THE JBBA

PEER-REVIEWED RESEARCH

OPEN ACCESS
ISSN Online: 2516-3957
ISSN Print: 2516-3949
https://doi.org/10.31585/jbba-7-1-(3)2024

Towards Confidential Chatbot Conversations: A Decentralised Federated Learning Framework

Hongxu Su[1], Cheng Xiang[1], Bharath Ramesh[2]
[1]*Department of Electrical and Computer Engineering, National University of Singapore, Singapore*
[2]*International Center for Neuromorphic Systems, Western Sydney University, Sydney, Australia*

Correspondence: bharath.ramesh03@u.nus.edu

Received: 13 December 2023 **Accepted**: 14 January 2024 **Published**: 28 February 2024

Abstract

The development of cutting-edge large language models such as ChatGPT has sparked global interest in the transformative potential of chatbots to automate language tasks. However, alongside the remarkable advancements in natural language processing, concerns about user privacy and data security have become prominent challenges that need immediate attention. In response to these critical concerns, this article presents a novel approach that addresses the privacy and security issues in chatbot applications. We propose a secure and privacy-preserving framework for chatbot systems by leveraging the power of decentralised federated learning (DFL) and secure multi-party computation (SMPC). Our DFL framework leverages blockchain smart contracts for participant selection, orchestrating the training process on user data while keeping the data local, and model distribution. After each round of local training by the participants, the blockchain network securely aggregates the model updates using SMPC, ensuring that participants' raw model parameters are not exposed to others. The global model is encrypted and stored in hypermedia protocols such as the InterPlanetary File System. Participants decrypt the global model updates using their private keys to further fine-tune their models. Iterative training rounds are executed through the blockchain network, with participants updating the model collaboratively using SMPC. Experiments show that our approach achieves comparable performance to centralised models while offering significant improvements in privacy and security. This article presents a ground-breaking solution to privacy and security challenges in chatbots, and we hope our approach will foster trust and encourage broader adoption of chatbot technology with privacy at the forefront.

Keywords: *Large language models, Privacy-centric machine learning, Decentralized federated learning, Multi-part computation, Knowledge distillation, Quantized language models*

JEL Classifications: *Privacy-Preserving Learning, Decentralised Federated Learning, Tiny Language Models (TinyLMs), Secure Multi-party Computation (SMPC), Blockchain Technology*

1. Introduction

Natural language processing (NLP) and Large language models (LLMs) have recently revolutionised human–computer interaction [1]. Advanced LLMs such as ChatGPT by OpenAI have shown their potential to transform various industries and automate language tasks on an unprecedented scale [2]. However, this surge in useful applications has also raised significant concerns about privacy, trust, and user data exploitation [3]. As these LLMs process large amounts of user data for training and fine-tuning, it is essential to address the potential risks associated with unauthorised data access, breaches, and misuse. Striking a delicate balance between leveraging the power of chatbot technology and protecting user privacy is a critical challenge for the widespread adoption and ethical deployment of these revolutionary NLP systems.

Federated learning (FL), originally proposed by McMahan et al. [4], is a promising solution for preserving user privacy, especially in the context of NLP technologies [5]. FL enables model training by distributing the learning process across individual user devices, thereby avoiding the need to centralise sensitive data on a single server. This approach keeps user data localised, encrypted, and under the user's control, ensuring that no raw personal information is exposed during the training process. By aggregating model updates from multiple users without sharing their individual data, FL enhances privacy protection and minimises the risk of data breaches and unauthorised access.

Despite the significant privacy advantages offered by FL, centralised FL implementations still pose certain threats. A centralised FL setup introduces the possibility of a single point of failure, where the central server becomes vulnerable to attacks, potentially compromising the privacy of users. Additionally, model privacy concerns arise as the central server might have access to aggregated model updates from various users, raising the risk of information leakage or even malicious central servers as an extreme example [6]. As such, striking the right balance between leveraging the benefits of FL's privacy-preserving capabilities and mitigating the challenges of centralised deployment remains a crucial area of research for fostering trust and upholding user privacy in the dynamic landscape of NLP.

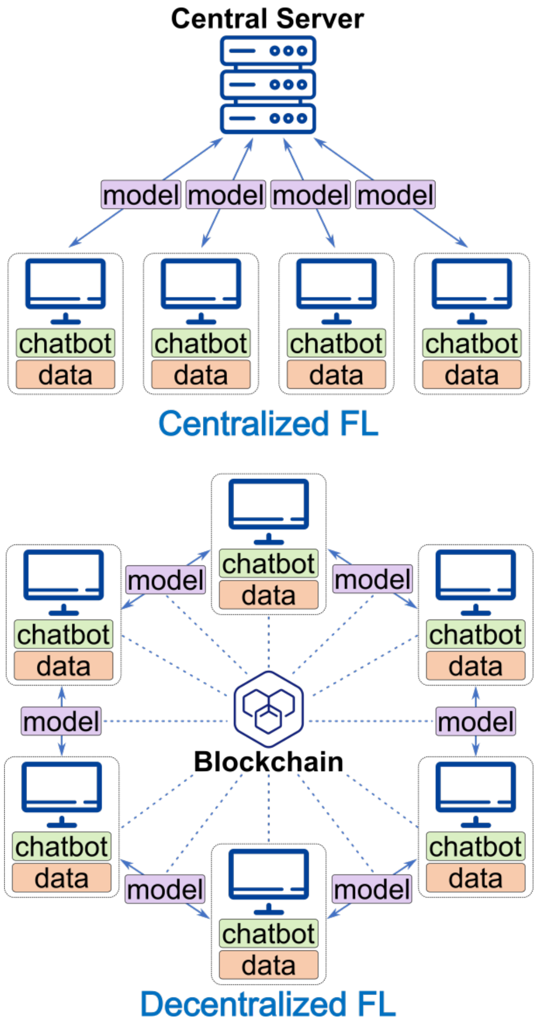


Figure 1. Centralised vs decentralised federated learning.

Blockchain-based learning is a promising alternative to centralised FL for addressing concerns about user privacy [7–9], especially for NLP technologies [10]. This approach mitigates the risks associated with a single point of failure by leveraging the decentralised and distributed nature of blockchain networks. In blockchain-based FL, or decentralised federated learning (DFL), participants (nodes) collaborate directly on the blockchain, contributing their encrypted model updates while maintaining control over their individual data [11]. The tamper-resistant nature of blockchain ensures data integrity and prevents unauthorised access, offering a more secure and privacy-preserving environment. Moreover, the use of blockchain smart contracts for aggregating model updates enables transparent and trustless computations without compromising individual users' data privacy. Embracing blockchain-based FL has the potential to revolutionise the chatbot landscape by instilling user confidence and reinforcing the protection of sensitive information throughout the FL process. Figure 1 showcases the differences between a centralised and a decentralised process and highlights the key differences in the setup, which is the blockchain infrastructure orchestrating the processing of FL instead of a central server.

Firstly, blockchain-based DFL offers promising solutions to user privacy concerns in chatbot applications, but it also introduces specific challenges that need to be carefully considered, as discussed in this latest survey article [11]. One of the main challenges is the scalability and latency of blockchain networks. Because DFL involves multiple participants performing computations and sharing model updates on the blockchain, the sheer volume of data and transactions may result in slower processing times and increased network congestion. InterPlanetary File System (IPFS) [12] can be adopted to address scalability and latency concerns. IPFS allows participants to store models without explicitly relying on the blockchain infrastructure, making it an ideal solution for model communication and storage in DFL. The storage burden is distributed across participants with IPFS, which alleviates the scalability issues faced by a central server or the blockchain network itself.

Secondly, different blockchains use various consensus algorithms, such as Proof-of-Work, Proof-of-Stake, or Practical Byzantine Fault Tolerance [11]. The choice of consensus mechanism affects network performance, energy consumption, and the level of decentralisation. To address this challenge, the DFL process can be adapted to different consensus mechanisms, ensuring compatibility with the selected blockchain. This adaptability allows blockchain-based DFL to optimise its performance while maintaining its privacy-preserving attributes.

Thirdly, blockchain-based DFL faces the challenge of selecting a suitable model evaluation mechanism without compromising on security. We address this by ensuring differential privacy-enabled models [13] are used for peer evaluation and subsequently rewarding users for their participation in the evaluation phase, instead of users allowing access to raw model parameters that may potentially expose the training data via inversion attacks. A related issue is how to perform the model aggregation for the FL process at the end of each epoch without exposing the models of each user. To this end, we implement secure multi-party computation (SMPC) techniques to enable collaborative model aggregation across multiple participants [14]. SMPC can let multiple users combine their private models without knowing each other's inputs. To the best of our knowledge, we are the first to introduce this for blockchain-based FL [11].

Finally, deploying LLMs locally for privacy-preserving DFL chatbot applications is challenging because they require a lot of computational power, especially during inference tasks. LLMs, such as GPT-4, have shown impressive language-generation capabilities, but their large size and complexity require powerful hardware resources for efficient real-time performance [15]. Local deployment on resource-constrained devices can result in slow response times, increased latency, and potential memory constraints, which can hinder the seamless user experience that is critical for chatbot interactions. Similarly, fully homomorphic encryption-enabled LLMs increase the latency while trying to preserve user privacy [16]. Additionally, the continuous evolution of LLMs with ongoing updates and improvements requires consistent access to the latest model versions, which may be impractical to maintain locally. As a result, striking a balance between leveraging the capabilities of LLMs and the computational constraints of local deployment is a critical consideration for achieving optimal performance in chatbot applications.

One way to address the challenges of deploying LLMs locally for DFL applications is to limit the chatbot's functionality and embrace Tiny Language Models (TinyLMs) [17]. TinyLMs are smaller versions of LLMs that have been optimised for specific tasks or domains, reducing the model size and computational requirements without sacrificing much performance. By using TinyLMs that are tailored to the specific needs of the application, one can achieve a more lightweight and responsive deployment, making it feasible to run the chatbot on resource-constrained devices. This strategic use of TinyLMs allows chatbot developers to strike a balance between offering valuable language processing capabilities and ensuring a smooth user experience without the burden of deploying unwieldy LLMs locally.

In summary, by combining the power of blockchain-based DFL with the generation of TinyLMs, we present a novel framework that revolutionises how language models are trained and deployed. Notably, our contribution extends beyond conventional methods by being among the first to implement SMPC in the context of DFL. This innovation ensures that participants' data remains confidential during the collaborative model aggregation process, enhancing the privacy and security of the overall system. In addition, the generation of TinyLMs through fine-tuning, distillation, and quantisation enables the creation of efficient language models suitable for deployment on resource-constrained devices. The next section details our framework, Section 3 demonstrates the effectiveness of our approach, and, finally, the article is concluded in Section 4.

2. Decentralised Federated Learning for TinyLMs

Secure DFL is a type of machine learning that allows multiple devices to collaboratively train a collaborative model without sharing their data and model with each other. This is done by having each device train a local model on its own data and then periodically exchanging encrypted updates with the other devices. The updates are then used to train a global model that is shared by all of the devices. Our blockchain-based DFL framework uses a smart contract to manage the entire process. The framework also includes a number of features that make it well-suited for NLP applications, including support for different learning algorithms, scalability, and security using SMPC, as shown in Figure 2.

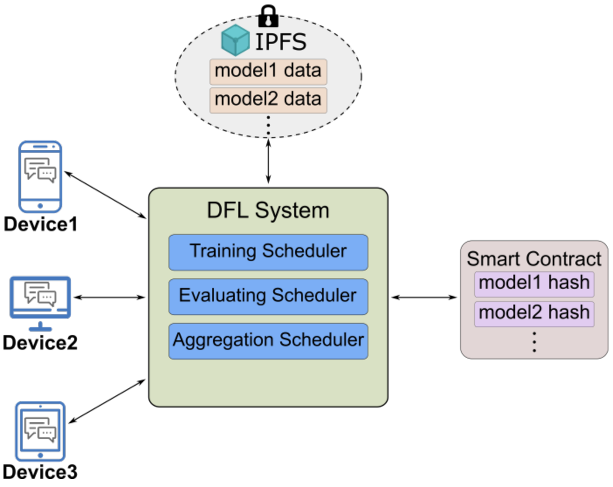


Figure 2. General architecture of our DFL for TinyLMs.

Iterative Federated Learning: Participants are identified as nodes based on predefined criteria such as a registration fee, ensuring their active participation in the DFL process. The baseline dataset is distributed among the selected participants. Each node also possesses its local data and performs individual model training combining it with the distributed dataset, fostering baseline model performance. The training scheduler determines the timing and frequency of training rounds, allowing participants' devices to contribute model updates at specified intervals. Participants contribute to model improvement by providing updated model parameters during each iteration. This iterative process allows the model to learn from various data distributions and adapt to diverse user preferences, which promote continuous model refinement.

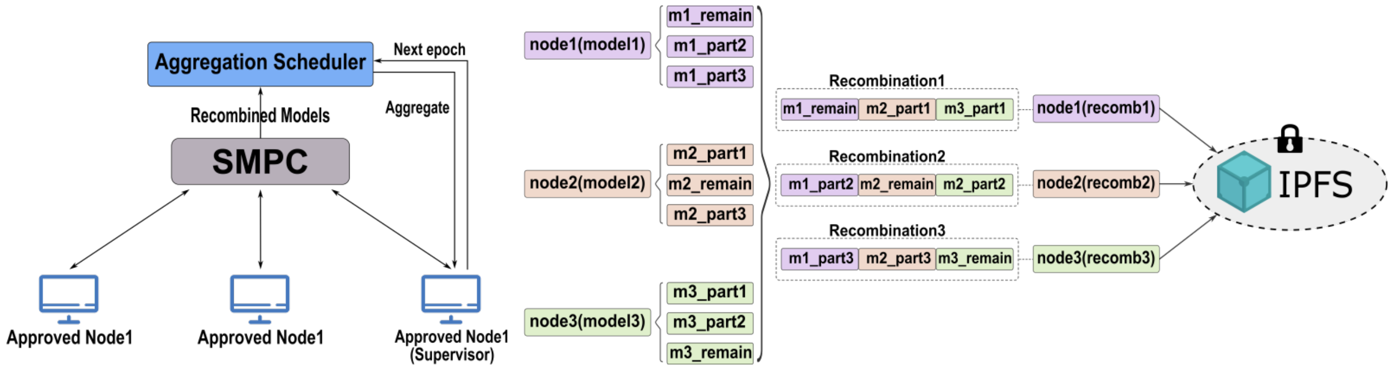


Figure 3. Aggregation scheduler of DFL using SMPC. Each node shares partial models with the other nodes, each of which is encrypted using the recipient node's public key for secure transmission, by uploading them onto the IPFS. Note that the partial model and the remaining model at each node when recombined constitute the original model. Each node now uses the partial models obtained from the other nodes and its own remaining model to recombine them. All these recombined models by the supervisor node, which is randomly chosen at each epoch, can now be securely averaged to get the averaged global model at the end of each aggregation phase. Since the partial models are securely shared with each node, the SMPC process allows averaging by distributing the data without revealing the model of any single node.

Secure Storage: All communication between blockchain nodes and transactions is carefully protected using cryptographic protocols and the blockchain's inherent consensus mechanisms. The global model is encrypted before being stored in the IPFS for added security and privacy. Nodes use their private keys to decrypt the model updates, updating their local models without revealing any sensitive data. This process ensures that the model updates are secure and private, while also allowing for efficient aggregation and distribution.

Privacy-Preserving Learning: Differential privacy measures are incorporated during the evaluation phase. In addition to the normally trained pristine models, we have utilised OPACUS [18] to train models with differential privacy for propagation and evaluation to prevent privacy leakage. The evaluation scheduler coordinates the evaluation of individual model updates contributed by participants during each training round. Participants submit their differential privacy-enabled model updates, and the evaluation scheduler provides an incentive to the nodes who are participating in evaluating pending models. These security features safeguard data and model updates from unauthorised access, ensuring that sensitive information remains confidential throughout the DFL process.

SMPC Collaborative Model Aggregation: To make collaborative model aggregation possible while maintaining individual model privacy, SMPC protocols are integrated into the aggregation scheduler of the smart contract. SMPC enables nodes to jointly compute the aggregated model without revealing their respective model updates. Nodes securely collaborate to combine their encrypted model parameters, ensuring that the raw model parameters remain private throughout the aggregation process. By employing this innovative method, the blockchain-based DFL framework ensures a privacy-centric, secure, and collaborative environment for training language models while preserving user data confidentiality and fostering trust in the decentralised chatbot ecosystem. A detailed implementation of this key feature is explained next.

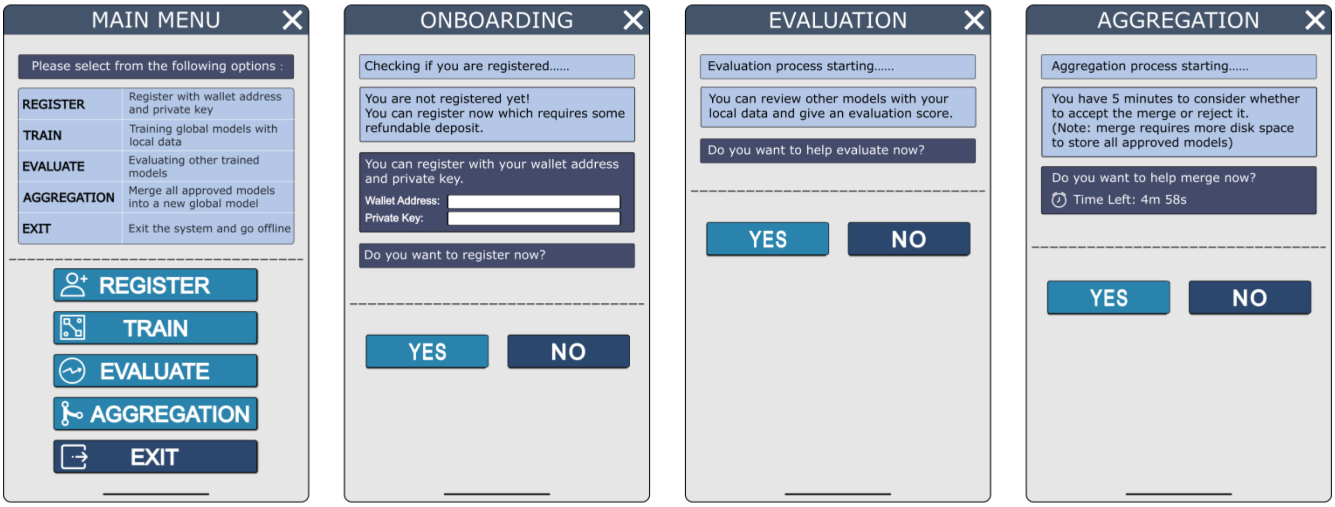
2.1 SMPC Implementation on the Ethereum Network

Figure 3 illustrates the SMPC implementation for the aggregation process. The aggregation scheduler securely combines these updates using SMPC, which allows individual model updates to be merged without revealing the model parameters, thus protecting the privacy of each participant's contribution. After the secure merging of model updates, the aggregation scheduler initiates the encryption of the final aggregated model. The aggregated model, which is encrypted for confidentiality, is stored in the IPFS, providing tamper-proof and immutable access for participants.

2.2 TinyLMs

TinyLMs are a viable solution in resource-constrained settings where deploying a large-scale language model (LLM) is impractical due to computational overhead. The process involves transforming an LLM into a more computationally feasible model while retaining its language processing capabilities through fine-tuning, distillation, and quantisation techniques.

Knowledge Distillation: Knowledge distillation is used to refine the TinyLM and compress it without a significant loss in performance. In this step, the fine-tuned LLM acts as a "teacher" model, providing soft target probabilities to guide the training of a smaller "student" model. The student model is trained to mimic the teacher's behaviour, thereby inheriting its language understanding capabilities. Through knowledge distillation, the student model effectively captures the essence of the LLM while reducing its size and complexity, resulting in a more lightweight TinyLM.



Please select from the following options :

REGISTER	Register with wallet address and private key
TRAIN	Training global models with local data
EVALUATE	Evaluating other trained models
AGGREGATION	Merge all approved models into a new global model
EXIT	Exit the system and go offline

Checking if you are registered......

You are not registered yet!
You can register now which requires some refundable deposit.

You can register with your wallet address and private key.

Wallet Address:
Private Key:

Do you want to register now?

Evaluation process starting......

You can review other models with your local data and give an evaluation score.

Do you want to help evaluate now?

Aggregation process starting......

You have 5 minutes to consider whether to accept the merge or reject it.
(Note: merge requires more disk space to store all approved models)

Do you want to help merge now?
Time Left: 4m 58s

Figure 4. User interface.

Quantisation for Model Pruning: The next step in generating TinyLM is quantisation, which reduces the model's computational requirements even further. Quantisation converts the model's high-precision weights to lower precision, such as 8-bit integers. This significantly reduces the model's memory footprint and computational cost, making it more feasible to deploy on resource-constrained devices. Although quantisation may result in some loss of precision, the impact on performance is often minimal, ensuring that TinyLM can still provide contextually relevant responses for chatbot applications.

Fine-tuning the LLM: The final step in creating a TinyLM is to fine-tune a pre-trained LLM on a domain-specific dataset. Fine-tuning the LLM allows the model to adapt to the target task by focussing on specific language patterns and contextual understanding relevant to the desired application [19]. This process helps tailor the LLM's vast language knowledge to the specific use case, making it more suitable for the intended application. A TinyLM can be generated from an LLM by fine-tuning, knowledge distillation, and quantisation. The TinyLM is tailored to a specific domain and pruned to a more computationally feasible size while retaining much of the language processing capabilities of its larger counterpart. This makes it well-suited for deployment in resource-constrained settings, enabling efficient and responsive chatbot interactions on a variety of devices with limited computational resources.

2.3 User Interface

To provide a seamless and interactive user experience within the decentralised chatbot ecosystem, we designed a user interface with four distinct modes of operation: Onboarding, Evaluation, Aggregation, and Exiting. These modes facilitate user engagement, incentivise active participation, and ensure timely model aggregation and updates (Figure 4).

In the Onboarding stage, new users can join the decentralised chatbot network as participants or nodes. During this phase, users register their devices with a small fee, which contributes to the FL process. As new nodes onboard the network, they receive instructions for training their local models and contributing to collective learning.

The Evaluation stage encourages users to actively participate in evaluating the pending model updates contributed by others. When a node submits its model update, it enters a validation queue for evaluation by other nodes. Nodes evaluate these updates based on performance metrics, model accuracy, and privacy compliance. Evaluators who provide valuable and accurate feedback receive incentives in the form of tokens or rewards. This incentivises nodes to actively contribute to the evaluation process, fostering a collaborative and transparent environment.

The Aggregation stage is responsible for combining the validated model updates from different nodes using SMPC. To ensure timely aggregation, a random node is selected as the designated aggregator. If the chosen aggregator does not respond within a predefined time frame, the system automatically selects another node to perform the aggregation process. This dynamic selection mechanism helps maintain the efficiency and continuity of the model aggregation process.

In the Exiting mode, nodes have the option to leave the decentralised chatbot network while preserving their privacy and data ownership. When a node decides to exit, its local model updates and data are securely deleted from its system. This ensures that participants can retain control over their data and contribute to the FL process as they see fit.

By incorporating these four modes of operation in the user interface, the decentralised chatbot ecosystem encourages active participation, rewards valuable contributions, and ensures seamless model aggregation. The user interface fosters a decentralised and democratic environment, where all participants play a crucial role in the collective improvement of the chatbot's language capabilities while safeguarding their privacy and data ownership.

3. Experimental Setup and Results

In this section, we present the experimental setup and results conducted to evaluate the performance and effectiveness of the proposed blockchain-based DFL framework, along with the generation of the TinyLM through fine-tuning, distillation, and quantisation processes. We deployed the DFL framework on the Ethereum blockchain using smart contracts to facilitate secure and privacy-preserving FL. Note that the DFL framework is agnostic to the model itself, and the experiments are to only showcase the efficacy of our DFL implementation.

We used the latest Falcon-7B LLM, which outperforms comparable open-source models (e.g., MPT-7B, StableLM, RedPajama, etc.) as witnessed on the OpenLLM leaderboard on the popular HuggingFace platform. It is a raw, pre-trained model, which should be further fine-tuned for most use cases. We chose the Open Orca-K16 dataset for our text summarisation task to fine-tune the LLM and the distilled model. The dataset contains pairs of input text and summary text.

For the generation of the TinyLM, we first employed knowledge distillation techniques [20] using the pre-trained LLM as the teacher model and a smaller but similar architecture as the student model on the Open Orca-K16 dataset. The student model was trained to mimic the teacher's behaviour by learning the value of its output logits, capturing its language understanding capabilities while reducing the model size significantly. Our code loads the test set, converts it into tokens, and then uses the ROUGE metric to evaluate the performance of the distilled model. The ROUGE precision score [21] (average of ROUGE-1, ROUGE-2, and ROUGEL scores) will provide an indication of how well the distilled model performs in comparison to the original teacher model. It is important to note that distilled models are typically expected to have slightly lower performance than their teacher models, but the student model is much faster and more efficient.

Quantisation was then applied to the distilled TinyLM to further compress the model's weights, achieving a more computationally feasible model without sacrificing performance (e.g., 8-bit integers instead of 32-bit floating-point numbers). In this work, we used relevant low-precision optimisers [22] and QLoRA [19] to obtain the fine-tuned quantised student model. Finally, this TinyLM is deployed using the DFL framework to give a further boost to the text summarisation performance on the Open Orca-K16 dataset.

3.1 DFL Evaluation Metrics

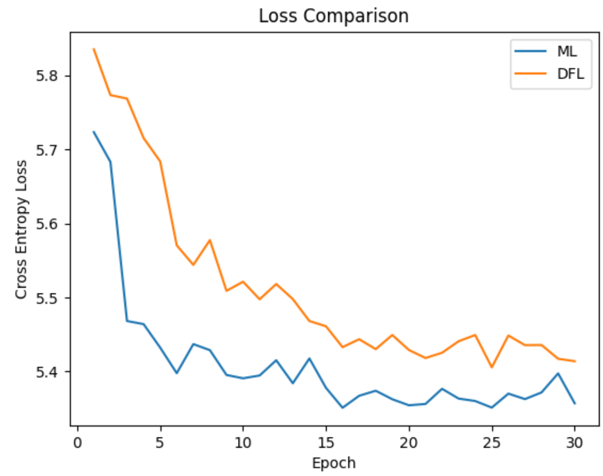
The distilled and quantised student TinyLM fine-tuning happens directly using the DFL framework. The TinyLM currently exhibits baseline proficiency in text summarisation tasks and a compact footprint that permits execution on standard end-user devices, such as a personal PC with a Graphics processing unit (GPU). The initial global model for five separate DFL nodes is established using the deployed TinyLM. These nodes possess private data for training, achieved by dividing the training dataset into five distinct segments, with each segment assigned to a respective node. Subsequently, each of these five nodes proceeds to conduct fine-tuning operations using its private data and subsequently evaluate the model's performance. The resulting fine-tuned private models are subjected to encrypted aggregation, facilitating the preservation of data privacy while also enabling parallel fine-tuning processes. This approach not only facilitates the utilisation of sensitive private data without compromising privacy but also harnesses individual computational resources to do the parallel training to expedite the model training process.

We evaluated the performance of the DFL framework using two metrics, including global model accuracy and training efficiency. The accuracy of the global model was assessed across multiple rounds of DFL using an independent test dataset and ultimately contrasted with the performance of a standalone TinyLM. This standalone TinyLM was trained normally without any FL, denoted as ML in the experiments, on the complete training dataset.

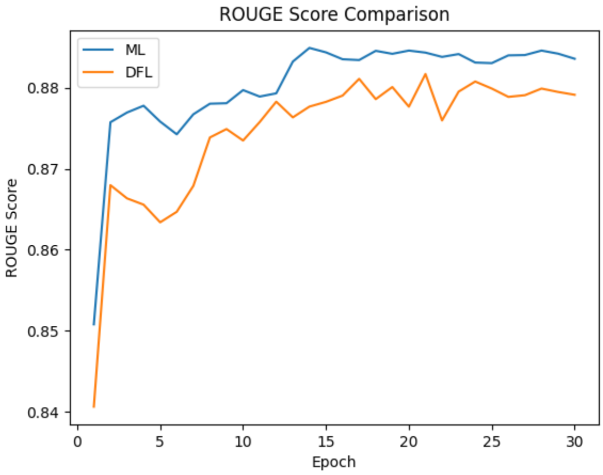
Secondly, as the DFL process splits the training burden across multiple nodes, it can potentially achieve faster convergence compared to a standalone ML model trained on the whole training dataset at a single machine. The DFL training efficiency was assessed by measuring the number of epochs needed to achieve the same performance using the TinyLM, with and without DFL, similar to the way the first metric is designed.

3.2 DFL Results

The size of the original Falcon-7b model was 14.43 GB, and the runtime GPU memory footprint was 26 GB. The size of the student model after knowledge distillation was just 700 MB, and the GPU memory footprint was 3.8 GB. After quantisation, the GPU memory footprint was further reduced to 3.0 GB. The final ROUGE scores of our TinyLM are: ROUGE-1-precision = 88.15%; ROUGE2-precision = 86.56%; ROUGE-L-precision = 88.06% using the first 20,000 samples from the Open Orca-K16 dataset. It is important to mention that we utilise a distinct portion of the dataset for quantisation and distillation. This decision is made to avoid redundancy in subsequent DFL processes. In summary, regarding the generation of the TinyLM, the fine-tuning and knowledge distillation processes yielded a student model that closely resembled the performance of the LLM while significantly reducing model size. The quantisation process further pruned the model, achieving a computationally efficient TinyLM suitable for deployment on resource-constrained devices. As mentioned earlier, the main contribution of the article is the DFL framework itself, and the model used is only for showcasing our successful implementation.



(a) Train/validation loss of the standalone ML model and DFL across the training rounds.



(b) The standalone ML and DFL model ROUGE precision scores on the test dataset across the epochs.

Figure 5. DFL results on the Open Orca-K16 dataset.

In our DFL experiments, we first filtered the dataset for very long input tokens and used part of the filtered dataset for training and the other part for testing with a ratio of 5:1 (this data was not the same one used for generating the TinyLM). We utilised 200,000 samples from the dataset for training and 40,000 samples for testing the standalone ML model and DFL. In our DFL experiments, we took a distribution of five nodes with an equal distribution of the training data. The ML training improved the ROUGE precision score of the TinyLM from 85.08% to 88.36%, while the DFL finally improved to 87.91%. Figure 5 depicts a comparison of the test loss change as well as the ROUGE precision score change, respectively. The results highlight the comparable training impact between DFL and standalone ML, a key outcome of this study. This underscores the successful functionality of our DFL implementation.

In addition to this, we also performed a comparison of the training efficiency of DFL and standalone ML. Under the premise of the same GPU (RTX A6000), each epoch of ML needs to process 200,000 samples, which takes about 55 minutes and 18.4 seconds; while a single node of DFL needs to process only 40,000 samples per epoch, which takes about 11 minutes and 9.3 seconds. Therefore, the time consumption is only 20.17% of that of ML, which is about the ratio of the total amount of data. Relevant to a real-world situation, since the GPUs used for DFL should theoretically have lower computational power than the GPUs used for ML, we arranged the DFL on a V100 GPU with lower computational power for the time computation. In this case, each epoch takes about 24 minutes and 30 seconds, which is only 44.30% of the time consumption of ML. This shows that DLF is more efficient compared to machine learning (ML) while having similar training capabilities.

The experimental results demonstrated the effectiveness of the DFL framework in collaboratively improving the language model. Through DFL, the chatbot's language capabilities were refined iteratively, resulting in enhanced model accuracy and contextually relevant responses. The results of our experiments highlight the potential of the proposed DFL framework in creating privacy-preserving and efficient TinyLMs. By leveraging blockchain technology and decentralised learning, the chatbot ecosystem can ensure user data privacy and foster trust among participants. Additionally, the generation of TinyLMs offers a practical solution for deploying language models on devices with limited computational resources, enabling efficient and responsive chatbot interactions in real-world scenarios.

4. Conclusion

Our article introduces a novel approach to address critical privacy and efficiency concerns in chatbot applications by harnessing the power of blockchain-based DFL and the generation of TinyLMs. Through the incorporation of SMPC within the Ethereum blockchain, we establish a secure and collaborative learning environment that preserves individual data privacy and fosters trust among participants. Additionally, by applying fine-tuning, knowledge distillation, and quantisation techniques, we successfully generated TinyLMs, significantly reducing the model size without compromising language processing capabilities. Our experiments demonstrate the efficacy of the proposed DFL framework and TinyLM generation, exhibiting similar model accuracy with much higher computational efficiency compared to standalone machine learning and real-world practicality. The combination of decentralised learning and lightweight language models introduces new possibilities for efficient chatbot deployments on resource-constrained devices, offering privacy-preserving and responsive language interactions in diverse domains. By integrating SMPC and encryption methodologies, our research advances the development of secure, efficient, and user-centric language processing applications, promising a future of decentralised chatbot technology that safeguards user privacy and empowers individuals to take control of their data.

Competing Interests:
None declared.

Ethical Approval:
Not applicable.

Author's Contribution:
BR is the main author for writing the manuscript and proofreading. SH collected the data and performed the experiments as supervised. XC co-supervised the project and provided timely feedback on a weekly basis.

Funding:
None declared.

Acknowledgements:
BR would like to acknowledge the efforts of Long Yinyun during the initial stages of the project in brainstorming sessions.

References

[1] G. Caldarini, S. Jaf, and K. McGarry, "A literature survey of recent advances in chatbots," *Information*, vol. 13, no. 1, p. 1: 41, 2022.

[2] A. Bahrini, M. Khamoshifar, H. Abbasimehr, R. J. Riggs, M. Esmaeili, R. M. Majdabadkohne, and M. Pasehvar, "Chatgpt: Applications, opportunities, and threats," in *2023 Systems and Information Engineering Design Symposium* (SIEDS), 2023, pp. 274–279.

[3] J. Homolak, "Opportunities and risks of chatgpt in medicine, science, and academic publishing: A modern promethean dilemma," *Croatian Medical Journal*, vol. 64, no. 1, p. 1, 2023.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[5] M. Liu, S. Ho, M. Wang, L. Gao, Y. Jin, and H. Zhang, "Federated learning meets natural language processing: A survey," arXiv preprint, 2021.

[6] L. Lyu, H. Yu, J. Zhao, and Q. Yang, *Threats to Federated Learning*. Cham: Springer International Publishing, 2020, pp. 3–16.

[7] H. Wu, B. Fang, C. Xiang, G. Cohen, A. van Schaik, and B. Ramesh, "Evolving neuromorphic systems on the ethereum smart contract platform," in *2022 IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain & Beyond (iGETblockchain)*. IEEE, 2022, pp. 1–6.

[8] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5926–5937, 2020.

[9] C. Li, Q. Shen, C. Xiang, and B. Ramesh, "A trustless federated framework for decentralized and confidential deep learning," in *2022 IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain & Beyond (iGETblockchain)*. IEEE, 2022, pp. 1–6.

[10] S. Aich, N. K. Sinai, S. Kumar, M. Ali, Y. R. Choi, M.-I. Joo, and H.-C. Kim, "Protecting personal healthcare record using blockchain & federated learning technologies," in *2022 24th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2022, pp. 109–112.

[11] Y. E. Oktian and S.-G. Lee, "Blockchain-based federated learning system: A survey on design choices," *Sensors*, vol. 23, no. 12, p. 5658, 2023.

[12] J. Benet, "Ipfs-content addressed, versioned, p2p file system," arXiv preprint arXiv:1407.3561, 2014.

[13] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[14] R.-H. Hsu, Y.-C. Wang, C.-I. Fan, B. Sun, T. Ban, T. Takahashi, T.-W. Wu, and S.-W. Kao, "A privacy-preserving federated learning system for android malware detection based on edge computing," in *2020 15th Asia Joint Conference on Information Security (AsiaJCIS)*, 2020, pp. 128–136.

[15] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, "Estimating the carbon footprint of bloom, a 176b parameter language model," arXiv preprint arXiv:2211.02001, 2022.

[16] X. Liu and Z. Liu, "Llms can understand encrypted prompt: Towards privacy-computing friendly transformers," arXiv preprint, 2023.

[17] P. Kaliamoorthi, A. Siddhant, E. Li, and M. Johnson, "Distilling large language models into tiny and effective students using pqrnn," arXiv preprint, 2021.

[18] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov, "Opacus: User-friendly differential privacy library in PyTorch," arXiv preprint arXiv:2109.12298, 2021.

[19] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," arXiv preprint arXiv:2305.14314, 2023.

[20] Y. Gu, L. Dong, F. Wei, and M. Huang, "Knowledge distillation of large language models," arXiv preprint arXiv:2306.08543, 2023.

[21] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004, pp. 74–81.

[22] T. Dettmers, M. Lewis, S. Shleifer, and L. Zettlemoyer, "8-bit optimizers via block-wise quantization," *9th International Conference on Learning Representations, ICLR*, 2022.