on the impact of various

LoRA configurations on the unlearning performance. The experiments were designed to assess the system's ability to selectively forget specific data points while maintaining the overall model accuracy.

Dataset: For our experiments, we utilized two datasets: the IMDB dataset and the Twitter dataset. The IMDB dataset is a widely-used benchmark dataset for sentiment analysis tasks, while the Twitter dataset provides real-world text data from the social media platform. We chose these datasets for several reasons:

- Relevance to LLM applications: Sentiment analysis is a common task for LLMs, and the IMDB dataset provides a representative sample of movie reviews, making it suitable for evaluating the performance of our framework in a well-established benchmark setting. On the other hand, Twitter data is highly relevant for various natural language processing tasks, such as sentiment analysis, topic modeling, and text classification. Using both datasets allows us to assess the performance of our framework in different contexts.
- Dataset size: With 50,000 reviews, the IMDB dataset is large enough to simulate a realistic federated learning scenario while still being manageable for experimental purposes. Similarly, the Twitter dataset contains a substantial number of tweets, providing a sufficiently large sample size to evaluate the scalability and efficiency of our proposed framework.
- Diversity of content: The IMDB dataset consists of movie reviews, which are relatively structured and focused on a specific domain. In contrast, tweets in the Twitter dataset cover a wide range of topics, opinions, and writing styles. By using both datasets, we can evaluate the robustness and adaptability of our framework in handling different types of text data.
- Presence of sensitive information: Twitter data often contains sensitive or personal information that users may wish to remove or forget. This characteristic makes the Twitter dataset particularly suitable for testing the effectiveness of our unlearning mechanism in selectively forgetting specific data points while preserving the overall model performance.

Evaluation Metrics: We selected accuracy as the primary evaluation metric for our experiments. Accuracy is a straightforward and intuitive measure that quantifies the proportion of correctly classified reviews after the unlearning process. By comparing the accuracy before and after unlearning, we can assess the effectiveness of our framework in selectively forgetting specific data points while maintaining the overall model performance.

Accuracy is particularly well-suited for evaluating the unlearning performance in our framework because:

- Direct measure of unlearning effectiveness: A successful unlearning process should remove the influence of specific data points on the model's predictions. By measuring the accuracy after unlearning, we can directly assess the extent to which the model has ”forgotten” the targeted data.
- Comparability across different configurations: Using accuracy as a standard metric allows us to compare the unlearning performance across various LoRA configurations, enabling us to identify the most effective settings for our framework.

Comparative Methods: Due to the novelty of our approach in integrating federated learning, blockchain technology, and unlearning capabilities for LLMs, there are currently no directly comparable methods available in the literature. Our framework is the first to address the challenge of selective unlearning in a federated learning setting for LLMs while ensuring data privacy and security through blockchain integration.

However, to provide a comprehensive evaluation of our framework, we compared the performance of different LoRA configurations within our system. By varying the LoRA hyperparameters, such as the rank and scaling factor, we aimed to identify the optimal settings that achieve the best balance between unlearning effectiveness and overall model accuracy.

The hardware setup for our experiments includes an Intel Xeon 6238R processor, 64GB of RAM, and an NVIDIA A6000 GPU, ensuring efficient handling of the computational tasks. The software environment consists of Ubuntu 20.04, Visual Studio Code, Hyperledger Fabric, FATE, and machine learning frameworks such as PyTorch and TensorFlow.

By focusing on the IMDB dataset, using accuracy as the primary evaluation metric, and comparing different LoRA configurations within our novel framework, we aim to provide a comprehensive and rigorous evaluation of our blockchain-based federated learning approach with unlearning capabilities for LLMs.

B. Results and Analysis

In this section, we present the results of our experiments and compare them with the Retrain from Scratch method. We focus on the effectiveness of our unlearning method in terms of accuracy reduction, highlighting the differences in performance and providing an analysis of why our method performs better or worse. Our experiments were conducted using different configurations of the LoRA method on both the IMDB and Twitter datasets.

We conducted experiments using different configurations of the LoRA method on the IMDB dataset. The Retrain from Scratch method serves as a benchmark for comparing the effectiveness of our unlearning approach. Some specific LoRA configurations used in our experiments are presented in Table I.

TABLE I
EXPERIMENTAL DATA (IMDB RESULTS)

LoRA Config	Initial Accuracy	Final Accuracy
r=8, alpha=4, dropout=0.3	99.15%	0.70%
r=16, alpha=2, dropout=0.2	97.75%	0.90%
r=32, alpha=4, dropout=0.1	94.30%	1.00%
r=8, alpha=4, dropout=0.4	98.45%	1.15%
r=32, alpha=4, dropout=0.4	95.15%	1.20%

Similarly, we also tested our method on the Twitter dataset. The results of these experiments are shown in Table II.

TABLE II
EXPERIMENTAL DATA (TWITTER RESULTS)

LoRA Config	Initial Accuracy	Final Accuracy
r=1, alpha=2, dropout=0.3	85.32%	8.27%
r=16, alpha=1, dropout=0.2	89.10%	9.72%
r=8, alpha=1, dropout=0.5	75.98%	10.06%
r=16, alpha=1, dropout=0.1	89.58%	10.47%
r=4, alpha=1, dropout=0.2	89.38%	10.63%

1) Unlearning Performance with Different alpha: Figure 2 illustrates the impact of different alpha values on the accuracy reduction of our LoRA-based unlearning method for the IMDB dataset. As the alpha value decreases, the final accuracy after unlearning generally decreases, with alpha=1 and alpha=2 achieving the lowest accuracies. This suggests that lower alpha values contribute to better unlearning performance in our approach. The improved accuracy reduction with lower alpha values can be attributed to the decreased capacity of the model to retain relevant information during the unlearning process, leading to more effective forgetting of target knowledge.

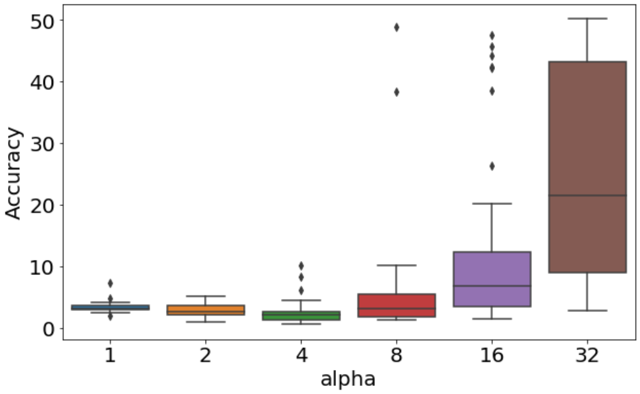


Fig. 2. Box Plot of Accuracy by Different Alpha Values (IMDB Dataset)

Similarly, Figure 3 illustrates the impact of different alpha values on the accuracy reduction of our LoRA-based unlearning method for the Twitter dataset. The trend observed is consistent with the IMDB dataset results. Lower alpha values lead to a greater reduction in final accuracy, indicating more effective unlearning. This further supports the notion that a decreased capacity to retain information facilitates better forgetting of targeted knowledge.

2) Unlearning Performance with Different droupout: Figure 4 depicts the effect of various dropout values on the accuracy reduction of our method for the IMDB dataset. The box plot reveals that dropout values of 0.4 and 0.5 generally lead to lower accuracies after unlearning compared to lower dropout values. This observation indicates that higher dropout regularization plays a crucial role in improving the unlearning performance. By introducing a significant level of noise during training, dropout helps the model forget specific data more effectively, resulting in better unlearning.

Similarly, Figure 5 depicts the effect of various dropout values on the accuracy reduction of our method for the Twitter dataset. The trend observed is consistent with the IMDB dataset results. Dropout values of 0.4 and 0.5 lead to a greater reduction in final accuracy, indicating more effective unlearning. This further supports the notion that higher dropout regularization improves the model's ability to forget specific data.

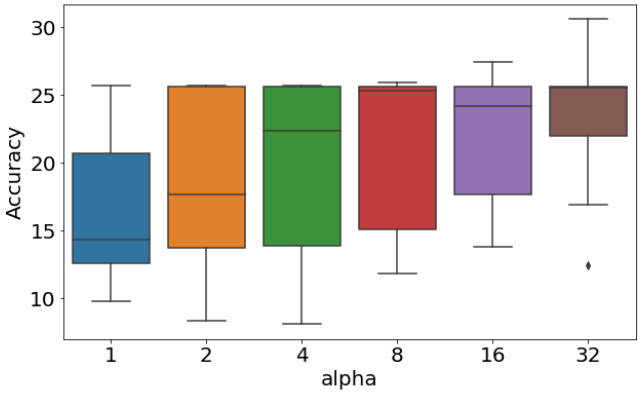


Fig. 3. Box Plot of Accuracy by Different Alpha Values (Twitter Dataset)

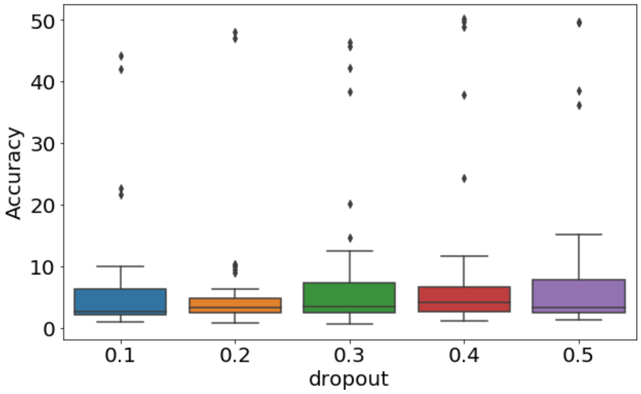


Fig. 4. Box Plot of Accuracy by Different Dropout Values (IMDB Dataset)

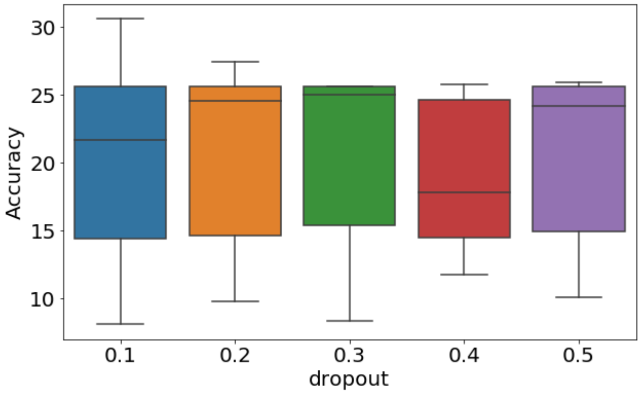


Fig. 5. Box Plot of Accuracy by Different Dropout Values (Twitter Dataset)

3) Unlearning Performance with Different rank: Figure 6 presents the relationship between different r values and the accuracy reduction of our LoRA-based unlearning method. The box plot shows that higher r values, particularly $r = 16$

and $r = 32$, tend to yield lower accuracies after unlearning compared to lower r values. This suggests that using a larger rank for the LoRA adaptation can be beneficial for unlearning performance. Higher r values may allow the model to capture more diverse information during unlearning, leading to better forgetting of target knowledge.

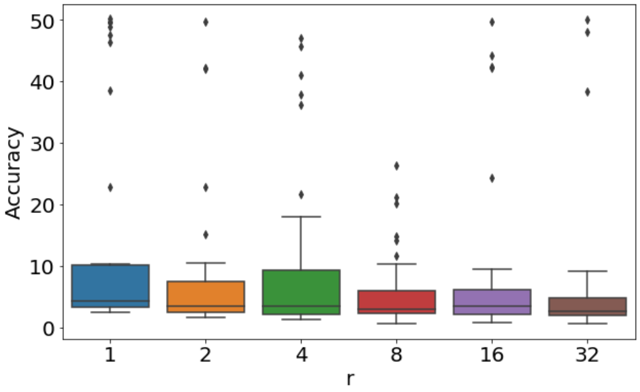


Fig. 6. Box Plot of Accuracy by Different r Values (IMDB Dataset)

Similarly, Figure 7 presents the relationship between different r values and the accuracy reduction of our LoRA-based unlearning method for the Twitter dataset. The trend observed is consistent with the IMDB dataset results. Higher r values, particularly $r = 16$, lead to a greater reduction in final accuracy, indicating more effective unlearning. This further supports the notion that a larger rank for the LoRA adaptation improves the model's ability to forget specific data.

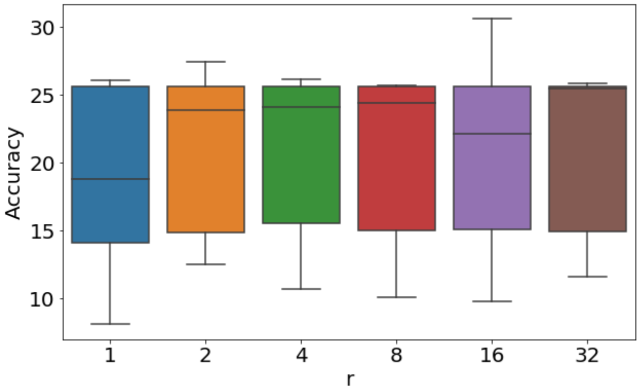


Fig. 7. Box Plot of Accuracy by Different r Values (Twitter Dataset)

4) Factors Influencing Performance: The analysis of the impact of alpha, dropout, and r values on accuracy reduction provides valuable insights into the factors influencing the effectiveness of our unlearning method. Both IMDB and Twitter datasets show that lower alpha values contribute to improved accuracy reduction by decreasing the model's capacity to retain relevant information. This trend is evident across both datasets, indicating that alpha is a critical parameter for controlling the unlearning process.

Higher dropout regularization consistently helps mitigate overfitting and enhances forgetting, leading to better unlearning performance in both datasets. Dropout values of 0.4 and 0.5 were particularly effective in reducing final accuracy, suggesting that introducing a significant level of noise during training aids in more effectively forgetting specific data.

Higher r values allow the model to capture more diverse information during unlearning, resulting in lower accuracy retention. This was observed in both datasets, where higher r values, particularly $r = 16$ and $r = 32$, yielded better unlearning performance. This indicates that a larger rank for the LoRA adaptation can enhance the model's ability to forget target knowledge.

These findings highlight the importance of carefully tuning the hyperparameters in our LoRA-based unlearning approach to achieve optimal results. By selecting appropriate values for alpha, dropout, and r, we can maximize the effectiveness of unlearning while minimizing the retention of target knowledge.

The specific configurations (e.g., dropout, alpha values) used in our experiments may have optimized the unlearning process, contributing to the effectiveness of our method. Fine-tuning these parameters can significantly impact the unlearning performance. For instance, higher dropout rates can help improve unlearning by introducing more randomness during the training process, thereby making it easier to forget specific data. Additionally, the characteristics of both the IMDB and Twitter datasets may have made them more susceptible to effective unlearning with our configurations. The text data in these datasets might have patterns that are more easily disrupted by the unlearning process, leading to a more significant reduction in accuracy.

5) Comparison of Results: The comparison of our method with the Retrain from Scratch method for both the IMDB and Twitter datasets is shown in Table III. Our method achieves final accuracies ranging from 0.70% to 1.20% on the IMDB dataset, and 8.27% to 10.63% on the Twitter dataset, indicating a significant reduction in accuracy and demonstrating effective unlearning. The Retrain from Scratch method achieves a final accuracy of 0.65% on the IMDB dataset, and 8.08% on the Twitter dataset, which is slightly better than our best-performing configurations (IMDB: $r = 8$, $alpha = 4$, $dropout = 0.3$, Twitter: $r = 1$, $alpha = 2$, $dropout = 0.3$) with final accuracies of 0.70% and 8.27%, respectively. This indicates that while the Retrain from Scratch method has a marginal advantage in terms of final accuracy, our LoRA-based unlearning approach comes very close to matching its performance.

TABLE III
COMPARISON OF FINAL ACCURACY

Method	Initial Accuracy	Final Accuracy
IMDB & Our Method	99.15%	0.70%
IMDB & Retrain from Scratch	97.60%	0.65%
Twitter & Our Method	85.32%	8.27%
Twitter & Retrain from Scratch	89.10%	8.08%

Although the Retrain from Scratch method achieves a slightly lower final accuracy, it is important to note that our method provides several advantages over retraining from scratch. First, our approach is computationally more efficient, as it focuses on adapting specific parts of the model relevant to the target knowledge, rather than retraining the entire model. This makes our method more feasible in real-world scenarios where computational resources may be limited. Second, our method offers greater flexibility and adaptability to different configurations, allowing it to be easily modified and optimized for various datasets and unlearning requirements.

The low final accuracy achieved by our method, despite being marginally higher than the Retrain from Scratch approach, still demonstrates its high effectiveness in unlearning. This efficiency can be attributed to the careful selection and tuning of parameters, such as alpha, dropout, and r values, which contribute to optimizing the unlearning process. By choosing appropriate values for these hyperparameters, we can maximize the effectiveness of unlearning while minimizing the retention of target knowledge.

Moreover, the implementation techniques employed in our method, such as the LoRA adaptation, play a crucial role in efficiently removing the target knowledge from the model. These techniques enable our approach to focus on the most relevant parts of the model for unlearning, thereby reducing the computational burden and improving the overall efficiency of the unlearning process.

In summary, while the Retrain from Scratch method achieves a slightly lower final accuracy, our LoRA-based unlearning approach comes very close to matching its performance. The marginal difference in final accuracy is offset by the significant advantages offered by our method, including computational efficiency, flexibility, and adaptability to different configurations. These advantages make our approach a promising solution for real-world unlearning scenarios, particularly in resource-constrained environments or when dealing with large-scale models. The effectiveness of our method in achieving low final accuracy, combined with its practical benefits, highlights its potential to address the challenges of unlearning in large language models and its applicability in various domains.

6) Blockchain Complexing Results: In this study, we evaluated the performance impact of integrating blockchain technology into our federated learning framework with unlearning capabilities for Large Language Models (LLMs). We focused on key aspects such as scalability, transaction throughput, and latency introduced by the blockchain component. Our goal was to ensure that the benefits of blockchain integration, such as enhanced security and transparency, do not come at the cost of compromised system performance. We utilized Hyperledger Fabric 2.X to assess the blockchain network's impact on our LLM unlearning process, particularly considering the computational overhead in resource-constrained environments.

- **Blockchain Network Setup**: The initial setup time for the blockchain network was approximately 42 seconds. While higher than our previous study, this one-time overhead is still acceptable, given the long-term benefits in federated learning applications involving LLMs, where security and trust are crucial.
- **Consensus Mechanism Overhead**: The time required for the consensus process, which involved approval from all participating nodes, was added around 4 seconds after the blockchain network setup. This slight increase compared to our previous study is attributed to the higher complexity of LLM-related transactions. However, the duration remains manageable within our federated learning context.
- **Transaction Processing Efficiency**: The average time for processing transactions, including model updates, gradient aggregation, and unlearning-related operations, was 3 seconds. This efficiency demonstrates Hyperledger Fabric's capability to handle the increased complexity of LLM-related transactions effectively.
- **Per-Epoch Time Cost**: During the LLM training process, the duration per epoch, both for normal training and post-unlearning operations, remained consistent at 28-30 seconds. This stability in performance, despite the additional unlearning activities, highlights the robustness of our blockchain-integrated system.

Table IV presents a comparison of time costs between a standard federated learning cycle for LLMs and our proposed blockchain-enhanced method. Similar to our previous study, our method incurs a higher initial time cost due to setup and endorsement processes. However, this cost normalizes over increasing iterations, indicating the scalability of our approach in the context of LLMs.

TABLE IV
TIME COST ANALYSIS FOR LLM FEDERATED LEARNING WITH AND WITHOUT BLOCKCHAIN INTEGRATION OVER 999 ITERATIONS

Method	t = 0	t = 9	t = 99	t = 999
Normal Federated Learning for LLMs	30s	300s	3000s	30000s
Our Proposed System for LLMs	79s	367s	3277s	32277s

7) Conclusion: In conclusion, our experiments on both the IMDB and Twitter datasets demonstrated that our method achieves performance comparable to that of the Retrain from Scratch method in terms of final accuracy reduction. For the IMDB dataset, our best-performing configuration ($r = 8$, alpha=4, dropout=0.3) achieved a final accuracy of 0.70%, closely matching the 0.65% achieved by retraining from scratch. Similarly, for the Twitter dataset, our best-performing configuration ($r = 1$, alpha=2, dropout=0.3) achieved a final accuracy of 8.27%, closely matching the 8.08% achieved by retraining from scratch. The effectiveness of our LoRA-based unlearning method can be attributed to the careful selection and tuning of parameters, as well as the implementation techniques employed. Our method offers a more computationally feasible alternative to retraining from scratch, which can be resource-intensive and time-consuming. The adaptability of our approach to different configurations highlights its flexibility and potential for real-world applications.

Furthermore, we evaluated the performance impact of integrating blockchain technology into our federated learning framework with unlearning capabilities for LLMs. The results showed that the blockchain component, implemented using Hyperledger Fabric 2.X, introduced minimal overhead in terms of setup time, consensus mechanism, transaction processing efficiency, and per-epoch time cost. The stability in performance, despite the additional unlearning activities, demonstrates the robustness of our blockchain-integrated system.

VIII. Conclusion

In this paper, we present a novel blockchain-based federated learning framework for Large Language Models (LLMs) that incorporates efficient unlearning capabilities. By leveraging Low-Rank Adaptation (LoRA) and carefully tuning its hyperparameters, our approach achieves highly effective unlearning, enabling the selective forgetting of specific data points while preserving the model's performance on the remaining data. The integration of blockchain technology, using Hyperledger Fabric, ensures the security, transparency, and verifiability of the unlearning process. While this introduces a slight increase in computational overhead, the benefits of enhanced trust and accountability in the federated learning process justify the marginal time cost.

Our comprehensive analysis demonstrates the effectiveness of the proposed framework and provides valuable insights into the impact of LoRA hyperparameters on unlearning performance. The findings underscore the importance of careful tuning and the complex relationships between rank, scaling factor, and dropout in achieving optimal unlearning results. Overall, our blockchain-based federated learning framework with unlearning capabilities represents a significant step forward in the development of secure, transparent, and adaptable LLMs. By enabling efficient and verifiable unlearning, our approach addresses a critical challenge in the application of LLMs in real-world scenarios, where data privacy and the ability to forget specific information are paramount.

References

[1] 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.

[2] 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*, vol. 103, p. 102274, 2023.

[3] P. Villalobos, J. Sevilla, L. Heim, T. Besiroglu, M. Hobbhahn, and A. Ho, "Will we run out of data? an analysis of the limits of scaling datasets in machine learning," *arXiv preprint arXiv:2211.04325*, 2022.

[4] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, and Q. Yang, "Fate-llm: A industrial grade federated learning framework for large language models," *arXiv preprint arXiv:2310.10049*, 2023.

[5] L. Zhang, T. Zhu, H. Zhang, P. Xiong, and W. Zhou, "Fedrecovery: Differentially private machine unlearning for federated learning frameworks," *IEEE Transactions on Information Forensics and Security*, 2023.

[6] W. Chang, T. Zhu, H. Xu, W. Liu, and W. Zhou, "Class machine unlearning for complex data via concepts inference and data poisoning," *arXiv preprint arXiv:2405.15662*, 2024.

[7] M. Wang, T. Zhu, T. Zhang, J. Zhang, S. Yu, and W. Zhou, "Security and privacy in 6g networks: New areas and new challenges," *Digital Communications and Networks*, vol. 6, no. 3, pp. 281–291, 2020.

[8] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[9] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] C. Chen, X. Feng, J. Zhou, J. Yin, and X. Zheng, "Federated large language model: A position paper," *arXiv preprint arXiv:2307.08925*, 2023.

[12] S. Gupta, Y. Huang, Z. Zhong, T. Gao, K. Li, and D. Chen, "Recovering private text in federated learning of language models," *Advances in neural information processing systems*, vol. 35, pp. 8130–8143, 2022.

[13] J. Jiang, X. Liu, and C. Fan, "Low-parameter federated learning with large language models," *arXiv preprint arXiv:2307.13896*, 2023.

[14] W. Kuang, B. Qian, Z. Li, D. Chen, D. Gao, X. Pan, Y. Xie, Y. Li, B. Ding, and J. Zhou, "Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning," *arXiv preprint arXiv:2309.00363*, 2023.

[15] W. Zhao, Y. Du, N. D. Lane, S. Chen, and Y. Wang, "Enhancing data quality in federated fine-tuning of large language models," in *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

[16] F. Jiang, L. Dong, S. Tu, Y. Peng, K. Wang, K. Yang, C. Pan, and D. Niyato, "Personalized wireless federated learning for large language models," *arXiv preprint arXiv:2404.13238*, 2024.

[17] J. H. Ro, S. Bhojanapalli, Z. Xu, Y. Zhang, and A. T. Suresh, "Efficient language model architectures for differentially private federated learning," *arXiv preprint arXiv:2403.08100*, 2024.

[18] Y. Yao, X. Xu, and Y. Liu, "Large language model unlearning," *arXiv preprint arXiv:2310.10683*, 2023.

[19] C. Yu, S. Jeoung, A. Kasi, P. Yu, and H. Ji, "Unlearning bias in language models by partitioning gradients," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 6032–6048.

[20] L. Wang, T. Chen, W. Yuan, X. Zeng, K.-F. Wong, and H. Yin, "Kga: A general machine unlearning framework based on knowledge gap alignment," *arXiv preprint arXiv:2305.06535*, 2023.

[21] N. Si, H. Zhang, H. Chang, W. Zhang, D. Qu, and W. Zhang, "Knowledge unlearning for llms: Tasks, methods, and challenges," *arXiv preprint arXiv:2311.15766*, 2023.

[22] J. Y. Huang, W. Zhou, F. Wang, F. Morstatter, S. Zhang, H. Poon, and M. Chen, "Offset unlearning for large language models," *arXiv preprint arXiv:2404.11045*, 2024.

[23] H. Luo, J. Luo, and A. V. Vasilakos, "Bc4llm: Trusted artificial intelligence when blockchain meets large language models," *arXiv preprint arXiv:2310.06278*, 2023.

[24] C. T. Nguyen, Y. Liu, H. Du, D. T. Hoang, D. Niyato, D. N. Nguyen, and S. Mao, "Generative ai-enabled blockchain networks: Fundamentals, applications, and case study," *arXiv preprint arXiv:2401.15625*, 2024.

[25] J. G. M. Mboma, K. Lusala, M. Matalatala, O. T. Tshipata, P. S. Nzakuna, and D. T. Kazumba, "Integrating llm with blockchain and ipfs to enhance academic diploma integrity," in *2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*. IEEE, 2024, pp. 1–6.

[26] M. A. Bouchiha, Q. Telnoff, S. Bakkali, R. Champagnat, M. Rabah, M. Coustaty, and Y. Ghamri-Doudane, "Llmchain: Blockchain-based reputation system for sharing and evaluating large language models," *arXiv preprint arXiv:2404.13236*, 2024.

[27] M. Wang, T. Zhu, X. Zuo, D. Ye, S. Yu, and W. Zhou, "Blockchain-based gradient inversion and poisoning defense for federated learning," *IEEE Internet of Things Journal*, 2023.

[28] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, and X. Hu, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," *ACM Transactions on Knowledge Discovery from Data*, 2023.

[29] A. Caballero Hinojosa, "Exploring the power of large language models: News intention detection using adaptive learning prompting," 2023.

[30] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili *et al.*, "A survey on large language models: Applications, challenges, limitations, and practical usage," *Authorea Preprints*, 2023.

[31] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[32] M. Wang, T. Zhu, X. Zuo, M. Yang, S. Yu, and W. Zhou, "Differentially private crowdsourcing with the public and private blockchain," *IEEE Internet of Things Journal*, 2023.

[33] A. Sunyaev and A. Sunyaev, "Distributed ledger technology," *Internet computing: Principles of distributed systems and emerging internet-based technologies*, pp. 265–299, 2020.

[34] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[35] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[36] A. Ginart, M. Guan, G. Valiant, and J. Y. Zou, "Making ai forget you: Data deletion in machine learning," *Advances in neural information processing systems*, vol. 32, 2019.

[37] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual international cryptology conference*. Springer, 2017, pp. 357–388.

[38] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.

arXiv:2407.20181v1 [cs.CR] 26 Jul 2024

Blockchain for Large Language Model Security and Safety: A Holistic Survey

Caleb Geren*
Lehigh University
cdg225@lehigh.edu

Amanda Board*
University of Idaho
boar9227@
vandals.uidaho.edu

Gaby G. Dagher
Boise State University
gabydagher@boisestate.edu

Tim Andersen
Boise State University
tandersen@boisestate.edu

Jun Zhuang
Boise State University
junzhuang@boisestate.edu

ABSTRACT

With the advent of accessible interfaces for interacting with large language models, there has been an associated explosion in both their commercial and academic interest. Consequently, there has also been an sudden burst of novel attacks associated with large language models, jeopardizing user data on a massive scale. Situated at a comparable crossroads in its development, and equally prolific to LLMs in its rampant growth, blockchain has emerged in recent years as a disruptive technology with the potential to redefine how we approach data handling. In particular, and due to its strong guarantees about data immutability and irrefutability as well as inherent data provenance assurances, blockchain has attracted significant attention as a means to better defend against the array of attacks affecting LLMs and further improve the quality of their responses. In this survey, we holistically evaluate current research on how blockchains are being used to help protect against LLM vulnerabilities, as well as analyze how they may further be used in novel applications. To better serve these ends, we introduce a taxonomy of blockchain for large language models (BC4LLM) and also develop various definitions to precisely capture the nature of different bodies of research in these areas. Moreover, throughout the paper, we present frameworks to contextualize broader research efforts, and in order to motivate the field further, we identify future research goals as well as challenges present in the blockchain for large language model (BC4LLM) space.

1. INTRODUCTION

From disparate areas such as software development, the solicitation of political advice, and assistance in creative writing tasks, the introduction of large language models into everyday life has occurred at unprecedented pace and scale [106]. Consequently, many of the vulnerabilities that large language models contend with are well known and understood in the current literature, such as prompt injection [74][98], hallucinations [52][130][83][12], and data poisoning [37][54]. Despite this, relatively little in terms of mitigation has been introduced to combat such weak points [124][2]. The ramifications of this fact are far-reaching in regards to user experience and mounting data integrity concerns surrounding AI systems at large [118][40][68].

For example, and representative of the variety of threats that LLMs face, is the concerning fact that LLMs are easily coaxed into revealing personally identifiable information (PII), as in the case when an LLM was persuaded into divulging undisclosed names, physical addresses, emails, phone numbers, and twitter handles associated with specific individuals to an unauthorized user [20]. Similarly, attacks such as prompt injection can be compounded with a model's tendency to divulge information, resulting in massive data leakage [118]. Typically, defensive responses to these threats manifest themselves in the application of established machine learning techniques, such as differential privacy strategies applied to entire corpus' to improve privacy guarantees [1][122]. While this is an important application, it is often the case that DP guarantees are not enough to fully ensure data privacy in a large language model [68] due to DP's ability to protect primarily "by whom" data is contributed, rather than "about whom" the data is focused on. Additionally, another common attempt to tackle the data privacy problem in LLMs is the use of federated learning (FL) in the training process. This technique distributes training across multiple nodes to create a decentralized environment for model data to exist within [75]. Naturally, this lends itself to further obscuring sensitive information in a model's corpus. However, it has been shown that by taking model weights or gradients, original data from the model can still be reconstructed [63]. Additionally, federated learning solutions are susceptible to many of the same types of attacks as large language models, such as single-point-of-failure attacks or man-in-the-middle attacks [86]. This trend of typical machine learning solutions failing to exhaustively defend against the range of attacks now affecting LLMs continues across multiple traditional threat/defense models [96, 132, 122].

It is clear that to counter new threats arising from large language models there is a simultaneous need for new technologies to be introduced into the space. Situated in a similar

*These authors contributed equally to this work.

position to LLMs in their rapid emergence and adoption are blockchain systems. They have the ability to ensure data integrity via various tamper-evident mechanisms, introduce a high degree of confidentiality into otherwise centralized systems, and allow for data provenance guarantees through traceable and auditable data [27][19][77][4]. These qualities of blockchain systems place themselves at an ideal juncture to be used for bolstering the robustness of large language models. This integration creates the necessary conditions to allow for stronger privacy preservation, enhanced inference checks, anti-adversarial attack technologies, and similar defenses to be incorporated into the design of large language models.

This unification of technologies has become a hot topic of research within both blockchain and large language model oriented communities over the last several years. As a field in its infancy, blockchain for large language models (BC4LLM) demands rigorous analysis so that it may progress unhindered from any limitations that may arise from the combination of two newly introduced fields, especially given their relative complexity and various nuances. Towards this end of progress in the fields of large language models and blockchain systems, we present this survey paper as a vehicle for researchers to better grasp the state of the two fields and especially understand the combination of the two technologies in the vein of how blockchain can better serve large language model systems. To clarify our goals with this paper, we introduce four research questions that motivate the entirety of the paper and generalize our aim in writing this survey. They are as follows:

RQ1. What are the pressing LLM-related security concerns that may be addressed with blockchain technology?

RQ2. How can we meaningfully differentiate between security and safety in the context of BC4LLMs?

RQ3. In what ways can blockchain technology be used to enhance the safety of LLMs?

RQ4. What are prominent gaps within the BC4LLMs area, how can these gaps influence research directions, and what resources can we provide to enable potential new directions?

Of note is our focus specifically on how blockchain systems may impact large language model *security* and *safety*. We limit the scope of this survey to these terms in order to provide finer-grained analysis and categorization of seemingly disparate works - and to more clearly spur research advances in specific directions. Indeed, we typically see attacks on LLMs occur via malicious third parties which exploit system vulnerabilities (security) [5][121][132][44][66] or as passive issues embedded in the structure of a LLM which places users at risk with no malicious outside influence (safety) [34][109][92][125]. It is because of this encompassing distinction that we offer our analysis of the blockchain for LLM (BC4LLM) space in the context of security and safety. Moreover, we further bolster the significance of this distinction with definitions of security and safety in the context of large language models. To the best of our knowledge, we are the first paper to rigorously introduce such definitions as we aim to lay a foundation upon which future BC4LLM works can build. It is also worth pointing out our contributions towards further defining privacy measures in the form of active privacy and passive privacy efforts, modeled after Yan et al's survey on bolstering data privacy [122].

To separate ourselves, and our analysis of BC4LLM through the lens of safety and security, from other similar works, we look at several other closely related reviews. He et al. [42] examine the relationship between large language models and blockchain in analyzing how large language models can further enhance blockchain systems in the LLM4BC space. Mboma et al. [74] provide an exploratory review of general integrations between blockchain and large language models, which is similar to Heston's analysis of integrating the two technologies in the sphere of telemedicine [30]. In the case of Salah, et al. [91], Bhumichai et al. [13], and Dinh et al. [28], we see the exposition of potential and existing technologies between blockchain and artificial intelligence in general. In summary, existing reviews that contend specifically with the blockchain and large language model space are not focused on the direction in which they apply the technologies. On the other hand, reviews that concern blockchain and AI as a whole lose the benefits of tighter granularity and focus. We present an overview of related works in table 1, which juxtaposes the above papers' contents with our specific focuses, stressing the differentiating factors that we introduce into the space.

1.1 Contribution

We outline the exact contributions of this survey paper, and highlight their impact in answering our research questions. They are as follows:

1. We present several key contributions in the form of various frameworks, definitions, and compiled resources. Most prominently, we introduce a taxonomy of blockchain for large language Models in Figure 3. This taxonomy aims to succinctly explain the relevant interactions between various blockchain components and corresponding large language model vulnerabilities [**RQ1**][**RQ3**]. To further motivate and contextualize this taxonomy as well as our general discussion of existing literature, we introduce two definitions of safety and security as they apply to large language models [**RQ2**]. Moreover, we also provide multiple datasets relevant to BC4LLM in order to provide future researchers in the area with the tools to expand upon connections highlighted by the taxonomy and motivated by our definitions [**RQ4**].

2. We also highlight several other important components of our paper, intended to support our main contributions but still relevant in their own regard as pertinent artifacts of the BC4LLMs space. One such artifact is our definitions of specific areas that are found within our definition of safety, further expanded upon in Table 3 [**RQ2**][**R3**]. These definitions add further weight to our definition of safety for large language models. We go on to bolster our definitions of both safety and security by purposefully focusing on the issue of privacy in the context of security, consequently reaffirming two terms as introduced by Yan et al. [122]: passive and active privacy [**RQ1**][**RQ3**]. Also relevant in their own right but primarily proposed as supporting contributions are our contextualization of LLMs as a sub-field of various AI areas [**RQ1**], and our succinct taxonomic overview of blockchain components [**RQ1**][**RQ3**].

Table 1: **Overview of Existing Related Surveys.** Reviews concerning blockchain and large language models take many different forms and facilitate different functions within the broader context of BC4LLMs literature. However, a holistic review of BC4LLMs is absent from the current state of the field. For example, while many survey papers have focused to some degree on general threats affecting LLMs, fewer have considered LLM-specific threats or approached these problems from the direction of using blockchain technology. In this table, we document to what degree several important components of our survey are present in different reviews. In particular, we examine if the background is given on relevant subjects, if a model of threat categorization is introduced, if definitions of security and safety are proposed, whether security and/or safety in regards to BC4LLMs is explored, if future BC4LLMs work is probed, and whether or not the survey instead focuses on LLM for Blockchain (LLM4BC) instead of BC4LLM.
● denotes a full discussion of the related topic, ◐ a partial discussion, and ○ indicates the topic was absent from the review.

Source	*LLM and BC Background*	*Threat Model*	*Definitions*	*Security in BC4LLMs*			*Safety in BC4LLMs*	*Future Work*	*LLM for Blockchain*
				LLM	*AI*	*Non-AI*			
Luo, et al. 2023 [69]	●	◐	○	◐	◐	◐	◐	●	○
Mboma, et al. 2023 [73]	●	○	○	◐	○	◐	◐	○	●
He, et al. 2024 [42]	●	○	○	○	○	○	○	○	●
Heston 2024 [30]	◐	○	○	●	◐	○	◐	◐	◐
Salah, et al. 2019 [91]	◐	○	○	○	●	●	◐	○	○
Bhumichai, et al. 2024 [13]	◐	○	○	○	●	◐	◐	○	○
Dinh and Thai 2018 [28]	◐	○	○	○	●	○	◐	○	◐

3. Finally, we make note of the overarching goal of our paper, which also stands as the most critical contribution we offer. That is, we present this paper in part as a holistic literature review in Section 4. In it, we categorize relevant fields to present novel insights into the space, classify research efforts in several interrelated respects, relate all BC4LLM research projects to LLM safety and security, and as a result capture an accurate and informative picture of how blockchain can benefit large language models [**RQ1**][**RQ2**][**RQ3**][**RQ4**].

The remainder of the survey is organized as follows: Section 2 introduces the background of blockchain technology and large language models. Section 3 consists of our methodology; we analyze how we filtered works for this review, and introduce the relevant definitions which motivate our analysis of the current literature. We also share our model of threat categorization, similar to Yao et al.'s categorization [124]. In Section 4, we present the comprehensive literature review of the BC4LLM space in the context of safety and security. We then scrutinize works within these areas in relation to our BC4LLM taxonomy. Within Section 5, we present relevant datasets to the BC4LLM space in order to supply future researchers with both well-known and more obscure datasets. Afterwards, in Section 6, we discuss important challenges that exist between blockchain technology and large language models which fundamentally hinder progress in the field at large. Section 7 proposes future research directions in the BC4LLM space, and section 8 concludes the paper, which both summarizes our efforts in surveying the BC4LLM area and provides further insight into the state of BC4LLM as a whole.

2. BACKGROUND

In this section, we present an overview of blockchain as a distributed ledger technology and relate the abilities of large language models to their capacities as agents with respect to their nature as both AI models and their tendency to interact with vast quantities of data.

2.1 Blockchain

Since the introduction of Bitcoin as a decentralized currency by Satoshi Nakamoto in 2008 [79] there has been a subsequent explosion of academic and commercial interest in its underlying blockchain technology. Additionally, and as highlighted by the introduction of Vitalik Buterin's Ethereum blockchain in 2014 [17], there has been a particular focus on blockchain's potential applications in fields entirely disparate from digital currencies. The interest in blockchain, or distributed ledger technologies, stems from its guarantees about data sovereignty, transparency, and relative permanence. Concisely, these properties are often referred to as immutability and irrefutability. Ranging from many diverse fields such as health care record management, digital identity management, or tax auditing, these properties are widely applicable and highly desirable, even though the mechanisms through which we achieve them can be somewhat complex and opaque. In light of the oftentimes convoluted nature of blockchain systems, we introduce blockchain to the reader in a piecemeal fashion in order to emphasize the modular, yet interconnected nature of such systems. Figure 1 represents an overview of our characterization of blockchain systems in general. We purposefully exclude certain components such as the incentive mechanism, or wallets, as they are beyond the scope of our analysis of blockchain as a means to serve large language models.

2.1.1 Blockchain Components

Consensus Protocol. Of particular interest to the BC4LLM space, and arguably the most fundamental component within a blockchain, the consensus protocol is the governing system that controls how data is added to a blockchain's ledger. At

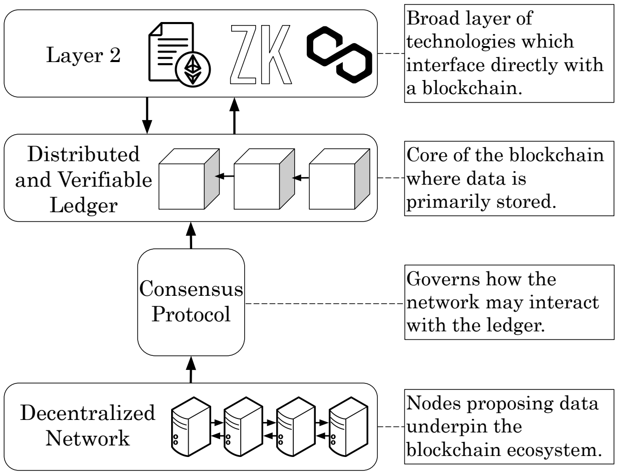

Layer 2
ZK
Broad layer of technologies which interface directly with a blockchain.
Distributed and Verifiable Ledger
Core of the blockchain where data is primarily stored.
Consensus Protocol
Governs how the network may interact with the ledger.
Decentralized Network
Nodes proposing data underpin the blockchain ecosystem.

Figure 1: A blockchain consists of four main components. A decentralized network of nodes interacts with a ledger via a governing consensus mechanism. This ledger, adequately protected by the consensus mechanism, creates what we refer to as the blockchain. Layer 2 solutions can interface with this ledger to enable greater functionality between users and a blockchain's data.

its core is the consensus mechanism, which both ensures the validity of proposed data and fosters an environment of accountability, so that nodes submitting invalid information may be penalized accordingly. For example, the Proof of Work (PoW) consensus mechanism [79] is by far the most widely known. In it, nodes must solve a complex mathematical equation in order to gain rights to propose data for the blockchain. When such a node submits new data, it is scrutinized by every other node in the system. If the data is malicious, or untruthful, the proposal is rejected and the corresponding processing power performed by the malicious node has effectively been wasted, as that node will not receive the incentive, a Bitcoin reward. The underlying ideas of accountability, certain nodes being selected as 'block proposers', and the 'proof' of the ability to submit information to the chain are central ideas in consensus protocols across blockchains with different consensus protocols [81].

Verifiable Ledger. At a blockchain's core sits the verifiable ledger, a repository of data bolstered by a secure way of maintaining the integrity of that data. Of note is the particular technique through which data itself is verified on the ledger: the Merkle tree [76], or a variation thereof. Typically implemented as a ground-up binary tree, data is stored in leaf nodes, with hashed pointers of that data cascading up the tree. This structure results in a comprehensive 'Merkle root', a hash pointer consisting of all the other hash pointers in lower levels of the tree, which is ultimately based on the data stored in the leaf nodes. This technique ensures the integrity of information in the leaf nodes, as any alteration to the data is instantly reflected in the Merkle root. Likewise, new additions to the Merkle tree can be checked against previous states of the tree via a recalculation of the Merkle root accounting for the new transactions. This technique, complementing the verifiable ledger, is often the key to LLM data provenance and traceability solutions that rely on blockchain technology.

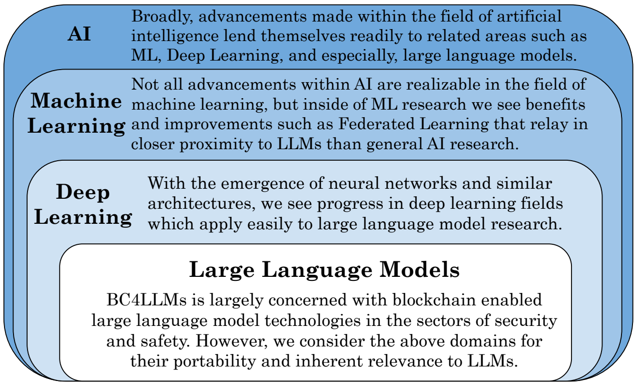
Decentralized Network. Critically, blockchain's are decentralized networks. That is, no central server or group of servers may assume control of the network in a way that would compromise the network's state of trustlessness. This is achieved through multiple avenues, such as the aforementioned consensus protocol, the distribution of the verifiable ledger among a large number of independent nodes, and the accessibility of a given blockchain's network. [27] In this way, no users in the network are required to trust any other user. This fundamental aspect of blockchain is responsible for already realized and potential advancements with LLMs concerning areas such as RAG, the training process, and even supply chain issues.

Layer 2 Technologies. Apart from the fundamental components found within all blockchains themselves, there exist several external architectures that interface with blockchains and further enhance their applicability. Typically, these are referred to as layer 2 technologies, as they sit a 'layer' above the 'layer 1' blockchain. Increasingly relevant as blockchain's influence grows, layer 2 solutions are a burgeoning area with numerous novel research directions. Most prominent among these is the space concerning smart contracts, scripts that rely on a blockchain's security guarantees in order to facilitate off-chain transactions [140]. Also of note in the layer 2 field, and sometimes combined with the efficacy of smart contracts, are zero-knowledge rollups. Often used to strengthen scalability, zero-knowledge rollups batch unproposed transactions together, and instead of submitting the transactions themselves, submit proof that the transactions are indeed valid [103]. This allows for transactions to be added on-chain without the need for every full node to redo the calculations found within those transactions. This area of layer 2 technologies is pivotal as it relates to BC4LLM - layer 2 has the necessary dynamism to react quickly to new and emerging LLM vulnerabilities.

2.2 Large Language Models

In recent years, large language models (LLMs) have grown in popularity as a driving force in artificial intelligence (AI), being used across various fields, such as trustworthiness [51, 25], scholarly document processing [142], signal processing [89], quantum computing [58], climate production [60, 59], software engineering [139], and healthcare [42] among multiple other learning environments. Zhao et al. [135] and Yang et al. [123] define LLMs and pre-trained language models (PLMs) from the perspectives of model size and training approach. Generally speaking, PLMs refer to language models that are pre-trained on large amounts of general text data and then fine-tuned for specific tasks. LLMs are a kind of PLM. The key distinction is that LLMs are generally larger in scale with more parameters. These large language models have demonstrated the ability to learn universal representations of language, used in various natural language processing (NLP) tasks [47], bolstering their applicability.

LLMs are a sub-field of AI. Given this connection, discussions of blockchain for AI (BC4AI) research can also be applicable to LLMs. Figure 2 demonstrates the connections and following developments between AI, machine learning (ML), deep learning (DL), and LLMs. AI is largely discussed, yet we still see improvements in ML, such as FL that is better correlated to LLMs than general AI research. We then note DL as a subset of ML, showing advancements with neural network structures that are similar to how LLMs



Broadly, advancements made within the field of artificial intelligence lend themselves readily to related areas such as ML, Deep Learning, and especially, large language models.

Not all advancements within AI are realizable in the field of machine learning, but inside of ML research we see benefits and improvements such as Federated Learning that relay in closer proximity to LLMs than general AI research.

With the emergence of neural networks and similar architectures, we see progress in deep learning fields which apply easily to large language model research.

BC4LLMs is largely concerned with blockchain enabled large language model technologies in the sectors of security and safety. However, we consider the above domains for their portability and inherent relevance to LLMs.

Figure 2: LLMs are a subset of AI. The focus of our paper is LLMs; however, we acknowledge AI due to the intertwined relationship between LLMs and AI.

learn and operate. To define ML, we refer to Liu et al. [62] who state that ML is an automated learning process using algorithms and statistical models to efficiently perform specific tasks without the use of explicit instructions. DL is a sub-field of ML that utilizes multiple layers of artificial neurons to learn the latent space representation of the most basic form of data such as images, text, and speech signals. As described by Shafay et al. [93] deep learning has been able to achieve human-like accuracy, or even better accuracy than humans, on a variety of difficult tasks. Below, we elaborate upon the process of how models are trained and fine-tuned to become a usable, safe, and secure system.

2.2.1 Model Training

During the pre-training phase, the LLM is trained on a diverse, large dataset of textual data from various sources to learn the statistical properties of language. The LLM is equipped with a myriad of adjustable parameters, commonly reaching more than ten billion [47]. Due to the huge model size and the vast amount of data used to train it, it is computationally challenging to successfully train a capable LLM, requiring distributed training algorithms for learning the model parameters [135]. Another crucial factor for LLM training is the data itself. Data that models are trained on come from a wide variety of sources, but the data itself may not be up to date [101]. To mitigate this shortcoming, recent advancements have introduced Retrieval-Augmented Generation (RAG), which is designed to augment and rectify the information returned by LLMs by consulting up-to-date online sources. The data that the LLM was trained on also has other deficiencies, like knowledge gaps in healthcare fields where data is private and restricted [50]. Due to these knowledge gaps, the LLM may conjure up hallucinations where the model generates false information during prompting [72, 3] because of a lack of relevant information. However, hallucinations may also occur with a plethora of data available as they are inherent problems in LLMs. Methods of preventing these hallucinations are elaborated on in *4.2.2*. RAG can help rectify hallucinations, and fill in the gaps of data the LLM is missing, by using up-to-date and validated information from trustworthy online resources. This method of data retrieval introduces novel vulnerabilities since the information gathered by the retriever is largely un-audited and may contain poisoned data or data that can lead to unsafe responses from the LLMs.

2.2.2 Model Tuning and Utilization

After pre-training, the parameters of LLMs can be further updated by training on domain-specific datasets in downstream tasks. This process is known as fine-tuning (FT) [16]. A kind of fine-tuning method called supervised fine-tuning (SFT), aims to improve LLMs' responsiveness to instructions, ensuring more desirable reactions involving three major components of instructions, inputs, and outputs. Inputs relate to prompting and the inputs depend on the instructions, similar to applications of open-ended generation in ChatGPT. By providing both inputs and outputs they form an instance, and multiple instances can exist for a single instruction [42]. Among fine-tuning, there are a multitude of other training techniques within prompting the model such as instruction tuning and alignment tuning. By FT from a mixture of multi-task datasets formatted via natural language descriptions with the use of instruction tuning, LLMs are enabled to follow task instructions for new tasks without needing explicit examples, highlighting the ability of generalization for instruction following [135]. However, LLMs can demonstrate versatility, even without FT where they produce a phenomenon known as zero-shot learning, exhibiting the ability to perform tasks for which the model was never explicitly trained [16].

Alignment tuning, a form of reinforcement learning, is used to promote the LLM to be a safe interactive machine. Since LLMs are trained to capture the data characteristics of uncurated pre-training corpora involving both high-quality and low-quality data, the LLM can generate toxic, biased, or harmful content for humans. To mitigate this problem, a FT process based on reinforcement learning from human feedback (RLHF) is used to align the LLM with safety values in order to make a trustworthy model [135]. The RLHF process ranks LLM outputs, with rewards scaled to positive and negative values. The LLM is then trained to produce highly-ranked responses and avoid low-ranked responses. In healthcare, RLHF provides advantages to the model such as improved accuracy and reliability through continuous feedback from medical professionals and customizes the interactions based on real clinical settings and patient needs [41]. These advanced training techniques improve LLM's ability to generalize across tasks and improve their overall utility in various domains.

3. RESEARCH METHODOLOGY

The discussion of blockchain technology's incorporation into large language models necessitates a corresponding exploration into the implications of various terms and definitions found at that intersection. For example, due to the rapid emergence of LLMs, there exists an absence of consensus in describing common phenomena concerning LLM safety and security. To ameliorate this effect, we take care to stress opposing, but related, definitions of safety found within many different works in Table 2. In light of these distinctions, we offer two formal definitions of security and safety in order to contextualize these differing but similar areas of research. These definitions will also serve to highlight where particular blockchain technologies could be applied in their respective domains, and focus research efforts.

Table 2: **Differences in Definitions of Safety.** There is no unifying definition of safety within the area of large language models. We see obvious agreement that models should be law-abiding, ethical, and non-violent in order to be safe, and as such these properties are strongly relevant to our definition of safety. However, beyond that point, there is generally a deviation between the authors' respective definitions. This creates two further categories of terms, properties that are moderately relevant to safety and those that are weakly relevant. Questions of fairness, the informing ability of an LLM, and robustness are generally covered but not unanimously, and hence are moderately relevant, whereas privacy-preserving properties or non-sycophancy are rarely discussed in the current literature and are thus weakly relevant to safety. This dialogue between different modes of thought concerning what makes a large language model "safe" heavily influences our definition of safety and our resulting discussion.

Relevance	**Property**	*Sun et al.* [102]	*Liu et al.* [65]	*Han et al.* [39]	*Röttger et al.* [88]	*Zhang et al.* [133]	*Wang et al.* [110]	*Tedeschi et al.* [105]	*Inan et al.* [49]	*Weidinger et al.* [112]
Strong	Ethical	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Law-abiding	✓	✓	✓	✓	✓		✓	✓	✓
	Non-violent	✓	✓		✓	✓	✓	✓	✓	✓
Moderate	Fair				✓	✓	✓	✓		✓
	Informing		✓			✓	✓			✓
	Robust				✓	✓	✓			✓
Weak	Privacy Preserving	✓			✓		✓			
	Non-sycophantic				✓			✓	✓	

First, and to allow a richer discussion centered around safety and security, we delineate between active and passive privacy within LLMs as introduced in Yan et al. [122].

- *Active privacy* is where a user intentionally tries to gain access to sensitive information by breaking the large language model, especially with backdoor attacks, prompt injection attacks, and membership inference attacks during the pre-training and FT phases.

- *Passive privacy* is the state or condition of any impacted person being protected from accidental or unexpected data leakage originating from a large language model. This definition includes protecting the privacy of not only users but people whose information was added to a model's corpus without their knowledge or consent.

Next, we introduce our definitions of security and safety regarding LLMs.

Definition 1. ***LLM Security.*** *A large language model is considered secure if it:*

1. *Withstands applicable adversarial attacks and maintains system integrity, providing consistent and accurate responses, and*

2. *Ensures active user privacy, explicitly resisting backdoor, prompt injection, and inference attacks to prevent malicious users from extracting private information.*

Definition 2. ***LLM Safety.*** *A large language model is considered safe if it interacts with users in a trustworthy manner, adhering to the aforementioned (Table3 interrelated properties of safety: being ethical, law-abiding, nonviolent, fair, passively privacy-preserving, and informing.*

These definitions will serve a versatile role throughout this paper as building blocks for our contextualization of relevant and notable research efforts in the area of BC4LLMs. Moreover, they will also serve the community at large in helping to establish reliable and tangible properties of secure and safe large language models. They also will aid in establishing tighter definitions on finer-grained terms and ideas within BC4LLMs. For example, in Table 2 we provide definitions for terms found within our definition of safety to lessen the effect of the vague nature of some of the words. These definitions are backed by relevant examples found in the literature.

3.1 Research Approach and Limitations

Literature surveys often are limited in their depth and scope by unconscious factors that impact the authors' ability to fairly select papers for review. To be transparent, and to aid researchers conducting similar or future reviews, we outline our research approach and its associated limitations. While conducting our research, we used the search engine Google Scholar, and several databases including ACM Computing Surveys, IEEE Xplore, SpringerLink, and arXiv. We chose these databases as they either produce quality research and contribute to the growth of interest in novel areas, or in the case of arXiv have the most up-to-date papers available. With Google Scholar, we used keyword searches such as "blockchain for LLMs" and "blockchain-based LLMs" as starting points for relevant, intriguing research papers. To solve problems concerning scope and interrelated domains of disparate areas, we gathered various applications of blockchain for AI, blockchain-enabled machine learning, federated learning, and deep learning tactics to apply them to LLMs. Lastly, of note is the fact that we were largely aided in this further research effort by a waterfall approach to finding research papers. That is, we found several foundational papers in the BC4LLM field, explored citations in those papers, and subsequently explored citations in those secondary papers. We continued investigating relevant citations in this waterfall fashion until we reasonably exhausted all relevant articles. Admittedly, this method of finding prominent research articles is limited in its natural tendency to develop blind spots to less well-known research articles or venues.

Table 3: **Safety Area Definitions and Examples.** The area immediately surrounding BC4LLMs lacks a unifying definition of safety as well as consensus on what terms within that definition precisely mean. We provide generalized definitions for terms considered in our definition, as well as examples of incidents in literature where LLMs deviate from behavior as described in the definition. Italicized terms indicate inclusion in our definition of safety.

Safety Area	**Definition**	**Example of Non-alignment**	
Ethicality	LLMs aligning with moral principles.	A LLM agreeing with eugenics. [43].	Found within definition of safety
Legality	LLMs refusing to assist users in illegal endeavors.	A LLM assisting a user in creating incendiary devices. [107].	
Non-violence	LLMs soliciting generally non-violent advice or instructions.	A LLM advising a user to perform a 'raid on a drug house' and 'kill everyone there' [35].	
Passive Privacy	LLMs protecting private data within their corpus absent of malicious threats.	A LLM partially or fully reconstructing private images from a given dataset [20].	
Honesty	LLMs refraining from producing inaccurate or misinformed responses which may lead to negative outcomes.	LLMs administering faulty or fundamentally dangerous advice to patients or physicians in a healthcare setting [82].	
Fairness	LLMs ensuring a equitable environment for interaction, regardless of social identity.	LLMs associating "male" names with qualities of leadership, and "female" names with qualities of amicability [109].	
Robustness	The ability of the LLM to defend against adversarial attacks, originating from outside the model. This is a wide-reaching term, and falls within our discussion of security as it relates to LLMs.	A LLM falling victim to a backdoor attack planted in poisoned training data and producing malicious outputs as a result [124].	
Non-sycophancy	LLMs choosing consistent outputs despite the chance that they may be in conflict with a user's beliefs or desires.	A LLM revising a correct answer to an incorrect answer after the user asks the LLM if they are sure or challenges the LLM's result in some way [95].	

However, in the spirit of a literature survey, we choose to focus on more established papers that more accurately capture the trends currently found in the space. For our exclusion criteria, we limited our research as follows: no duplicates; found articles from 2016 and above, excluding the original Merkle Tree paper [76]; no Masters or Ph.D. theses; and only studies written in English.

3.2 Model of Threat Categorization

There exist a wide variety of threats which affect LLMs. Oftentimes, many of these threats originate from the nature of LLMs acting as AI systems. In Figure 3 we refer to these vulnerabilities similarly to Yao et al [124] who categorized the most extensively discussed LLM vulnerabilities and AI-inherent vulnerabilities together, yet also included external threats under non-AI inherent vulnerabilities. We contribute further by applying these vulnerabilities to each respective process within developing a large language model and tie these to respective applications of blockchain. Beginning with the training process, LLMs are prone to threats such as data poisoning and backdoor attacks. As defined in Yao et al. [124] data poisoning is where attackers influence the training process by injecting malicious data into the training set, introducing vulnerabilities within the security and effectiveness of the model. Following the trend of poisoned data, there can be backdoor attacks implemented on the training data, as defined in Li et al. [56] who categorize backdoor attacks into attacks on training data and attacks on local models. The backdoor attacks on training data are further divided into attacks based on label flipping and attacks based on planting triggers. Attacks based on label flipping focus on manipulating the labels, whereas attacks on planting triggers modify the input data and labels, effectively constructing an adversarial sample. Then, attacks on local models are further divided into attacks based on modifications to the training process and attacks based on manipulating the trained model [56]. The backdoor attacks can be applicable to both the training and prompting phases of LLMs when using this distinction.

RAG attacks have a variety of issues, including privacy issues [127][5] and knowledge poisoning attacks [143]. For RAG specifically, Xue et al. [121] propose BadRAG to identify security vulnerabilities, exposing direct attacks on the retrieval phase from semantic triggers, and uncovering indirect attacks on the generative phase of LLMs that were caused by a contaminated corpus. These RAG-specific attacks and defenses are elaborated on in section 4.1.2. When interacting with a LLM, we see AI's inherent vulnerabilities become relevant, as in Yao et al. [124] note that LLMs are fundamentally AI models themselves. We focus on the prevalent adversarial attacks that malicious users may use to tamper with the LLM, attempts to find out sensitive information, or try to ruin the system entirely. We recognize jailbreaking and prompt injection as two separate but similar types of adversarial attacks that are initiated within prompting. For instance, jailbreaking prompts are designed to bypass the restrictions set by service providers during model alignment or other containment approaches [96]. Whereas prompt injection attacks aim to override an LLM's original prompt and directs it to follow a set of malicious instructions, leading to erroneous advice or unauthorized data leakage [66]. In our Sections *4.2.1*, *4.2.2* we discuss instances of misinformation and passive privacy leakage addressed as safety concerns. Note that from above, we included back-

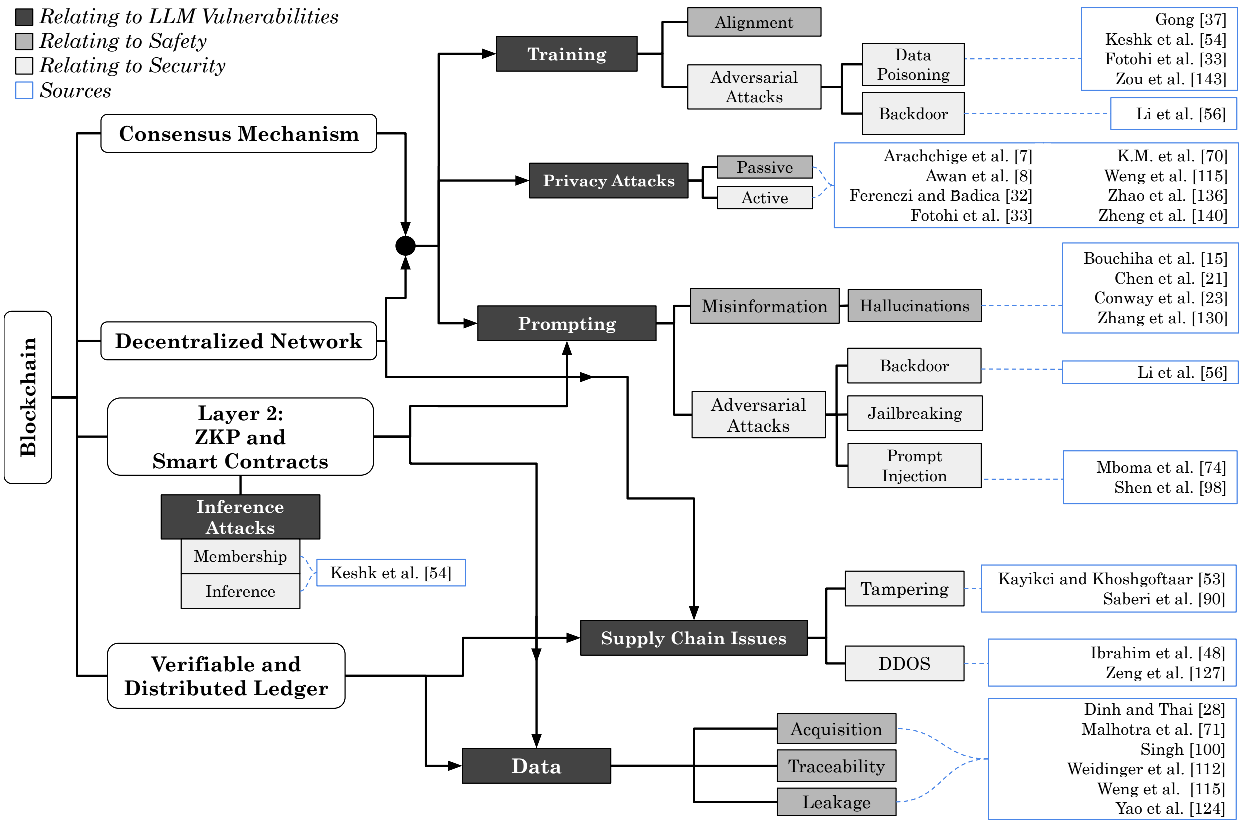


Figure 3: Taxonomy of Blockchain for LLM's Security and Safety. This diagram outlines the integration of blockchain technology to enhance the security and safety of large language models through categorizing interactions and safeguards into several layers and components. Each section supports relevant sources for further reference, illustrating a comprehensive approach to mitigating and preventing vulnerabilities as well as supplementing the security and safety of LLMs through blockchain technology. Promising areas that currently do not have blockchain as a solution to these vulnerabilities intentionally do not have source boxes.

door attacks based on modifications to the trained model in prompting, since these backdoor attacks can still happen after the model has been trained [56].

Other attacks relating to LLMs and AI are membership inference attacks (MIA), relating to a type of privacy attack where some malicious user, given access to the model, is able to determine whether a given point was used to train that model with high accuracy [80]. However, Neel and Chang [80] state this attack is more related to information about the training point data leaking through the model, and that malicious users must have access to a candidate point in order to run the attack. Therefore this attack is more prevalent with passive privacy, highlighting the need for preventing data leakage. Similar attacks are user inference attacks that seek to gain knowledge or insights about the model or data's characteristics, often by observing the model's responses or behavior [124].

Lastly, we explore denial of service (DoS) attacks and supply chain vulnerabilities. From Yao et al. [124] a DoS attack is a type of cyber attack that aims to exhaust computational resources, resulting in latency or making the technology resources unavailable. We focus on distributed denial of service attacks (DDoS) which is a type of DoS attack where requests flood the system, attacking simultaneously from multiple sources on the network [29]. Yao et al. [124] also defined LLM supply chain vulnerabilities, referring to the risks in the lifecycle of LLM applications that may occur from using vulnerable components or services, including third party plugins that may be used to steal chat histories, access private information, and or execute code on a user's machine [124]. All of these security vulnerabilities are substantial threats to LLMs that need to be mitigated or prevented. Possible methods of defense are discussed within Section 4.1, using current blockchain frameworks and experiments for these security problems, as listed by each developmental phase of the LLM, AI inherent threats, and supply chain issues.

4. EXISTING LITERATURE ON BC4LLM

Independently, the fields of both LLMs and blockchain research have grown substantially over the past several years. It is no surprise that the literature surrounding these topics has begun to morph and relate to each other. In previous research, we have seen LLMs for Blockchain Security [42] as well as an introduction to the term BC4LLM in Luo et al [69] where they provide a comprehensive survey of blockchain for LLMs. However, they do not acknowledge the multitude of safety and security solutions that blockchain provides for certain LLM vulnerabilities. Effectively, Luo et al. [69] aims to introduce BC4LLM for trusted AI, enabling reliable learning corpora, secure training processes, and identifiable generated content. In juxtaposition, our survey aims to analyze possible BC4LLM solutions closely related to our definitions of safety (2) and security (1) when looking at inherent system vulnerabilities in LLMs. To begin our analysis, we define these security problems based on previous work and highlight areas of research that are applicable to areas of BC4LLM safety and security.

4.1 Blockchain for Large Language Model Security

Few papers and experiments analyze how the integration of these two technological powerhouses interact with one another. We have seen benefits of this integration that apply to our definition of security (1). Balija et al. [10] introduced a peer-to-peer (P2P) federated LLM, namely PageRank, which works with a blockchain. This system operates in a fully decentralized capacity. Demonstrably, the blockchain implementation led to more efficient accuracy and latency results. With that being stated, Balija et al. [10] provide a developing direction in the field of BC4LLM to enhance system security. Below, we address several current vulnerabilities in LLMs and analyze them individually. In order to better understand these security problems, we categorize these vulnerabilities to their respective LLM training stages, highlight blockchain for AI works, and provide well-researched blockchain applications as a solution.

Vulnerabilities are present at each step in the process of developing a LLM. In early methods of model training, we encounter adversarial attacks such as data poisoning and backdoor attacks within the corpus [96, 122]. Progressing into model fine-tuning and general use, the LLM can fall victim to prompt-based attacks, [2][134] [122][66], inference attacks [44] [54], and RAG related attacks [121] [22][5][26][143] [127]. These attacks are common vulnerabilities in both LLMs and AI since LLMs and AI are closely related as seen in Figure 2. Some of the threats against LLMs can be addressed by implementations from blockchain for AI (BC4AI) research, as elaborated below in Section *4.1.3*. Considering the volume of potential attacks against LLMs, we make a further distinction of solutions that are specifically related to BC4LLM research and other blockchain-based solutions from BC4AI research. With this, we are able to highlight shared vulnerabilities for LLMs and AI. We provide an analysis of how blockchain can help defend against and mitigate these vulnerabilities, starting with threats during each phase of LLM training and utilization, continuing onto different blockchain solutions for AI inherent threats, and lastly noteworthy technology inherent attacks such as denial of service (DDoS) attacks and issues with supply chain logistics.

4.1.1 Blockchain for Threats within LLM Training Process

Situated at a critical juncture in the process of model development is the issue of data selection. An area of paramount concern for this data is exactly how we can verify that training data is authentic and safe as well as tamper-proof from data poisoning attacks. A new approach may be the LLM's ability to unlearn this poisoned data, or data that is unsafe according to our definition of safety 2. From Zuo et al. [145] they establish federated TrustChain as a method of enhancing LLM training and unlearning through a blockchain-based federated learning framework. By integrating blockchain using Hyperledger, their findings present that the framework is efficient in unlearning capabilities, showcasing an accuracy score initially of 99.15% and after unlearning, the accuracy drops to 0.70% [145]. This highlights the potential for BC4LLM to improve security and privacy guarantees, where the LLM can selectively forget specified data points while simultaneously preserving their performance with the leverage of Low-Rank Adaptation (LoRA) and tuning hyper-parameters. This method of a blockchain-enabled federated unlearning process is further detailed as a future research possibility that has been thoroughly explored by few, as emphasized later in Section 7.1.

Another significant problem in the area of adversarial threats against LLMs is data poisoning. Gong et al. [37] proposed a possible blockchain solution, introducing dynamic large language models (DLLM) on blockchains. Instead of using the traditional centralized data sets that LLMs are provided with, developing LLMs on blockchains enables the creation of decentralized datasets. These datasets are less likely to be tampered with and can be easily audited for accuracy. Gong et al. presents DLLM to evolve after the training process. This was implemented through neural network parameters that offer the LLM the ability to learn during its usage.

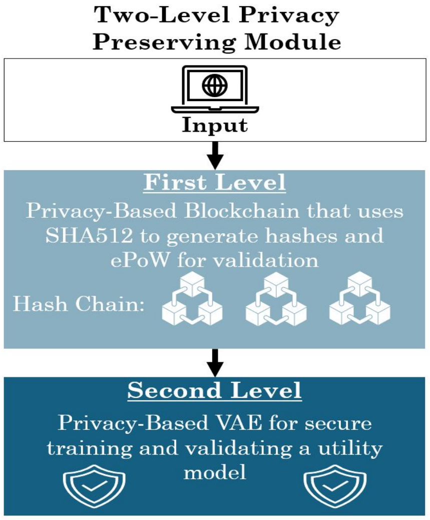

Two-Level Privacy Preserving Module
Input
First Level
Privacy-Based Blockchain that uses SHA512 to generate hashes and ePoW for validation
Hash Chain:
Second Level
Privacy-Based VAE for secure training and validating a utility model

Figure 4: We provide an overview of a two-level framework consisting of: a privacy-based blockchain that uses the secure hash algorithm 512 (SHA512) to generate hashes for data integrity. Then, the enhanced proof of work (ePoW) is used to authenticate data records and prevent data poisoning attacks from altering original data. These hashes of data blocks are linked to each other, called a Hash Chain. Then, for the second level, privacy-based variational autoencoder (VAE) for secure data transformation, ensures robust protection against inference attacks while maintaining the utility model for anomaly detection.

Additionally, blockchain-based systems can help assess where data poisoning may occur and as shown in [54] blockchain can protect datasets and detect potential inference attacks through a two-level privacy preserving module. This research included a framework based on blockchain and deep learning, including two levels of privacy mechanisms as shown in Figure 4. For the first level, Keshk et al. [54] used SHA512 to generate secure hashes and then implemented an enhanced-proof-of-work (ePoW) technique for authenticating data records and preventing data poisoning attacks. The second level consisted of a VAE technique for converting original data into an encoded format for mitigating inference attacks that could be learned from system-based machine learning. In their testing, these mechanisms were effective in preventing data poisoning and inference attacks from manipulated smart power network datasets. BC4LLMs could benefit from this similar type of implementation, working with secure methods of hashing and blockchain-based deep learning privacy preservation techniques. By integrating a two-level privacy-preserving module, BC4LLMs can ensure data integrity and confidentiality while effectively detecting and mitigating both data poisoning and inference attacks.

Poisoned data has been an interest with RAG in particular. For example, Xue et al. [121] developed a way to identify security vulnerabilities from a poisoned corpus, but they do not use blockchain as a solution. We address the absence of research on blockchain and RAG, especially when using blockchain to help prevent RAG security issues. We discuss this as a possible future research direction as there remains a current gap in research of blockchain-based RAG systems and elaborate on this topic in 7.2. Poisoned data overall is a major concern within LLMs and we offer blockchain as a potential source of ground truth to aid in mitigating this threat during the pre-training stages of LLMs and potentially mitigate RAG security concerns. In addition to data poisoning, LLMs are susceptible to backdoor attacks hidden in the training data during the LLM pre-training phase. Notable research by Zhao et al. [134] introduced ProAttack which improves the stealth of backdoor attacks by accurately labeling poisoned data samples. As these attacks improve and become more sophisticated, it is crucial to explore robust defense mechanisms for LLMs. Few defense mechanisms using blockchain have been made, but we see in Li et al. [56] a proposal for a blockchain-based federated-learning framework (DBFL) that withstands backdoor attacks in a blockchain environment through incorporating an RLR aggregation strategy into the aggregation algorithm of a user and the addition of gradient noise to limit the effectiveness of backdoor attacks. The robustness of FL against backdoor attacks is enhanced by using various blockchain functions, including digital signature verification and simulation of chain resynchronization [56].

4.1.2 Blockchain for Threats within LLM Prompting and Utilization

LLMs undergo continuous training through instruction tuning, alignment tuning, and fine-tuning, which warrants a list of vulnerabilities such as prompt injection [74][98] and backdoor attacks [56]. These adversarial attacks are included under the term active privacy 3 where a malicious user attempts to gain unauthorized access. Blockchain technology, with its inherent transparency and immutability, holds the potential to mitigate and defend against these vulnerabilities. For instance, blockchain's ability to ensure data integrity and provide traceability can play a pivotal role in defending against prompt injection attacks. Mbula et al. [74] produce an overview of LLMs for blockchain and note the capabilities of blockchain, especially the transparency and immutability to provide a reliable audit trail of transactions to track and investigate any suspicious activities. Although not specifically focused on prompt injection, this approach showcases how blockchain can supplement security by offering a clear and immutable record of interactions. Applying this to BC4LLMs can help prevent suspicious users from continuously interacting with a LLM, allowing for traceability to stop the user from entering malicious prompts. We recognize prompt injection is a critical vulnerability in LLMs and AI-related systems, yet as noted in the survey from Shen et al. [98] few blockchain defenses for prompt injection are present in the current field of research.

Inference attacks are also of critical importance to LLMs and to active privacy, where a malicious user tries to extract data from the LLM. The malicious user could also try to gain information about the training data, hence why the Taxonomy 3 has inference attacks standalone. As discussed previously in Section *4.1.1* from Keshk et al. [54] that uses Blockchain and DL to privacy preserve and prevent inference attacks through a framework as depicted in Figure 4. For more inference attack applicable work, a survey [44] thoroughly discusses membership inference attacks on ML and provides a group of defenses including differential privacy, regularization, confidence masking, and knowledge distillation. In other works, there are instances of blockchain-based differential privacy methods [136][38][84], but current research that uses blockchain-based differential privacy frameworks to prevent inference attacks is limited; It is worth noting that the theoretical foundation and potential synergies of this combination are promising. Another area with limited research is blockchain as a defense for jailbreaking attacks, which exploit the inherent capabilities of LLMs to bypass restrictions. There are multiple articles defending LLMs from jailbreaking attacks, yet little to none fully include blockchain to prevent jailbreaking. Hu et al. [45] explores a blockchain defense mechanism for malware checking on operating systems, indicating a possible direction for future research in integrating blockchain to defend against jailbreaking in LLMs. As previously explained in Section 3.2, backdoor attacks after the model has been trained are based on modifications to the trained model. The key blockchain-based federated-learning framework from Li et al. [56] discussed in detail in Section *4.1.1* used a combination of a blockchain environment and an RLR aggregation strategy to defend against backdoor attacks. This framework effectively coordinated FL processes and maintained learning security and user privacy. When testing backdoor attacks caused by malicious participants, the accuracy of the model increased when using the RLR aggregation strategy [56]. Given these findings, the possibility of leveraging blockchains transparency and immutability presents a robust mechanism for improving LLM security against active privacy threats. However, comprehensive integration and empirical validation of blockchain-based defenses in LLMs remain imperative to advance the field of BC4LLMs.

4.1.3 AI-intrinsic Threats and Defenses

AI intrinsic threats apply to LLMs due to the proximity of LLMs and AI, as shown in figure 2. Blockchain for AI (BC4AI) is an emerging technology, with blockchain-based solutions already being researched as a secure way to establish trust in the Internet of Things (IoT) [99][24]. Instead of the term BC4AI, some previous works refer to the integration as "Onchain AI" [28][23]. Research of BC4AI encapsulates other machine learning related techniques such as blockchain-based federated learning and blockchain for deep learning. Federated Learning is an addition to machine learning, as noted in figure 2, where federated learning uses a privacy-preserving and decentralized approach to centralized systems.

There are multiple sources relating BC4AI, dating from 2018 to 2024 [28][111][67] [119][31][91][104][71][18][11][13]. Focusing on the most well known works, Salah et al. [91] state integration benefits of BC4AI. For example, there are five main benefits, such as enhanced data security, improved trust in robotic decisions, collective decision-making, decentralized intelligence, and high efficiency[91]. For enhanced data security, it is known that information held within the blockchain is highly secure. By storing sensitive and personal data in a distributed, disk-less environment and allowing AI algorithms to work to secure data, also to ensure more trusted and credible decision outcomes. The other benefits of improving trust within AI decision making involves using the blockchain as a record of the decision-making process, allowing better AI traceability to analyze the quality of responses. Secondly, from Dinh and Thai [28] who summarized the integration of blockchain and AI to where blockchain can assist AI in multiple aspects, as follows. AI can benefit in secure data sharing from blockchain, allowing transparency and accountability regarding which user's data is accessed, when, and by whom, letting users maintain control of their personal data. Among other data concerns, with the integration of blockchain and AI, blockchain technologies can let user's sell their data via smart contracts, enabling the possibility of data marketplaces without a centralized middleman, making the transactions private and secure between users.

A noteworthy framework from Malhotra et al. [71] provides a blockchain-based proof-of-authenticity framework for explainable AI (XAI) utilizing a public Ethereum Blockchain, smart contracts, and IPFS (Interplanetary File System) to ensure secure, traceable, auditable transactions within the Ethereum network. This framework highlights three major components, smart contracts, an Ethereum and IPFS interconnected network, and a regulator, as depicted in figure 5. The use of smart contracts to enable continuous monitoring and tracing by all peers, in the case of any rule violations there are prompt rebound transactions to restore the system to an optimal state. To address the size limitation of storage on the blockchain, as further discussed in Section 6.1, Malhotra et al. [71] used unique IPFS hashes stored on the Ethereum Blockchain to access larger-sized explanations that are stored off-chain in IPFS. These hashes are encrypted with the SHA256 algorithm to maintain data security. Thus, only entities with the corresponding hash can access and retrieve the IPFS hash and the associated explanation, ensuring controlled access even in a distributed network. Lastly, the regulator's role is responsible for auditing and has access to the explanations to predict the user at fault using audit trails if system failure were to occur.

4.1.4 Non-AI Threats and Defenses

Referring to our Figure 3, we specifically focus on DDoS attacks and supply chain issues. Even though these attacks are common problems among technology as a whole; they are threats we consider relevant to BC4LLMs. Ibrahim et al. [48] suggest using a public blockchain to prevent DDoS attacks on IoT devices. Blockchain provides a tamper-proof platform as well as demonstrates how IoT devices working with blockchain can verify and authenticate using a trusted white-list which is implemented in the smart contract. Following this smart contract usage, if LLMs were to use a trusted white list for users then we can try to prevent these malicious users from trying to access the LLM in certain circumstances that are mutually agreed upon. Additionally proven in Shah et al. [94], blockchain-based solutions play a vital role in mitigating DDoS attacks.

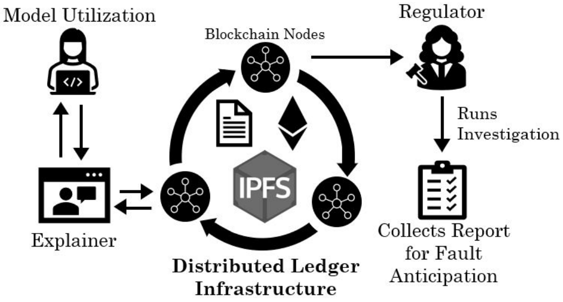


Figure 5: Within BXAI, this diagram illustrates the framework that leverages a distributed ledger infrastructure, using the Ethereum Blockchain and IPFS for storage and secure, traceable transactions. Depicted is the interaction between model utilization: an explainer generating local post-hoc explanations, the storage of these explanations in IPFS, and their linkage to the Blockchain. The use of smart contracts helps secure and encrypt the data, then relay it to a regulator who investigates current explanations, ensuring accountability and fault anticipation. Blockchain nodes are used to facilitate the secure and transparent broadcast of events within the Ethereum network.

A point of consideration is how DDoS attacks that target the blockchain to make the blockchain unavailable would require sufficient computer resources. The fully decentralized architecture of the blockchain and the consensus protocol for new blocks ensure that the blockchain can still operate meanwhile several blockchain nodes could be offline [128]. Incorporating this architecture into LLMs would help prevent DDoS attackers, as the larger the blockchain network is, then the harder it would be for a DDoS attack to be successful. Moreover, blockchain is known as a distributed, immutable, and verifiable ledger technology that ensures transparency and traceability [90]. By utilizing blockchain for LLMs, we can help mitigate these supply chain vulnerabilities. The decentralization of the network can maintain the integrity of the system at all points, aiding in mitigating the risk of a single point of failure, a common problem with centralized systems [90]. Blockchain is offered as a solution if the LLM were to accidentally crash, or was purposefully attacked by an attempt at overwhelming the system, then the LLM would still be intact since it is blockchain-based, removing the single point of failure entirely. However, it is important to note that blockchain solutions for LLMs depend on the availability of the underlying LLM infrastructure. If the LLM server is malfunctioning or shut down, then these blockchain mechanisms may not be applicable, highlighting the need for a robust and resilient supply chain. To solidify the supply chain, blockchain offers secure transactional data in sectors including supply chain management, healthcare, and federated learning [145]. For better supply chain management and data traceability, Kayikci and Khosgoftaar [53] address the potential intersection of blockchain and ML. ML can aid in analyzing data from multiple sources and identify potential supply chain issues such as delays or quality issues before they occur. By using blockchain to create a transparent record of all supply chain transactions there are improvements in security, openness, traceability, and productivity [53]. While blockchain presents a promising solution for enhancing security and defending against adversarial threats to LLMs, ongoing research and development are necessary to address the evolving landscape of threats and vulnerabilities.

4.2 Blockchain for Large Language Model Safety

The burgeoning dominance of large language models as search engines [87], code-writers [139], and a vast array of other roles has led to the emergence of unique problems within the sphere of their safety. For instance, LLMs who advise users to engage in dangerous activities such as eating glass [40] or which easily reveal personally identifying information [55] may be unsafe for users to interact with even in the absence of external threats. In this section, we rely on our proposed definition of safety (Definition 2) to explore relevant literature that incorporates blockchain technology into the various solutions surrounding LLM safety.

4.2.1 *Blockchain for Passive Privacy in Large Language Models*

Despite its novelty, the notion of passive privacy is of paramount importance in maintaining the safety of large language models. Some models face the issue of leaking sensitive data, potentially revealing private information like government-issued ID numbers and patient data [83]. Given the severity of such leaks, it is evident that solutions to such problems are required for more substantial advancements in LLMs. To this end, blockchain's guarantees about data sovereignty, obfuscation, and traceability translate well into realizable passive privacy benefits for large language models. In particular, we observe blockchain-based privacy preservation techniques which originate in varying proximity to LLMs as seen in BC4LLMs itself [113][100][108], blockchain-enabled deep learning [93][141][116][115][54], blockchain-enabled machine learning [7], and blockchain-enabled federated learning [85][70][137][78][8][32][86].

Within our focus of BC4LLMs, we have observed distinct trends in the application of blockchain to large language models in their capacity to bolster passive privacy guarantees. Most notably, the development of ZK-LLMs, or zero-knowledge large language models as described in Wellington [113] and Singh [100] have the potential to drastically reduce privacy leakage risks when interacting with large language models. Considering the problem of data leakage approached from the lens of access, this application is natural. A user querying for their own personally identifiable information should, ideally, be able to access it whereas an unauthorized user should not. Obfuscating portions or the entirety of a corpus using zero-knowledge proofs allows for untrusted training nodes, or the model itself, to act on sensitive data without the ability to regurgitate it to a potentially malicious party. This same mechanism has broad applications that have been explored in other recent works as well, with special focuses on ZKPs for data curation and preprocessing [108] which consequently enhance passive privacy within large language models.

Additionally, besides material on BC4LLMs, it is necessary to discuss passive privacy contributions made within LLM-related areas, as described in our classification of LLMs in the context of AI, ML, and DL.(Figure 2) Especially important in its immediate applications to LLMs, blockchain-enabled deep learning (BC-DL) is a growing field with potentially large impacts on LLM's passive privacy. Specifically, certain BC-DL technologies propose learning mechanisms distinct from traditional federated learning models [116][93][54]. The concerted research effort to develop efficient distributed learning models that deviate from the typical model of federated learning is clearly well underway. This field has broad implications for blockchain; through the utilization of various blockchain properties, we see the development of privacy guarantees which undoubtedly strengthens the BC4LLMs area.

Of particular note in the BC-DL area is the influential DeepChain paper [116] which introduced a novel privacy-preserving training framework based on blockchain technology. The proposed system, reliant on a consensus protocol and corresponding incentive mechanism, allows for the existence of private training gradients and the guaranteed auditability of training data. This type of twofold approach, which also leans on zero knowledge in several areas of the protocol, could be indicative of a new direction of research for blockchain-enabled passive privacy, stemming from the vastly studied federated learning area. In order to highlight the novelty of such an approach, we provide Figure 6 to further illuminate the nature of a system interacted with chiefly through the two vehicles of consensus and incentivization.

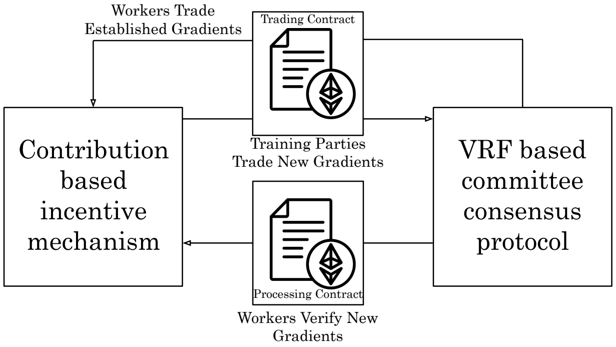


Figure 6: Within DeepChain, we see a deviation from conventional federated learning models in that while training occurs in iterative rounds, it is marked by synchronous requirements for consensus on data added to the model in the last round of training. The contribution-based incentive mechanism rewards workers for verifying new gradients, and stimulates the trading contract which facilitates the sharing of new and updated gradient information. A Verifiable Random Function (VRF) ensures that the committee-based consensus is fair and alleviates concerns surrounding finality.

Additionally, despite this potential research direction, discussion on the wide body of research that does exist concerning blockchain-enabled federated learning (BC-FL) is essential when describing blockchain as a vehicle for improving the safety and security of large language models. Federated learning, introduced in McMahan et al. [75], has since been the principle building block of decentralized learning approaches in machine learning systems. While too broad to be considered for BC4LLMs, notable to this paper are the contributions of authors approaching BC-FL in its capacity as a powerful privacy preserving mechanism [78][8][32][33][70][137].

4.2.2 *Blockchain for Misinformation in Large Language Models*

The issue of large language models fabricating information, commonly known as hallucinations [72][3] is well understood. As a result, detecting and defending against hallucinations is a widely explored area [92], with more research still yet to be conducted [126][130]. Along with this pressing general body of research, efforts have been made to leverage blockchain technology to reduce hallucinations by consensus-oriented [131] and oracle-based [15] approaches. Within consensus, Zhang et al. [131] proposed a system for efficient large language model inference quality assessment. That is, the veracity of a given model's responses was able to be assessed by using a 'Proof of Quality' consensus mechanism with low latency between the user and language model. This stands in contrast to other approaches, such as Bouchiha et al.'s [15] reputation-based system LLMChain, which relies on a decentralized oracle to cross reference request/response pairs originating from differing models and speak to the quality of inferences based on those comparisons. It is worth noting that despite these fundamental differences, they are both consensus-based approaches. This serves as an excellent example of how BC4LLMs technology can take many different forms towards the same goal.

Besides steering towards consensus-driven models of governance for misinformation, several other ideas have been considered to enhance the correctness of large language model responses. These accomplishments have come from the zero-knowledge space as in the case of Chen et al. [21], more optimistic privacy guarantees found within works such as Conway et al. [23], and even straightforward applications of the verifiable ledger as in Yazdinejad et al. [125]. Chen et al. propose zkML, a compiler that enables TensorFlow models to be translated readily into zk-SNARK halo2 circuits via either KZG or IPA commitments. This conversion allows for any portion, or the entirety of, an LLM to gain the properties of zero-knowledge, knowledge soundness, and completeness. Through this, and with potential connections to verifiable databases, zkML gains the powerful ability to audit inferences and ensure their accuracy. This research avenue is particularly promising due to both the efficient and potentially on-chain verification of zk-SNARKs as well as the extensibility of zkML to virtually any ML model.

Distinct from the zkML approach is that of Conway et al. with opML [23], which opts for an optimistic approach reliant on a fraud proof rather than a ZK proof to catch erroneous outputs within a certain challenge period. Clearly, there exist trade-offs in this implementation when compared to the zkML approach. Optimistic rollups are desirable in the sense that they are performant, but if implemented in a RAG environment, or similarly situated between the user and a model, latency issues can quickly become dominant. Apart from proof-oriented mechanisms and worth noting is the work of Yazdinejad et al. [125], which focuses on detecting deep fakes using blockchain's verifiable ledger. While not directly applicable to the realm of LLMs due to the non-atomic nature of data within a language model, important insights can be drawn from the paper. Namely, BC4LLMs could benefit greatly from a proposed hashing method applied to particularly sensitive data areas such as names, addresses, or even health-care related parts of corpora. This hash could be used as a guarantee of data veracity and could potentially prevent unsafe behaviors such as sycophancy, deception, or unfairness. Indeed, this hashing mechanism has the potential to be used as a final check for the LLM to verify that it is submitting information to the user that is consistent with standards agreed upon when information was originally committed to the ledger. Many similar vehicles for the maintenance of data integrity exist, albeit currently limited by scaling issues on-chain [27].

5. DATASETS RELEVANT TO BC4LLM

Developing synergistic technologies that incorporate blockchain and large language models is vital to securing a future consisting of safe, trustworthy LLMs. To this end, relevant datasets with which researchers may conduct experiments are of paramount importance. Moreover, it is often the case that blockchain-enabled systems require unorthodox training sets and edge cases in order to fully capture the dynamism and robustness of respective implementations. Therefore we have collected and summarize the relevance of certain datasets in Table 4. Of note are the standard datasets such as MNIST, CIFAR-10, SQuAD, and MS-MARCO, and we also give special attention to less widely known datasets which still may prove useful in certain academic settings.

6. PROMINENT CHALLENGES WITHIN BC4LLM

Despite the promise of the emerging BC4LLMs field, there are several innate challenges that delay progress and inhibit potential research directions. Typically, these are derived from certain limitations in blockchain technology, large language models, or deficits in the way that blockchain can serve large language models.

6.1 Corpus on Blockchain

Large language models require massive amounts of data to operate, with modern corpus sizes typically exceeding dozens of terabytes in volume [135]. This inherent quality of LLMs is situated opposite to problems that blockchain systems typically prefer to solve. Reconciling the limitations of blockchain's throughput and capacity for data handling with LLMs ballooning size is perhaps the most pressing issue in the BC4LLMs space.

Multiple attempts in which researchers have proposed zero-knowledge proofs for scalability have been discussed, such as in Wellington [113] and Singh [100]. However, employing zero-knowledge technology solely for scalability, and not privacy, is a doubtful long-term prospect due to the demands of generating zero-knowledge proofs, even with incredibly small circuits. Largely, this is due to the long-standing problem of the MSM within the context of generating ZKPs [120].

Table 4: **Datasets Relevant to BC4LLMs**

Dataset	Use Case	Description	Papers
MNIST[1]	Pattern Recognition	Images of handwritten digits for pattern recognition applications oe vulnerability analysis.	[116][114][75][32][97][57]
CIFAR-10[2]		Labeled images used in capacities from improving pattern recognition to zk-SNARK benchmarks.	[114][21][75]
MS MARCO[3]		Collection of human answered questions, used in training corpora as well as simulating RAG attacks.	[121][143][22]
MedMINST[4]		Collection of medical images from case studies.	[121][143][22]
Natural Questions[5]	Poisoned RAG	Open domain question answering dataset, incorporating questions from users and rigorous answers.	[121][143][22]
HotpotQA[6]		Question answering dataset with multi-hop questions and supervised, regulated, answers.	[143][22]
MT BENCH[7]	LLM Evaluation	Ranked pairwise expert human preferences for various model responses.	[138][15]
SQuAD[8]		Reading comprehension dataset comprised of questions posed on Wikipedia article with answers as sections of those corresponding articles.	[145][46]
IMDB Dataset[9]	Sentiment Analysis	Movie reviews	[145][46]
SafetyBench[10]	Safety Evaluation	Large number of multiple choice questions focused on evaluating the safety of large language models.	[133]
Tweets2011[11]	Sensitive Information Handling	List of scraped tweet identifiers and corresponding tweets from early 2011.	[145]
MTSamples Scrape[12]		Sample transcription medical reports from various disciplines and areas.	[145, 46]
DRC Diplomas[13]		Highschool diplomas from the Democratic Republic of the Congo.	[10]
HealthCareMagic[14]		Real patient-doctor conversations found through the HealthCareMagic website, capturing the nature of patient vocabulary.	[5]
Enron Emails[15]		Large set of emails generated by employees of the Enron Corporation.	[145, 46]
LLMGooAQ[16]		Comprehensive database capturing question and answers from a wide variety of domains.	[15]
GooAQ[17]		Large scale question answering dataset aimed at developing a vast selection of question types.	[15]
The Pile[18]		Massive and open source data set consisting of a combination of roughly 20 other datasets.	[134]

[1] https://yann.lecun.com/exdb/mnist/ [2] https://www.cs.toronto.edu/ kriz/cifar.html [3] https://microsoft.github.io/msmarco/
[4] https://medmnist.com/ [5] https://ai.google.com/research/NaturalQuestions [6] https://hotpotqa.github.io/
[7] https://paperswithcode.com/dataset/mt-bench [8] https://rajpurkar.github.io/SQuAD-explorer/
[9] https://developer.imdb.com/non-commercial-datasets/ [10] https://github.com/thu-coai/SafetyBench
[11] https://trec.nist.gov/data/tweets/ [12] https://mtsamples.com/ [13] https://minepst.gouv.cd/palmares-exetat/
[14] https://huggingface.co/datasets/RafaelMPereira/HealthCareMagic-100k-Chat-Format-en
[15] https://huggingface.co/datasets/preference-agents/enron-cleaned [16] https://github.com/mohaminemed/LLMGooAQ/
[17] https://huggingface.co/datasets/allenai/gooaq [18] https://pile.eleuther.ai/

Moreover, current WebGPU and/or WASM implementations could likely not support the throughput that client-based LLMs would require. For these reasons, it is unlikely that zero knowledge could be employed as a definitive solution for scalability issues within BC4LLMs without extensive developments in zk-SNARK generation research.

The disconnect between large language models' growing size and the aversion of blockchain's to store data readily on chain is a major research challenge.

6.2 Reliance on Oracles

The security guarantees of blockchain technology, while robust, often leave little room for interoperability with external systems [27]. That is, the blockchain can most easily interact with information on the chain, leaving little room to consider issues such as fact-checking or moral alignment. Oftentimes, to develop mechanisms that seek to provide assistance with LLM toxicity or factuality, oracles are used to bridge this gap [31][34]. Serving as mediators between chains and online sources, oracles are trusted parties that deliver information through a variety of protocols and frameworks. However, introducing a trusted party into an otherwise trustless system has been a long-standing weak point in this solution [74]. Exploring non-oracle-based options for ground truth solutions, or toxicity checks, would greatly enhance the security guarantees of blockchain within LLMs.

6.3 Energy Consumption

The crux of many issues concerning large language models stems from their need to consume and process inordinate amounts of data [135]. This requirement, in turn, creates a corresponding need for large language models to consume equally as colossal quantities of energy during their training stages, as well was during run time [69]. On the other side of the coin, we have energy issues within blockchain as well. Oftentimes, blockchain systems struggle to limit energy costs as the demands of both consensus mechanisms and the validation of proposed transactions incur huge computational costs [74]. Due to their tendency's to consume large amounts of compute resources, we see an unfortunate dissonance between implementing BC4LLM at scale without large efforts to cut back on energy costs, possibly through a shift away from both transformer architectures and proof of work type consensus mechanisms [81].

7. FUTURE RESEARCH DIRECTIONS

There exist several critically overlooked areas within LLMs that may benefit greatly from the introduction of blockchain technologies. The most prominent of these areas include blockchain federated unlearning, RAG, differential privacy, data provenance, and toxicity mitigation.

7.1 Blockchain Federated Unlearning

Privacy regulations are paramount in the online realm, especially concerning the "right to be forgotten" and user data privacy which are critical considerations when working with LLMs and blockchain. Federated blockchain unlearning offers LLMs the ability to erase learned data. Within our research, we identified four recent papers that have implemented blockchain federated unlearning frameworks. As noted previously in Section *4.1.1*, Zuo et al. [145] developed a federated TrustChain framework for blockchain-enhanced LLM training and unlearning, focusing on the impact of Low-Rank Adaptation (LoRA) hyperparameters on unlearning performances and integrating Hyperledger Fabric to ensure the security, transparency, and verifiability of the unlearning process. In another study, Zuo et al. [144] presented a trustworthy approach towards federated learning with blockchain-enhanced machine unlearning. This implementation differs from Trustchain, where Zuo et al. [144] used a machine unlearning mechanism that utilized two types of clients for training and unlearning, smart contracts for process automation, and a blockchain network for secure, immutable record-keeping. Beyond the above works [144], Liu et al. [64] introduced Blockchain Federated Unlearning (BlockFUL) as a versatile framework that redesigns the blockchain structure using a Chameleon Hash (CH) technology to simplify model updates and reduce the computational and consensus costs associated with unlearning tasks. Additionally, BlockFUL ensures the integrity and traceability of model updates, including privacy-preserving results from these blockchain-based unlearning operations [64]. Lin et al. [61] propose a framework with a proof of federated unlearning protocol that also utilizes the Chameleon hash function to verify data removal and eliminate the data contributions stored in other clients' models. Both, Liu et al. and Lin et al. use CH functions in their blockchain-enabled federated unlearning processes. The applications of key blockchain components, such as on-chain smart contracts and hash mappings for verifying data removal, may enable LLMs to forget personal data effectively. Blockchain for unlearning is an emerging area of research with significant potential for further innovation.

7.2 Blockchain Enhanced Retrieval Augmented Generation

Considering the novelty of RAG, there exists a plethora of research that focuses on how RAG may be attacked, consequently weakening the integrity of large language models. [121][5][26][143][46][127][22] However, there exists a vacuum yet to be filled with potential work concerning how these attacks may be mitigated, especially where defenses can make use of blockchain technology. In particular, calls for the inspection of blockchain's potential role in RAG deployment [9] have been made, and preliminary work done in exploring this intersection where it concerns blockchain's potential usefulness in user experience [129] and performance assessment [84] has been conducted. However, efforts concentrated on the enhancement of security and safety within RAG systems are absent from the current literature. A concerted effort towards bettering BC4LLMs where it concerns RAG would be both mutually and independently beneficial for blockchain technology and large language models.

7.3 Blockchain for Privacy Guarantees in Large Language Models

The clear connection between federated learning, blockchain, and large language models allows for the field of differential privacy to enter BC4LLMs' sphere of relevance. Major contributions concerning the impact of differential privacy on related areas such as deep learning have already been made [1], but issues such as privacy budget exhaustion still loom large in the space [14]. Moreover, despite conclusions that blockchain can help with privacy budget exhaustion [38, 136], few efforts have been conducted in exploring these solutions. Indeed, there is a need for more relevant research in order to realize the full measure of blockchain's impact on this area.

7.4 Blockchain for Transparency and Data Provenance in Large Language Models

Several recent papers have urged for increased data accountability measures to be placed on organizations developing large language models, especially where it concerns issues of data acquisition [12][41]. Additionally, worth noting are direct calls for the introduction of blockchain technology to help solve the issue of data provenance [117] in large language models. Largely, while this has been answered with responses in the realms of auditability [56], straightforward data tracking solutions have remained absent from the literature, despite relatively simple conceptual formulations [28]. Towards this goal of achieving improved data provenance within large language models corpus', RAG databases, and even in-context learning repositories, there is a need for more explorations into this natural application of distributed ledger technology to problems of explainable AI concerning large language models.

7.5 Blockchain for Non-toxic Large Language Models

Encompassing vital attributes such as ethics, legality, and non-violence, developing non-toxic large language models has been and will continue to be a major focus of the field for the foreseeable future [65][39][98]. There is no doubt that automated filtering of generated toxic content is one of the most pressing challenges concerning the safety of large language models [36][6]. This is because in essence, filtering inferences negatively impacts the quality of LLM responses, whereas manual human annotation is a costly and complex process [6]. Therefore, the applications of blockchain technology in this regard, while currently limited, are compelling. Considering one of the most groundbreaking achievements in ML within the past several years, federated learning has allowed for massive strides to be made within the spaces of securing training sets, user privacy, and even misinformation defense. A similar approach, aimed at the problem of toxicity, could be a hugely beneficial endeavor to the field. Moreover, imagining such a model is not difficult. Developing consensus around what is considered correct in a model and using that to propagate gradients and parameters is not dissimilar to the decisions that must be made about what is or is not toxic given the state of certain corpora. Given a concentrated research program, automated non-toxicity could very well have excellent solutions founded within the blockchain space.

8. CONCLUSION

In this survey, we presented how large language models have been applied to various facets of everyday life, revealing significant system vulnerabilities such as data poisoning, hallucinations, jailbreaking, and privacy attacks. Despite widespread recognition of these issues, effective mitigation strategies remain limited. Traditional machine learning solutions, such as differential privacy and federated learning have been applied, but fall short in providing comprehensive protection against the unique threats faced by LLMs. The adoption of blockchain technology presents a promising avenue for enhancing the security and safety of LLMs. Blockchain systems offer robust mechanisms for ensuring data integrity, provenance and encrypted frameworks which can be leveraged to bolster LLM defenses. By incorporating blockchain-based defenses, it is viable to achieve stronger privacy preservation, and reliable data, and offer LLMs to be more resilient against adversarial threats.

Furthermore, it is critical to establish clear definitions of security and safety in the context of LLM technologies. We conclude that security for LLMs pertains to the ability to tolerate applicable adversarial attacks while maintaining system integrity to provide consistent and accurate responses. Whereas safety for LLMs is the model's capacity to interact with users in a trustworthy manner, contingent upon adhering to ethical concerns, law-abiding, non-violent, fair, passively privacy-preserving, and informing. Additionally, differentiating between active and passive privacy measures will aid in developing more targeted and effective privacy-preserving strategies. These distinctions and definitions provide a foundational framework for future research in the BC4LLM space. From analyzing the integration of blockchain technology and LLMs, we provided a taxonomy in Figure 3 where previous research done in the field of BC4LLMs can apply to security and safety problems that LLMs face. We recognize various gaps in the BC4LLM space that need to be looked into for further consideration. In refining our understanding of relevant concepts, we see that the intersection of blockchain technology and LLMs holds significant potential for addressing the current shortcomings in LLM security and safety. With our contributions, we can use BC4LLM to provide more secure, reliable, and safe AI systems.

References

[1] Martin Abadi et al. "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* 2016, pp. 308–318.

[2] Sara Abdali et al. "Securing Large Language Models: Threats, Vulnerabilities and Responsible Practices". In: *arXiv preprint arXiv:2403.12503* (2024).

[3] Josh Achiam et al. *GPT-4 Technical Report.* 2024.

[4] Omar Ali et al. "A Comparative Study: Blockchain Technology Utilization Benefits, Challenges and Functionalities". In: *IEEE Access* 9 (2021), pp. 12730–12749.

[5] Maya Anderson, Guy Amit, and Abigail Goldsteen. "Is My Data in Your Retrieval Database? Membership Inference Attacks Against Retrieval Augmented Generation". In: *arXiv preprint arXiv:2405.20446* (2024).

[6] Usman Anwar et al. *Foundational Challenges in Assuring Alignment and Safety of Large Language Models.* 2024.

[7] Pathum Chamikara Mahawaga Arachchige et al. "A Trustworthy Privacy Preserving Framework for Machine Learning in Industrial IoT Systems". In: *IEEE Transactions on Industrial Informatics* 16.9 (2020), pp. 6092–6102.

[8] Sana Awan et al. “Poster: A Reliable and Accountable Privacy-Preserving Federated Learning Framework using the Blockchain”. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 2561–2563.

[9] Anandaganesh Balakrishnan. *Enhancing Data Engineering Efficiency With AI: Utilizing Retrieval-augmented Generation, Reinforcement Learning From Human Feedback, and Fine-tuning Techniques*. 2024.

[10] Sree Bhargavi Balija, Amitash Nanda, and Debashis Sahoo. “Building Communication Efficient Asynchronous Peer-to-Peer Federated LLMs with Blockchain”. In: *Proceedings of the AAAI Symposium Series* 3.1 (2024), pp. 288–292.

[11] Nishant Baranwal Somy et al. “Ownership Preserving AI Market Places Using Blockchain”. In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 156–165.

[12] Emily M. Bender et al. “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big”. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT ’21. 2021, pp. 610–623. DOI: 10.1145/3442188.3445922.

[13] Dhanasak Bhumichai et al. “The Convergence of Artificial Intelligence and Blockchain: The State of Play and the Road Ahead”. In: *Information* 15.5 (2024).

[14] Anis Bkakria et al. “Optimal Distribution of Privacy Budget in Differential Privacy”. In: *Risks and Security of Internet and Systems: 13th International Conference, CRiSIS 2018, Arcachon, France, October 16–18, 2018, Revised Selected Papers 13*. 2019, pp. 222–236.

[15] Mouhamed Amine Bouchiha et al. “LLMChain: Blockchain-based Reputation System for Sharing and Evaluating Large Language Models”. In: *arXiv preprint arXiv:2404.13236* (2024).

[16] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020.

[17] Vitalik Buterin. “Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.” In: *white paper* 3.37 (2014).

[18] Davide Calvaresi et al. “Explainable Multi-Agent Systems Through Blockchain Technology”. In: *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*. Springer International Publishing, 2019, pp. 41–58.

[19] Bin Cao et al. “Blockchain Systems, Technologies, and Applications: A Methodology Perspective”. In: *IEEE Communications Surveys & Tutorials* 25.1 (2023), pp. 353–385.

[20] Nicholas Carlini et al. *Extracting Training Data from Diffusion Models*. 2023.

[21] Bing-Jyue Chen et al. “ZKML: An Optimizing System for ML Inference in Zero-Knowledge Proofs”. In: *Proceedings of the Nineteenth European Conference on Computer Systems*. 2024, pp. 560–574.

[22] Pengzhou Cheng et al. “TrojanRAG: Retrieval-Augmented Generation Can Be Backdoor Driver in Large Language Models”. In: *arXiv preprint arXiv:2405.13401* (2024).

[23] K. D. Conway et al. *opML: Optimistic Machine Learning on Blockchain*. 2024.

[24] Jerry Cuomo. *How blockchain adds trust to AI and IoT*. 2020.

[25] Chengyuan Deng et al. “Deconstructing The Ethics of Large Language Models from Long-standing Issues to New-emerging Dilemmas”. In: *arXiv preprint arXiv:2406.05392* (2024).

[26] Gelei Deng et al. “Pandora: Jailbreak GPTs by Retrieval Augmented Generation Poisoning”. In: *arXiv preprint arXiv:2402.08416* (2024).

[27] Advait Deshpande et al. “Distributed Ledger Technologies/Blockchain: Challenges, opportunities and the prospects for standards”. In: *Overview report The British Standards Institution (BSI)* 40.40 (2017), pp. 1–34.

[28] Thang N. Dinh and My T. Thai. “AI and Blockchain: A Disruptive Integration”. In: *Computer* 51.9 (2018), pp. 48–53.

[29] Sidi Boubacar ElMamy et al. “A Survey on the Usage of Blockchain Technology for Cyber-Threats in the Context of Industry 4.0”. In: *Sustainability* 12.21 (2020), p. 9179.

[30] Thomas F. Heston. “Perspective Chapter: Integrating Large Language Models and Blockchain in Telemedicine”. In: *A Comprehensive Overview of Telemedicine [Working Title]*. IntechOpen, 2024.

[31] Shaokun Fan et al. “Blockchain as a trust machine: From disillusionment to enlightenment in the era of generative AI”. In: *Decision Support Systems* 182 (2024).

[32] Andras Ferenczi and Costin Bădică. “A Fully Decentralized Privacy-Enabled Federated Learning System”. In: *Computational Collective Intelligence: 15th International Conference, ICCCI Proceedings*. 2023, pp. 444–456.

[33] Reza Fotohi, Fereidoon Shams Aliee, and Bahar Farahani. “Decentralized and robust privacy-preserving model using blockchain-enabled Federated Deep Learning in intelligent enterprises”. In: *Applied Soft Computing* 161 (2024), p. 111764.

[34] Paula Fraga-Lamas and Tiago M. Fernández-Caramés. “Fake News, Disinformation, and Deepfakes: Leveraging Distributed Ledger Technologies and Blockchain to Combat Digital Deception and Counterfeit Reality”. In: *IT Professional* (2020).

[35] Deep Ganguli et al. *Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned*. 2022.

[36] Samuel Gehman et al. “RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 3356–3369.

[37] Yuanhao Gong. *Dynamic Large Language Models on Blockchains*. 2023.

[38] Leong Mei Han, Yang Zhao, and Jun Zhao. *Blockchain-Based Differential Privacy Cost Management System*. 2020.

[39] Tessa Han et al. *Towards Safe Large Language Models for Medicine*. 2024.

[40] Stefan Harrer. “Attention is not all you need: the complicated case of ethically using large language models in healthcare and medicine”. In: *EBioMedicine* 90 (2023).

[41] Kai He et al. *A Survey of Large Language Models for Healthcare: from Data, Technology, and Applications to Accountability and Ethics*. 2024.

[42] Zheyuan He et al. “Large Language Models for Blockchain Security: A Systematic Literature Review”. In: *arXiv preprint arXiv:2403.14280* (2024).

[43] Dan Hendrycks et al. “Aligning AI With Shared Human Values”. In: 2020.

[44] Hongsheng Hu et al. *Membership Inference Attacks on Machine Learning: A Survey*. 2022.

[45] Qinwen Hu, Muhammad Rizwan Asghar, and Sherali Zeadally. “Blockchain-based public ecosystem for auditing security of software applications”. In: *Computing* 103.11 (2021), pp. 2643–2665.

[46] Zhibo Hu et al. “Prompt Perturbation in Retrieval-Augmented Generation based Large Language Models”. In: *arXiv preprint arXiv:2402.07179* (2024).

[47] Xiaowei Huang et al. “A survey of safety and trustworthiness of large language models through the lens of verification and validation”. In: *Artificial Intelligence Review* 57.7 (2024).

[48] Rahmeh Fawaz Ibrahim, Qasem Abu Al-Haija, and Ashraf Ahmad. “DDoS Attack Prevention for Internet of Thing Devices Using Ethereum Blockchain Technology”. In: *Sensors* 22.18 (2022).

[49] Hakan Inan et al. *Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations*. 2023.

[50] Ziwei Ji et al. “Survey of Hallucination in Natural Language Generation”. In: *ACM Computing Surveys* 55.12 (2023), pp. 1–38.

[51] Haibo Jin et al. “JailbreakZoo: Survey, Landscapes, and Horizons in Jailbreaking Large Language and Vision-Language Models”. In: *arXiv preprint arXiv: 2407.01599* (2024).

[52] Enkelejda Kasneci et al. “ChatGPT for good? On opportunities and challenges of large language models for education”. In: *Learning and Individual Differences* 103 (2023).

[53] Safak Kayikci and Taghi M. Khoshgoftaar. “Blockchain meets machine learning: a survey”. In: *Journal of Big Data* 11.1 (2024).

[54] Marwa Keshk et al. “A Privacy-Preserving-Framework-Based Blockchain and Deep Learning for Protecting Smart Power Networks”. In: *IEEE Transactions on Industrial Informatics* 16.8 (2020), pp. 5110–5118.

[55] Siwon Kim et al. “ProPILE: Probing Privacy Leakage in Large Language Models”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[56] Lu Li, Jiwei Qin, and Jintao Luo. “A Blockchain-Based Federated-Learning Framework for Defense against Backdoor Attacks”. In: *Electronics* 12.11 (2023).

[57] Zonghang Li et al. “Byzantine Resistant Secure Blockchained Federated Learning at the Edge”. In: *IEEE Network* 35.4 (2021), pp. 295–301.

[58] Zhiding Liang et al. “Unleashing the potential of llms for quantum computing: A study in quantum architecture design”. In: *arXiv preprint arXiv:2307.08191* (2023).

[59] Fudong Lin et al. “Comprehensive transformer-based model architecture for real-world storm prediction”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2023, pp. 54–71.

[60] Fudong Lin et al. “MMST-ViT: Climate Change-aware Crop Yield Prediction via Multi-Modal Spatial-Temporal Vision Transformer”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 5774–5784.

[61] Yijing Lin et al. *Blockchain-enabled Trustworthy Federated Unlearning*. 2024.

[62] Bo Liu et al. *When Machine Learning Meets Privacy: A Survey and Outlook*. 2021.

[63] Ming Liu et al. *Federated Learning Meets Natural Language Processing: A Survey*. 2021.

[64] Xiao Liu et al. *Decentralized Federated Unlearning on Blockchain*. 2024.

[65] Yang Liu et al. *Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment*. 2024.

[66] Yi Liu et al. *Prompt Injection attack against LLM-integrated Applications*. 2024.

[67] Vasco Lopes and Luís A. Alexandre. “An Overview of Blockchain Integration with Robotics and Artificial Intelligence”. In: *arXiv preprint arXiv:1810.00329* (2018).

[68] Nils Lukas et al. *Analyzing Leakage of Personally Identifiable Information in Language Models*. 2023.

[69] Haoxiang Luo, Jian Luo, and Athanasios V. Vasilakos. “BC4LLM: Trusted Artificial Intelligence When Blockchain Meets Large Language Models”. In: *arXiv preprint arXiv:2310.06278* (2023).

[70] Sameera K. M. et al. “Privacy-Preserving in Blockchain-based Federated Learning Systems”. In: *Computer Communications* (2024).

[71] Diksha Malhotra, Poonam Saini, and Awadhesh Kumar Singh. “Blockchain-based proof-of-authenticity frameworks for Explainable AI”. In: *Multimedia Tools and Applications* 83.13 (2024), pp. 37889–37911.

[72] Joshua Maynez et al. *On Faithfulness and Factuality in Abstractive Summarization*. 2020.

[73] Jean Gilbert Mbula Mboma et al. “Integrating LLM with Blockchain and IPFS to Enhance Academic Diploma Integrity”. In: *2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*. 2024, pp. 1–6.

[74] Jean Gilbert Mbula Mboma et al. “Assessing How Large Language Models Can Be Integrated with or Used for Blockchain Technology: Overview and Illustrative Case Study”. In: *2023 27th International Conference on Circuits, Systems, Communications and Computers (CSCC)*. 2023, pp. 59–70.

[75] H. Brendan McMahan et al. *Communication-Efficient Learning of Deep Networks from Decentralized Data.* 2017.

[76] Ralph C. Merkle. “A Digital Signature Based on a Conventional Encryption Function”. In: Springer, 1988, pp. 369–378.

[77] Du Mingxiao et al. “A review on consensus algorithm of blockchain”. In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC).* 2017, pp. 2567–2572.

[78] Anudit Nagar. *Privacy-Preserving Blockchain Based Federated Learning with Differential Data Sharing.* 2019.

[79] Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. In: (2008).

[80] Seth Neel and Peter Chang. *Privacy Issues in Large Language Models: A Survey.* 2024.

[81] Cong T. Nguyen et al. “Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities”. In: *IEEE Access* (2019).

[82] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. *Med-HALT: Medical Domain Hallucination Test for Large Language Models.* 2023.

[83] Xudong Pan et al. “Privacy Risks of General-Purpose Language Models”. In: *2020 IEEE Symposium on Security and Privacy (SP).* 2020, pp. 1314–1331.

[84] Young-Hoon Park, Yejin Kim, and Junho Shim. “Blockchain-Based Privacy-Preserving System for Genomic Data Management Using Local Differential Privacy”. In: *Electronics* 10.23 (2021).

[85] Attia Qammar et al. “Securing federated learning with blockchain: a systematic literature review”. In: *Artificial Intelligence Review* 56.5 (2023), pp. 3951–3985.

[86] Youyang Qu et al. “Blockchain-enabled Federated Learning: A Survey”. In: *ACM Computing Surveys* 55.4 (2023), pp. 1–35.

[87] Liz Reid. *Generative AI in Search: Let Google do the searching for you.* 2024.

[88] Paul Röttger et al. *SafetyPrompts: a Systematic Review of Open Datasets for Evaluating and Improving Large Language Model Safety.* 2024.

[89] Kangrui Ruan et al. “S2e: Towards an end-to-end entity resolution solution from acoustic signal”. In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2024, pp. 10441–10445.

[90] Sara Saberi et al. “Blockchain technology and its relationships to sustainable supply chain management”. In: *International Journal of Production Research* 57.7 (2019), pp. 2117–2135.

[91] Khaled Salah et al. “Blockchain for AI: Review and Open Research Challenges”. In: *IEEE Access* 7 (2019), pp. 10127–10149.

[92] Oshani Seneviratne. “Blockchain for Social Good: Combating Misinformation on the Web with AI and Blockchain”. In: *Proceedings of the 14th ACM Web Science Conference 2022.* 2022, pp. 435–442.

[93] Muhammad Shafay et al. “Blockchain for deep learning: review and open challenges”. In: *Cluster Computing* 26.1 (2023), pp. 197–221.

[94] Zawar Shah et al. “Blockchain Based Solutions to Mitigate Distributed Denial of Service (DDoS) Attacks in the Internet of Things (IoT): A Survey”. In: *Sensors* 22 (2022).

[95] Mrinank Sharma et al. *Towards Understanding Sycophancy in Language Models.* 2023.

[96] Erfan Shayegani et al. *Survey of Vulnerabilities in Large Language Models Revealed by Adversarial Attacks.* 2023.

[97] Meng Shen et al. “Exploiting Unintended Property Leakage in Blockchain-Assisted Federated Learning for Intelligent Edge Computing”. In: *IEEE Internet of Things Journal* 8.4 (2021), pp. 2265–2275.

[98] Tianhao Shen et al. *Large Language Model Alignment: A Survey.* 2023.

[99] Saurabh Singh et al. “Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city”. In: *Sustainable Cities and Society* 63 (2020).

[100] Shridhar Singh. “Enhancing Privacy and Security in Large-Language Models: A Zero-Knowledge Proof Approach”. In: *International Conference on Cyber Warfare and Security* 19.1 (2024), pp. 574–582.

[101] Tobin South et al. “Secure Community Transformers: Private Pooled Data for LLMs”. In: (2023).

[102] Lichao Sun et al. *TrustLLM: Trustworthiness in Large Language Models.* 2024.

[103] Xiaoqiang Sun et al. “A Survey on Zero-Knowledge Proof in Blockchain”. In: *IEEE Network* 35.4 (2021).

[104] Hamed Taherdoost. “Blockchain Technology and Artificial Intelligence Together: A Critical Review on Applications”. In: *Applied Sciences* 12.24 (2022), p. 12948.

[105] Simone Tedeschi et al. *ALERT: A Comprehensive Benchmark for Assessing Large Language Models' Safety through Red Teaming.* 2024.

[106] Timm Teubner et al. “Welcome to the Era of ChatGPT et al.” In: *Business & Information Systems Engineering* 65 (2023).

[107] Casey Tonkin. *'ChatGPT, help me make a bomb'.* Information Age. 2023.

[108] Imdad Ullah et al. “Privacy Preserving Large Language Models: ChatGPT Case Study Based Vision and Framework”. In: *arXiv preprint arXiv:2310.12523* (2023).

[109] Yixin Wan et al. *"Kelly is a Warm Person, Joseph is a Role Model": Gender Biases in LLM-Generated Reference Letters.* 2023.

[110] Boxin Wang et al. *DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models.* 2023.

[111] Qianlong Wang et al. “AI at the Edge: Blockchain-Empowered Secure Multiparty Learning With Heterogeneous Models”. In: *IEEE Internet of Things Journal* 7.10 (2020), pp. 9600–9610.

[112] Laura Weidinger et al. *Sociotechnical Safety Evaluation of Generative AI Systems.* 2023.

[113] Sean Wellington. "BasedAI: A decentralized P2P network for Zero Knowledge Large Language Models (ZK-LLMs)". In: *arXiv preprint arXiv:2403.01008* (2024).

[114] Chenkai Weng et al. *Mystique: Efficient Conversions for Zero-Knowledge Proofs with Applications to Machine Learning.* 2021.

[115] Jian Weng et al. "Auditable privacy protection deep learning platform construction method based on block chain incentive mechanism". US Patent 11,836,616. 2023.

[116] Jiasi Weng et al. "DeepChain: Auditable and Privacy-Preserving Deep Learning with Blockchain-Based Incentive". In: *IEEE Transactions on Dependable and Secure Computing* 18.5 (2021), pp. 2438–2455.

[117] Karl Werder, Balasubramaniam Ramesh, and Rongen (Sophia) Zhang. "Establishing Data Provenance for Responsible Artificial Intelligence Systems". In: *ACM Transactions on Management Information Systems* 2 (2022), pp. 1–23.

[118] Amy Winograd. "Loose-Lipped Large Language Models Spill Your Secrets: The Privacy Implications of Large Language Models Notes". In: *Harvard Journal of Law & Technology (Harvard JOLT)* 2 (2022).

[119] Leon Witt et al. *Blockchain and Artificial Intelligence: Synergies and Conflicts.* 2024.

[120] Charles F. Xavier. *PipeMSM: Hardware Acceleration for Multi-Scalar Multiplication.* Cryptology ePrint Archive, Paper 2022/999. 2022.

[121] Jiaqi Xue et al. "BadRAG: Identifying Vulnerabilities in Retrieval Augmented Generation of Large Language Models". In: *ArXiv preprint arXiv:2406.00083* (2024).

[122] Biwei Yan et al. *On Protecting the Data Privacy of Large Language Models (LLMs): A Survey.* 2024.

[123] Jingfeng Yang et al. *Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond.* 2023.

[124] Yifan Yao et al. "A Survey on Large Language Model (LLM) Security and Privacy: The Good, The Bad, and The Ugly". In: *High-Confidence Computing* (2024), p. 100211.

[125] Abbas Yazdinejad et al. "Making Sense of Blockchain for AI Deepfakes Technology". In: *2020 IEEE Globecom Workshops (GC Wkshps.* 2020, pp. 1–6.

[126] Hongbin Ye et al. *Cognitive Mirage: A Review of Hallucinations in Large Language Models.* 2023.

[127] Shenglai Zeng et al. "The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG)". In: *arXiv preprint arXiv:2402.16893* (2024).

[128] Rui Zhang, Rui Xue, and Ling Liu. "Security and Privacy on Blockchain". In: *ACM Computing Surveys* 52.3 (2020).

[129] Ruichen Zhang et al. *Interactive AI with Retrieval-Augmented Generation for Next Generation Networking.* 2024.

[130] Yue Zhang et al. *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models.* 2023.

[131] Zhenjie Zhang et al. *Proof of Quality: A Costless Paradigm for Trustless Generative AI Model Inference on Blockchains.* 2024.

[132] Zhexin Zhang et al. *Defending Large Language Models Against Jailbreaking Attacks Through Goal Prioritization.* 2024.

[133] Zhexin Zhang et al. *SafetyBench: Evaluating the Safety of Large Language Models with Multiple Choice Questions.* 2023.

[134] Shuai Zhao et al. "Prompt as Triggers for Backdoor Attack: Examining the Vulnerability in Language Models". In: 2023.

[135] Wayne Xin Zhao et al. *A Survey of Large Language Models.* 2023.

[136] Yang Zhao et al. "A Blockchain-Based Approach for Saving and Tracking Differential-Privacy Cost". In: *IEEE Internet of Things Journal* 8.11 (2021), pp. 8865–8882.

[137] Yang Zhao et al. "Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices". In: *IEEE Internet of Things Journal* 8.3 (2021), pp. 1817–1829.

[138] Lianmin Zheng et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.* 2024.

[139] Zibin Zheng et al. *A Survey of Large Language Models for Code: Evolution, Benchmarking, and Future Trends.* 2024.

[140] Zibin Zheng et al. "An overview on smart contracts: Challenges, advances and platforms". In: *Future Generation Computer Systems* 105 (2020), pp. 475–491.

[141] Xudong Zhu, Hui Li, and Yang Yu. "Blockchain-Based Privacy Preserving Deep Learning". In: *Information Security and Cryptology.* 2019, pp. 370–383.

[142] Jun Zhuang and Casey Kennington. "Understanding survey paper taxonomy about large language models via graph representation learning". In: *arXiv preprint arXiv:2402.10409* (2024).

[143] Wei Zou et al. "PoisonedRAG: Knowledge Poisoning Attacks to Retrieval-Augmented Generation of Large Language Models". In: *arXiv preprint arXiv:2402.07867* (2024).

[144] Xuhan Zuo et al. *Federated Learning with Blockchain-Enhanced Machine Unlearning: A Trustworthy Approach.* 2024.

[145] Xuhan Zuo et al. *Federated TrustChain: Blockchain-Enhanced LLM Training and Unlearning.* 2024.