PureLLM: A Blockchain-Driven Decentralized PFL for Robust and Resource-Efficient Next-Gen LLMs

Md Tayeb Adnan, Paul Angelo Oroceo, Jae-Min Lee, and Dong-Seong Kim
Networked Systems Laboratory, Department of IT Convergence Engineering,
Kumoh National Institute of Technology, Gumi, South Korea.
(mdtayebadnan, oroceopaul, ljmpaul, dskim)@kumoh.ac.kr

2025 Sixteenth International Conference on Ubiquitous and Future Networks (ICUFN) | 979-8-3315-2487-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICUFN65838.2025.11169926

***Abstract*—Large language models (LLMs) trained on extensive publicly available datasets have achieved groundbreaking performance across diverse domains. However, both contemporary and future LLMs face four major challenges: (i) Data Availability, (ii) Data Bias and Quality, (iii) Computational Resources, and (iv) Data Privacy and Security. To address these challenges, we propose PureLLM, a blockchain-driven decentralized personalized federated learning (PFL) framework for the development of high-performance, resource-efficient next-generation LLMs. By leveraging decentralized federated learning, PureLLM enables collaborative training across diverse data sources without exposing sensitive user information. A blockchain-powered consensus mechanism ensures data integrity, security, and fairness, mitigating biases and enhancing trust in model updates. Additionally, our approach optimizes resource utilization by dynamically allocating computational workloads across distributed nodes, reducing energy consumption and improving scalability.**

Experimental evaluations demonstrate that PureLLM achieves superior model performance while maintaining strong privacy guarantees and reducing computational overhead. This framework paves the way for democratized and secure LLM training, fostering an inclusive AI ecosystem where individuals and organizations can contribute to and benefit from cutting-edge language models without compromising privacy or efficiency.

***Index Terms*—LLM, PureChain, Federated Learning, Decentralized**

I. Introduction

Large Language Models (LLMs) have revolutionized artificial intelligence (AI) with exceptional capabilities in natural language understanding, generation, translation, and code synthesis, impacting various industries. These models, trained on vast datasets, are central to modern AI systems, powering applications from virtual assistants to scientific discovery tools. However, building and deploying these models comes with significant challenges regarding sustainability, inclusivity, and ethical deployment.

A key issue is data availability. While LLMs need large, diverse datasets to function effectively, high-quality, domain-specific data is often scarce or siloed due to legal, organizational, or ethical barriers. Additionally, much of the accessible data is outdated or overused, hindering LLM improvement [3]. By 2030, publicly available data is expected to be insufficient to support continued LLM development, making private data essential.

Data owners are reluctant to share their private data due to privacy concerns, and while individuals can download models and train them on their private datasets, this leads to isolated models lacking interconnected learning. Thus, collaborative training frameworks that preserve privacy and promote shared learning are necessary for model scalability.

Data bias and quality are other critical concerns. LLMs trained on uncurated or imbalanced datasets may reinforce harmful stereotypes, misinformation, or marginalize minority voices, leading to unfair AI behavior. Additionally, the computational resources required to train these models, such as thousands of GPUs and massive energy consumption, concentrate development within a few large organizations, exacerbating technological divides.

Lastly, ensuring data privacy and security is vital, particularly in sensitive sectors like healthcare, finance, and education. Traditional centralized learning models increase the risk of data breaches [10], misuse, or regulatory non-compliance, underscoring the need for secure, decentralized learning approaches.

To overcome these multifaceted challenges, there is an urgent need for a paradigm shift—one that emphasizes decentralization, personalization, and trust. In this paper, we introduce PureLLM, a novel blockchain-driven decentralized Personalized Federated Learning (PFL) framework designed to train next-generation LLMs in a privacy-preserving, scalable, and energy-efficient manner. PureLLM combines the strengths of federated learning, which enables collaborative model training without sharing raw data, with the robustness and transparency of blockchain technology, which enforces trust, auditability, and tamper-proof model aggregation.

Unlike conventional federated learning, PureLLM introduces personalization layers that adapt the global model to local data distributions, significantly improving performance on edge devices and user-specific applications. Additionally, the integration of a blockchain-based consensus mechanism eliminates the need for centralized coordination and ensures secure, fair, and verifiable contributions from all participants. Resource-aware scheduling and dynamic workload balancing further optimize energy usage and enable deployment across a wide spectrum of devices, from high-end servers to mobile devices and IoT nodes.

Through extensive experimentation and benchmarking, we demonstrate that PureLLM not only matches or exceeds the performance of traditional centralized and federated LLMs but also achieves superior outcomes in terms of privacy preservation, computational efficiency, and resilience to ma-

licious or untrustworthy participants. The proposed architecture represents a significant step toward democratizing LLM development—unlocking the potential for widespread, secure, and inclusive AI innovation.

To summarize, our contributions are as follows:

- We propose PureLLM, a novel blockchain-driven decentralized Personalized Federated Learning (PFL) framework tailored for training next-generation LLMs while preserving data privacy, enhancing model personalization, and ensuring security.
- We integrate blockchain with federated learning to eliminate centralized control, leveraging smart contracts and decentralized consensus for fair, tamper-proof, and auditable model updates across a wide network of devices.
- We implement a resource-aware training pipeline that features dynamic workload distribution and energy-efficient computation across nodes.

II. RELATED WORK

A. Large Language Models

Large language models (LLMs), including GPT-3.5/4 [5], [14] and Llama2 [14], have achieved notable success across a variety of domains [4], [6], [8], [11]–[13], [15]. These models are typically trained through a three-phase process: (1) auto-regressive pre-training on extensive datasets like C4 and Pile , allowing LLMs to acquire broad general knowledge of the world ; (2) instruction tuning on pairs of instructions and corresponding responses, enabling LLMs to follow specific directives ; and (3) value alignment, where human or AI-annotated preference datasets are used to integrate human values into the model [2, 3]. At present, these stages primarily utilize publicly available data, either released by the public [2, 3] or generated by AI. However, experts predict that high-quality public datasets will be depleted by 2026 [11], signaling a potential bottleneck for existing LLMs, as larger datasets generally enhance model performance. In response, recent efforts have focused on training LLMs with large-scale, privately-held datasets [4, 5]. For instance, BloombergGPT, trained on 40 years of financial data, has exhibited strong performance in finance. Nonetheless, in real-world scenarios, the data available to individual entities may be limited, whereas pooling data from multiple parties could create a substantial dataset for training a robust LLM . As a result, the future development of LLMs must consider collaborative training on distributed private data, ensuring privacy protection.

B. Federated Learning

Federated learning (FL) offers significant potential for enabling privacy-preserving collaborative model training. FL allows multiple parties (clients) to jointly train a shared global model without the need to exchange raw data, with coordination by a central server. The typical FL process includes four steps: broadcasting the global model from the server to the clients, training the local model at the client, uploading the local model to the server, and updating the global model through aggregation on the server.

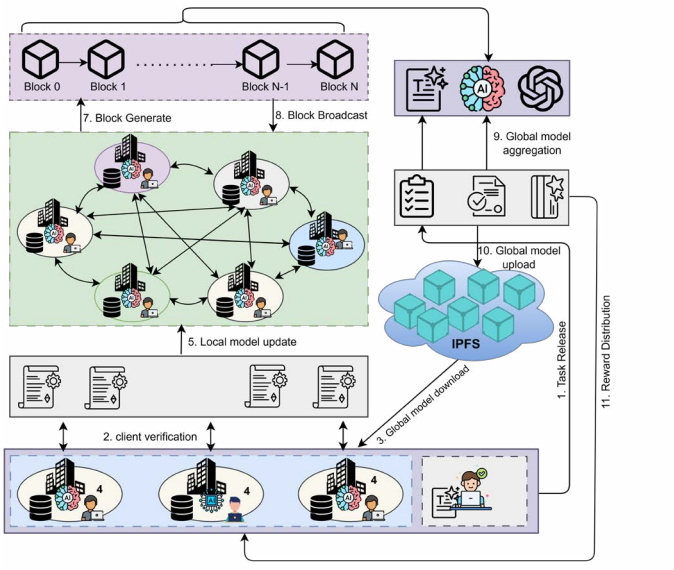

Fig. 1. Overview of the proposed PureLLM

The standard FL approach, FedAvg , has been shown to provide only moderate performance, particularly when dealing with data heterogeneity. As a result, several FL algorithms have been developed to improve FL performance. (1) On the client side, various methods have been proposed to enhance consistency across local models, which in turn improves the aggregated model's performance. For instance, FedProx aims to regularize the distance between the local and global models, while SCAFFOLD uses control variates to correct gradients of local models. (2) On the server side, several methods focus on refining the aggregation process to enhance the global model's performance . FedAvgM and FedOPT incorporate momentum into the global model update, while FedNova and FedDisco adjust the weights used for aggregating local models.

While the performance of these methods has primarily been evaluated in the context of image classification and small models, their effectiveness in the training of current LLMs remains unclear. This paper aims to be the first to explore the application of these methods in LLM training, offering new insights and identifying the most suitable techniques for federated LLM training.

C. Federated Learning and Large Language Models

Federated learning (FL) presents a promising solution to the challenge of dwindling public data resources for training large language models (LLMs). By allowing multiple participants to collaboratively train a model without sharing raw data, federated LLMs can leverage a broader, more diverse range of datasets [9], [16]. Chen et al. introduce the concept of federated LLMs, which encompasses federated pretraining, fine-tuning, and prompt engineering, and discuss the unique challenges and potential strategies for its implementation, highlighting the advantages of federated learning over traditional LLM training methods. Gupta et al. present FILM, a novel attack methodology for federated learning of language

models. This study is the first to demonstrate the feasibility of recovering text data from large batch sizes, evaluating various defense strategies, and suggesting new avenues for improving privacy in language model training. Additionally, a recent paper [4] proposed an industrial federated learning framework aimed at optimizing the training of large LLMs, addressing the dual challenges of computational resource constraints and data privacy concerns [1]. The LP-FL methodology utilizes Low-Rank Adaptation (LoRA) to reduce model parameters within federated learning, facilitating effective local fine-tuning and global model federation. FederatedScope-LLM (FS-LLM) offers a comprehensive framework for optimizing LLMs within a network, covering everything from data preparation to outcome assessment. Despite these advancements, the frameworks still face limitations, particularly regarding the transparency of LLM training processes [2]. To address this, introduced an automated data quality control pipeline that uses data valuation algorithms to assess and improve the quality of training samples across collaborative platforms, ensuring enhanced model performance while safeguarding data privacy. Further challenges arise in the wireless domain, where highlights privacy concerns, inefficient data handling, and high communication costs, demonstrating the effectiveness of their methods through simulations [7]. Ro et al. present scale-invariant modifications to the Coupled Input Forget Gate (CIFG) and transformer models, which significantly enhance federated learning performance by improving convergence speed and balancing privacy-utility trade-offs In recent studies, several preliminary papers have explored federated learning in the context of LLMs, with some offering position papers but lacking empirical results. FATE-LLM investigates federated fine-tuning for LLMs but focuses only on conventional tasks like advertisement generation, rather than more complex processes such as instruction tuning or value alignment. Both FederatedScope-LLM and Shepherd focus on federated instruction tuning but face several limitations. First, their empirical results are insufficient due to relatively limited training and evaluation datasets (e.g., Shepherd [58] uses only one training and one evaluation dataset). Second, neither of them addresses the critical step of value alignment, which is essential for deploying contemporary chatbots [3]. Third, both rely solely on the FedAvg method for federated learning, ignoring other FL algorithms that may offer better performance depending on the task. In contrast, this paper provides the most comprehensive exploration of FL and contemporary LLMs to date.

III. Methodology

A. Overview

This framework utilizes blockchain technology to enable decentralized, privacy-preserving federated learning (PFL) for training large language models (LLMs). Clients register securely through a blockchain-based process using JSON Web Tokens (JWTs). The task publisher releases federated learning tasks via smart contracts, specifying essential parameters like data size, accuracy, and rewards. Clients participate by training models locally, updating parameters, and securely sharing them for aggregation. Blockchain ensures transparency, security, and decentralized verification, maintaining the integrity of the global model while preserving client data privacy. Figure 1 illustrates the overview of our proposed PureLLM framework.

B. Workflow

1) Client Register: In the proposed framework, each client must first complete an enrollment process within the blockchain network. Algorithm 1 outlines a simple and efficient method for registering a client, leveraging a unique identifier, and securing communication through a JSON Web Token (JWT). The process begins by checking if the client's unique identifier (Cid) already exists in the user pool (Upool). If the identifier is found, the registration is terminated, signaling that the client is already registered. If not, the algorithm proceeds by generating a public-private key pair using the keyGenerator() function.

With the keys generated, a JWT is created for the client, which, along with the client's ID, is securely stored to complete the registration. The user pool is then updated to include the newly registered client ID, confirming the success of the registration. Finally, the algorithm returns a success status along with the generated JWT, indicating the client's successful registration and providing them with a secure token for all future interactions. This approach ensures a robust and secure client registration system by utilizing cryptographic keys and JWTs, offering both simplicity and high-level security in client management.

Algorithm 1 Client Registration Algorithm

Require: Client ID (Cid), Key Generator function (`keyGenerator()`), JWT Generator function (`generateJWT()`)
Ensure: Registration Success status ($RegisterSuccess$), JWT Token

$RegisterSuccess \leftarrow$ False
if $Cid \in$ Upool **then**
  return "Client ID already exists."
end if
$(Pk, Sk) \leftarrow$ keyGenerator()
$jwt \leftarrow$ generateJWT(Pk, Sk)
$Cid \leftarrow jwt \leftarrow SC$
Upool $\leftarrow$ Upool $\cup\, Cid$
$RegisterSuccess \leftarrow$ True
return $RegisterSuccess, jwt$

2) Task Release: the task publisher announces the federated learning task to the entire network of potential clients. This is achieved by invoking a task-publishing smart contract, which records all relevant task information on the blockchain. Key task requirements, such as the data size, desired training accuracy, model aggregation algorithm, and reward structure, are also specified in this step. The task release provides the necessary context for the entire federated learning operation

and ensures that all clients are aware of the task's scope and conditions.

3) Participant Client Selection: Once the task is released, the task publisher selects clients based on their available resources and willingness to participate. Clients with sufficient computational resources and low training costs are prioritized. The selection process ensures that only capable participants are chosen, optimizing the overall efficiency and effectiveness of the federated learning process.

4) Global Model Download: Upon selection, each participating client downloads the global model from the blockchain. In the first round of training, the global model consists of the initial parameters recorded by the task publisher. For subsequent rounds, the global model downloaded by clients is updated based on the aggregation of local model parameters contributed by the clients in the previous training round. This continuous update of the global model ensures the model evolves progressively with each iteration.

5) Local Model Training: Once clients download the global model, they train it on their local data, updating the model parameters. This local training allows each client to adapt the model to its specific data without sharing raw information, ensuring privacy. After completing local training, clients upload the updated model parameters for verification and aggregation, contributing to the improvement of the global model.

6) Local Model Upload: Once local model training is completed, the client uploads the updated model parameters to the blockchain. These model parameters, reflecting the learning performed on the client's local data, are stored in the blockchain for future aggregation.

7) Model Verification: To maintain the integrity of the system, blockchain nodes perform a verification process. This step ensures that the uploaded local model parameters are legitimate and reliable. By verifying the model updates, the blockchain nodes prevent the incorporation of faulty or unreliable model updates that could negatively impact the global model's accuracy and stability.

8) Block Generation: After the verification of local model parameters, blockchain nodes with bookkeeping rights proceed to generate a candidate block. This block contains all verified local model updates. The nodes obtain the right to generate the block through a consensus algorithm, which ensures that the process is decentralized and secure. The generation of the block records the local model parameters in the blockchain, providing an immutable ledger of updates.

9) Block Broadcast: Once a node has generated the candidate block, it broadcasts the block to other nodes in the network. These nodes verify the legitimacy of the block and the model parameters contained within it. If the block passes the verification of the majority of nodes, it is added to the respective local ledgers of the nodes. This ensures that all participants in the blockchain network have access to the updated block and that the global model remains consistent across all clients.

10) Global Model Aggregation: Following the block broadcast, the task publisher invokes the model aggregation smart contract to aggregate the model parameters stored in the verified block. This aggregation process combines the individual updates into a unified global model, which is then recorded in the blockchain. The resulting global model reflects the collective learning of all participating clients and serves as the basis for the next round of model training.

11) Reward Distribution: The final phase of the workflow involves the automatic distribution of rewards to the participating clients. The task publisher invokes the reward distribution smart contract, which allocates rewards based on each client's contribution to the federated learning task. In addition, blockchain nodes that participated in verifying and broadcasting the block also receive corresponding rewards. This incentivizes participants to engage actively in the learning process and helps sustain the federated learning ecosystem.

Algorithm 2 PureLLM Workflow

Require: TaskPublisher, Clients, Blockchain, task_info
Ensure: UploadSuccess, TrainingProcess
UploadSuccess, TrainingProcess = False
TaskPublisher broadcasts task and records on Blockchain (task_info)
for each Client in Clients **do**
  TaskPublisher selects Client based on resources and cost
  Client downloads global model from the Blockchain
  for epoch = 1 to n **do**
    Client trains local model using private dataset
    Client uploads local model parameters to Blockchain
    Blockchain nodes verify the Client's model parameters
    if Client's identity is valid **then**
      Blockchain stores the Client's model parameters in a block
    else
      return Client identity check failed
    end if
  end for
end for
TaskPublisher aggregates all Client model parameters into global model
TaskPublisher stores the global model in the Blockchain
TaskPublisher distributes rewards to Clients and Blockchain nodes
UploadSuccess = True
TrainingProcess = True
return UploadSuccess, TrainingProcess

IV. Privacy and Security Analysis

Our blockchain-based decentralized personalized federated learning framework for Large Language Models (LLMs) addresses privacy and security concerns. By keeping data local and utilizing blockchain's transparency and immutability, our approach ensures secure, privacy-preserving model training with personalized updates while maintaining data integrity.

A. Privacy Analysis

Our blockchain-based decentralized personalized federated learning framework enhances privacy by decentralizing data storage and ensuring that data remains under the control of individual participants. In federated learning, each client i maintains its own dataset D_i, and only model parameters θ_i are shared, not raw data. The global model θ_g is updated iteratively without revealing the local data and can be represented as:

$$\theta_g = \sum_{i=1}^{k} p_i \theta_i$$

where θ_g represents the global model, θ_i is the local model of client i, and p_i is the weight of client i's data in the global model. This ensures that the data remains private while contributing to the optimization of the global model.

Blockchain further strengthens privacy by providing an immutable and transparent record of model updates and transactions. Every interaction, including the uploading of model updates, is recorded on the blockchain, ensuring that unauthorized access or modification of data is prevented. Smart contracts automate privacy policies, ensuring that all participants adhere to predefined terms, thus reducing the risk of human error or malicious behavior.

B. Security Analysis

Security in our framework is significantly enhanced through blockchain's inherent features, including its immutability and distributed consensus mechanisms. In a typical blockchain network with proof-of-work (PoW) or other consensus protocols, security can be analyzed in terms of the game-theoretic framework:

$$\max_{\text{honest nodes}} [\text{reward}] \quad \text{vs} \quad \max_{\text{attackers}} [\text{deviation from protocol}]$$

where the Nash equilibrium ensures that no party can improve their outcome by unilaterally changing their strategy. This makes it computationally infeasible for attackers to tamper with the blockchain once a majority of honest nodes participate.

Additionally, cryptographic techniques such as digital signatures and secure hash functions (e.g., SHA-256) are used to verify the authenticity and integrity of the model updates. The model update θ_i of each client is cryptographically signed to prevent unauthorized modifications:

$$\text{Signature}(\theta_i, \text{private key}) \rightarrow \text{validity check}$$

Moreover, the use of smart contracts enforces predefined rules and conditions on the blockchain, ensuring that participants only execute authorized operations. This reduces the chances of security breaches caused by human error or malicious actions.

The decentralized nature of blockchain eliminates single points of failure, making it difficult for attackers to control the system. A successful 51% attack, where an attacker controls more than half of the network's computational power, is practically infeasible as the number of honest nodes increases exponentially. This ensures that the system remains secure even as the network scales.

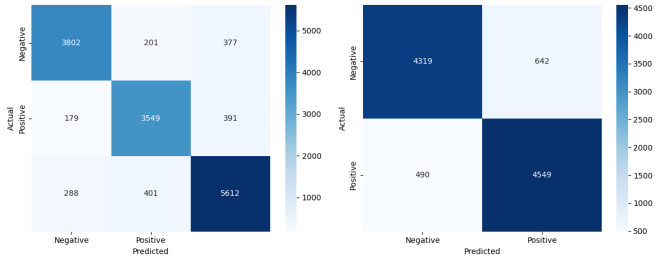

Fig. 2. Confusion matrix of two datasets: Twitter and IMDB

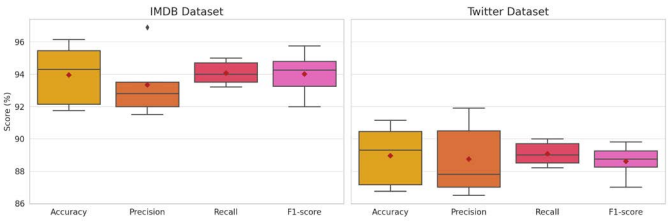

Fig. 3. Performance Distribution across LoRA Configurations

V. PERFORMANCE EVALUATION

A. Dataset

For our experiments, we employed two datasets: the IMDB dataset and the Twitter dataset. The IMDB dataset is a well-known benchmark in sentiment analysis, widely used for evaluating text classification models. With a total of 50,000 reviews, it is large enough to represent a realistic federated learning scenario, yet still manageable for experimental purposes. On the other hand, the Twitter dataset consists of a substantial collection of tweets, offering a diverse set of real-world social media data. This dataset provides an ample sample size, enabling us to assess the scalability and efficiency of the proposed framework effectively. Together, these datasets allow us to test the robustness and applicability of our approach in both controlled and real-world scenarios.

B. Result Analysis

In this section, we examine the effectiveness of our method in terms of accuracy reduction, comparing performance across different configurations. Our experiments were conducted using various configurations of the LoRA method on both the IMDB and Twitter datasets. While ensuring high levels of security and privacy preservation, we also prioritized the effectiveness of the model's performance.

We first evaluated the IMDB dataset using different LoRA configurations. The "Retrain from Scratch" method was used as a baseline to assess the impact of our approach. A summary of the specific LoRA configurations utilized in our experiments

is provided in Tables I and II. The confusion matrix and box plot of both are illustrated in Figure 2 and 3, respectively.

In addition to security and privacy considerations, our analysis focuses on understanding the performance variations and identifying why certain configurations performed better or worse compared to the baseline. By striking a balance between model performance and privacy preservation, we aimed to demonstrate that privacy-enhancing techniques can be effectively integrated without sacrificing the quality of the model's predictions.

TABLE I
EXPERIMENTAL RESULTS ON IMDB DATA

LoRA Config	Accuracy	Precision	Recall	F1-score
r=1, alpha=2, dropout=0.4	95.45%	96.90%	94.70%	94.80%
r=4, alpha=1, dropout=0.2	92.15%	91.50%	95.00%	94.25%
r=8, alpha=4, dropout=0.3	96.15%	92.00%	93.50%	95.75%
r=16, alpha=2, dropout=0.2	91.75%	95.50%	94.00%	93.25%
r=32, alpha=4, dropout=0.1	94.30%	92.80%	93.20%	92.00%

TABLE II
EXPERIMENTAL RESULTS ON TWITTER DATA

LoRA Config	Accuracy	Precision	Recall	F1-score
r=1, alpha=2, dropout=0.4	90.45%	91.90%	89.70%	89.80%
r=4, alpha=1, dropout=0.2	87.15%	86.50%	90.00%	89.25%
r=8, alpha=4, dropout=0.3	91.15%	87.00%	88.50%	85.75%
r=16, alpha=2, dropout=0.2	86.75%	90.50%	89.00%	88.25%
r=32, alpha=4, dropout=0.1	89.30%	87.80%	88.20%	87.00%

VI. CONCLUSION

In this paper, we introduced PureLLM, a blockchain-driven decentralized private federated learning (PFL) framework designed to enhance the performance and resource efficiency of next-generation large language models (LLMs). By leveraging blockchain technology, PureLLM ensures privacy and security while enabling collaborative model training across distributed parties. Our experiments demonstrated that PureLLM achieves high performance and resource efficiency without compromising the privacy of the training data. This work opens new avenues for scalable and secure LLM training, with potential applications in various industries requiring both high-performance models and stringent data privacy considerations. Future research will focus on optimizing the framework for even greater scalability and investigating its applicability to other domains beyond LLMs.

ACKNOWLEDGMENT

This work was partly supported by Innovative Human Re- source Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201612, 33%) and by Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 33%) and by the MSIT, Korea, under the ITRC support program(IITP-2025-RS-2024-00438430, 34%)

REFERENCES

[1] Usama Arshad and Zahid Halim. Blockllm: A futuristic llm-based decentralized vehicular network architecture for secure communications. *Computers and Electrical Engineering*, 123:110027, 2025.

[2] Mouhamed Amine Bouchiha, Quentin Telnoff, Souhail Bakkali, Ronan Champagnat, Mourad Rabah, Mickaël Coustaty, and Yacine Ghamri-Doudane. Llmchain: Blockchain-based reputation system for sharing and evaluating large language models. In *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 439–448, 2024.

[3] Haotian Dong, Jingyan Jiang, Rongwei Lu, Jiajun Luo, Jiajun Song, Bowen Li, Ying Shen, and Zhi Wang. Beyond a single ai cluster: A survey of decentralized llm training. *arXiv preprint arXiv:2503.11023*, 2025.

[4] Mohtasin Golam, Adnan Md Tayeb, Mst Ayesha Khatun, Md Facklasur Rahaman, Ali Aouto, Oroceo Paul Angelo, Dong-Seong Kim, Jae-Min Lee, and Jung-Hyeon Kim. Blackicenet: Explainable ai-enhanced multimodal for black ice detection to prevent accident in intelligent vehicles. *IEEE Internet of Things Journal*, pages 1–1, 2025.

[5] Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023.

[6] Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLOS Digital Health*, 2(2):e0000198, 2023.

[7] Haoxiang Luo, Jian Luo, and Athanasios V. Vasilakos. Bc4llm: A perspective of trusted artificial intelligence when blockchain meets large language models. *Neurocomputing*, 599:128089, 2024.

[8] Adnan Md Tayeb and Tae-Hyong Kim. Unestformer: Enhancing decoders and skip connections with nested transformers for medical image segmentation. *IEEE Access*, 12:190996–191009, 2024.

[9] Francesco Piccialli, Diletta Chiaro, Pian Qi, Valerio Bellandi, and Ernesto Damiani. Federated and edge learning for large language models. *Information Fusion*, 117:102840, 2025.

[10] Adnan Md Tayeb, Md Mahinur Alam, Mst Ayesha Khatun, Mohtasin Golam, Md Facklasur Rahaman, Md Javed Ahmed Shanto, and Dong-Seong Kim. Pureparking: A decentralized, secure framework for parking space sharing using blockchain. In *Proceedings of the Korea Institute of Information and Communications Sciences Conference*, Jeju, South Korea, June 2024.

[11] Adnan Md Tayeb, Jae-Min Lee, and Dong-Seong Kim. Hiclassgen: High-resolution image augmentation with class and shape controllable diffusion models. In *2025 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 0814–0819, 2025.

[12] Adnan Md Tayeb, Hope Leticia Nakayiza, Heejae Shin, Seungmin Lee, Chaesoo Lee, YeongHun Lee, Dong-Seong Kim, and Jae-Min Lee. Defectgen: Few-shot defect image generation using stable diffusion for steel surface analysis. In *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 2087–2092, 2024.

[13] Adnan Md Tayeb, Hope Leticia Nakayiza, Heejae Shin, Seungmin Lee, Jae-Min Lee, and Dong-Seong Kim. Defectdiffusion: A generative diffusion model for robust data augmentation in industrial defect detection. In *2025 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 0066–0071, 2025.

[14] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[15] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

[16] Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6137–6147, New York, NY, USA, 2024. Association for Computing Machinery.