IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 March 2024, accepted 31 May 2024, date of publication 7 June 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3411091

RESEARCH ARTICLE

PureFed: An Efficient Collaborative and Trustworthy Federated Learning Framework Based on Blockchain Network

MADE ADI PARAMARTHA PUTRA[1], **(Member, IEEE),**
NYOMAN BOGI ADITYA KARNA[2], **(Senior Member, IEEE), REVIN NAUFAL ALIEF**[3],
AHMAD ZAINUDIN[4], **(Member, IEEE), DONG-SEONG KIM**[3], **(Senior Member, IEEE),**
JAE-MIN LEE[3], **(Member, IEEE), AND GABRIEL AVELINO SAMPEDRO**[5], **(Member, IEEE)**

[1]Faculty of Information Technology and Design, Primakara University, Denpasar 80226, Indonesia
[2]School of Electrical Engineering, Telkom University, Bandung 40257, Indonesia
[3]Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea
[4]Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea
[5]School of Management and Information Technology, De La Salle-College of Saint Benilde, Manila 1004, Philippines

Corresponding author: Jae-Min Lee (ljmpaul@kumoh.ac.kr)

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Institute for Information & Communication Technology Promotion grant funded by the Korea government through the Ministry of Science and ICT (IITP-2024-2020-0-01612, 50%) and by Priority Research Centers Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003, 50%).

ABSTRACT This paper introduces PureFed, an innovative Federated Learning (FL) framework designed for efficiency, collaboration, and trustworthiness. In the background of FL research, it was observed that previous frameworks often neglected participant privacy, a critical aspect not aligned with the core FL concept. Additionally, there was room for improving the efficiency of existing frameworks. PureFed addresses these shortcomings by offering participants the flexibility to initiate FL tasks or join existing ones without sharing any private data and removing unnecessary actions that led to an inefficient system. Leveraging blockchain technology, it employs smart contracts to ensure traceability and immutability, enhancing the security of the framework. Additionally, PureFed employs symmetric key encryption and dual digital signature mechanisms using ECDSA to guarantee the confidentiality and integrity of shared models. To expedite model convergence, PureFed incorporates a dynamic aggregation scheme, selecting the most suitable model from three distinct techniques: FedAvg, accuracy-based, and loss-based. Furthermore, the framework introduces a dynamic incentive and punishment mechanism to incentivize collaboration and maintain trust. Extensive performance evaluations reveal PureFed's significant advantages. It outperforms its counterparts by 63.39% and 67.72% in terms of smart contract deployment and interaction gas costs, respectively. Lastly, scalability analyses indicate PureFed's ability to adapt efficiently, achieving target accuracy in fewer rounds.

INDEX TERMS Blockchain, collaborative federated learning, incentive mechanism, smart contracts, trustworthiness.

I. INTRODUCTION

Massive data transfers between various devices have prompted industries to harness vast amounts of information, transforming it to develop intelligent systems capable of learning and making precise decisions based on the provided data [1]. This process enhances overall system performance, efficiency, and capabilities. These intelligent systems are often associated with Artificial Intelligence (AI), which mimics the concept of human cells in processing information

The associate editor coordinating the review of this manuscript and approving it for publication was Mueen Uddin.

and producing outputs [2]. AI can effectively determine various conditions based on the training data used to build the model. Typically, larger datasets result in more robust models with better performance compared to models trained with limited data [3]. Despite the significant volume of data transferred between devices, entities capable of collecting this information are limited. As a result, the data becomes extremely valuable and cannot be shared to protect user privacy or for the sake of industry or individual profits [4].

Traditional AI models are typically trained in a centralized manner, where all information from numerous participants is stored on a single central server. This server then conducts training based on the gathered data. After the training is completed, the model parameters are sent to all participants. However, this approach is inefficient, as some entities have concerns about sharing their private information [5]. To address this issue, Federated Learning (FL) has been introduced to train models in a distributed manner [6]. FL requires participants to perform local training using their own computing resources without sending any sensitive information to the server. This approach ensures user data privacy and reduces communication costs.

FL supports decentralized learning, preserving user privacy and reducing communication overhead. It has found applications in environment-specific scenarios aimed at improving system performance. This improvement encompasses aspects such as model training and distribution across multiple devices. For example, in vehicular networks, FL can be employed to detect misbehavior in intelligent transportation systems, helping to combat Sybil attackers [7]. In terms of smart additive manufacturing, FL can be employed to enhance fault detection accuracy through collaboration among multiple manufacturers. Each manufacturer can train the model using their data without sharing the actual dataset with the FL server. In our previous work, we introduced hierarchical federated transfer learning (HFTL) to enhance the efficiency of the FL system [8]. Similarly, for medical purposes, FL can be utilized to perform distributed training across different hospitals without sharing hospital or patients' private information. This approach is effective for continuously improving model performance, such as determining COVID-19 disease based on a patient's lung condition or any other medical image [9].

In terms of data sharing permissions, FL can be categorized into two techniques: silo and cross-silo. Most FL systems operate under the silo mechanism due to data privacy concerns. In silo FL, participants' data is exclusively used to train their local models without sharing their data. In contrast, in cross-silo FL, data can be forwarded to the FL server to enable collective learning and create a more accurate global model [10]. In this paper, we focus on silo FL to ensure the security of participants' data. On one hand, silo FL has been optimized for specific use cases and environments, as previously described, to achieve faster model convergence among all participants. Typically, silo FL participants are determined before the FL process begins. Not every individual or organization can join the FL process since collaborative learning is designed to be private. On the other hand, limiting the number of silo FL participants might lead to longer convergence times, especially if the data distribution among participants is non-independent and not identically distributed (non-IID) [11].

Furthermore, the current FL framework tasks are initiated by the aggregator, which is the FL server, to perform specific tasks. This restricts FL participants from initiating tasks, limiting participation, reducing innovation, and ultimately inhibiting experimentation that could lead to novel solutions. Moreover, anyone should have the ability to initiate and participate in collaborative learning, with the system responsible for ensuring that participants meet the requirements such as dataset availability, computing power, or reputation. Based on these conditions, a framework that allows for task creation and participation by any individual is required. The system should permit anyone to start FL tasks without limitations. While a marketplace for collaborative AI model training has been discussed previously [12], the design and requirements of the existing system do not align with the FL concept, as the dataset is shared among participants. Additionally, the communication overhead increases as the total number of participants grows.

Therefore, a FL framework that supports collaborative learning for any individual is needed. It should not only support traditional FL processes but also ensure that every FL process is executed efficiently and recorded accurately for the sake of trustworthiness. To the best of our knowledge, this is the first work that considers the collaborative FL process, allowing anyone to join distributed training using blockchain networks and smart contract to ensure the trustworthiness of the framework. The major contributions of this paper are summarized as follows:

1) We propose an efficient, collaborative, and trustworthy FL framework named PureFed, which acts as a marketplace for distributed AI model training. PureFed allows any individual to create a task or participate in existing tasks. The framework utilizes blockchain technology due to its immutability, which is accessed via a smart contract to ensure the trustworthiness of all participants by recording every action performed in the FL process. The proposed PureFed framework consists of: (i) Task Publisher, (ii) Task Allocator, (iii) Task Validator, and (iv) Decision Controller.
2) We have designed a task allocator that adaptively assigns a participant as a worker and validator for specific tasks. The task allocator takes into account participants' reputations and the size of their datasets.
3) We have developed a secure model parameter transfer method between all participants and the FL server using a dual signature mechanism. Initially, the Advanced Encryption System (AES) symmetric keys of all participants are stored in the blockchain network. Then, the

Elliptic Curve Digital Signature Algorithm (ECDSA) is used to sign the encrypted model parameters before they are stored in the blockchain network. The second signature is added by the validator if the worker model is validated.

4) We have designed a dynamic aggregation scheme that assesses the validated model using three distinct conditions in each federation round: the traditional approach, average accuracy, and average loss. This approach ensures the performance of the new global model.
5) We propose a dynamic incentive mechanism to attract additional participants to join the PureFed framework. The incentive mechanism is determined based on each participant's contribution for each federation round. Additionally, a punishment mechanism is addressed to prevent participants from providing malicious or fake parameters that could affect the global model aggregation.
6) We conducted an extensive performance evaluation of the PureFed framework, considering accuracy, precision, recall, and F1-score, using different aggregation techniques and varying numbers of participants with non-IID data distribution. Furthermore, we discuss the impact of malicious participants and provide insights into blockchain-related metrics such as gas cost and transaction time.

The remainder of the article is structured as follows. Section II describes previous work in FL optimization and AI-based collaboration frameworks. Section III provides a detailed overview of the proposed PureFed system, including dual signature mechanisms, dynamic parameter aggregation schemes, and reputation mechanisms. In Section IV, we discuss simulation scenarios and performance evaluation. Finally, Section V concludes this work and outlines future research directions.

II. RELATED WORK

Generally speaking, an FL system can be regarded as a collaborative learning process involving participants and the FL server, aiming to achieve improved model performance. However, many FL systems have been traditionally designed for close collaboration, which can limit their potential for broader applications. This section delves into the state-of-the-art in FL optimization, followed by an exploration of collaborative FL that leverages the blockchain network.

A. FEDERATED LEARNING OPTIMIZATION

Originating from the first FL system introduced by McMahan et al. in [6], the adoption of FL has spread widely. Optimization for FL can be categorized into several aspects, including client selection, aggregation techniques, and security enhancements. In our previous work [13], we introduced a client selection mechanism that selects the best-performing model among all participants, resulting in faster convergence times compared to traditional client selection methods. Regarding aggregation techniques, Zhao et al. introduced a secure and efficient aggregation method designed to handle byzantine failures. They also proposed a sampling method to strike a balance between efficiency and performance [14]. Additionally, different model aggregation methods have been explored, including layer-wise approaches [15] and considering the contributions of FL participants [16].

In terms of security enhancements, the adoption of differential privacy and blockchain technology has led to more secure FL systems. For instance, authors in [17] introduced differential privacy in FL for medical image analysis by adding Gaussian noise to the parameters before forwarding them to the FL server. Furthermore, research has investigated the performance trade-offs when utilizing local differential privacy, taking into account factors such as privacy, utility, and communication [18]. Lastly, the use of blockchain for FL was covered in our previous work, which employed a layer 2 network to achieve faster transaction speeds compared to blockchain layer 1 networks [19].

B. BLOCKCHAIN-BASED FL COLLABORATION

As an advancement in security mechanisms, blockchain has been adopted to enhance the security of the FL process by uploading model parameters to the blockchain instead of forwarding them directly to the FL server [20]. For example, ChainFL proposed a mechanism to store model parameters in an Ethereum-based blockchain via smart contracts. The presented results show that the FL approach may reduce model performance as the dataset is divided among multiple participants, compared to centralized model learning [21]. Despite the trade-off in accuracy, the preservation of participant data remains a major concern in FL systems. FL can also be categorized into two types based on the aggregation process: synchronous and asynchronous. The authors in [22] introduce a technique to dynamically adjust the FL system based on local model and global model conditions.

Furthermore, the traceability and immutability of blockchain ensure that every FL process is recorded in an auditable manner. For instance, FLChain introduced an auditable FL system with trust and incentives. Trustworthy participants are rewarded, while malicious ones are punished [23]. Another work by Kim et al. introduces a blockchain-based FL system (BlockFL) that incentivizes FL participants based on their contributions, which are determined by the dataset size [24].

Most of the aforementioned studies focused on single optimization to enhance the FL system. Additionally, the collaborative framework that allows any individual or organization to initiate or join the FL system has not been discussed. The collaboration framework introduced in [12] is not applicable to preserve participant data privacy, which is an important aspect in FL. Therefore, a collaborative framework

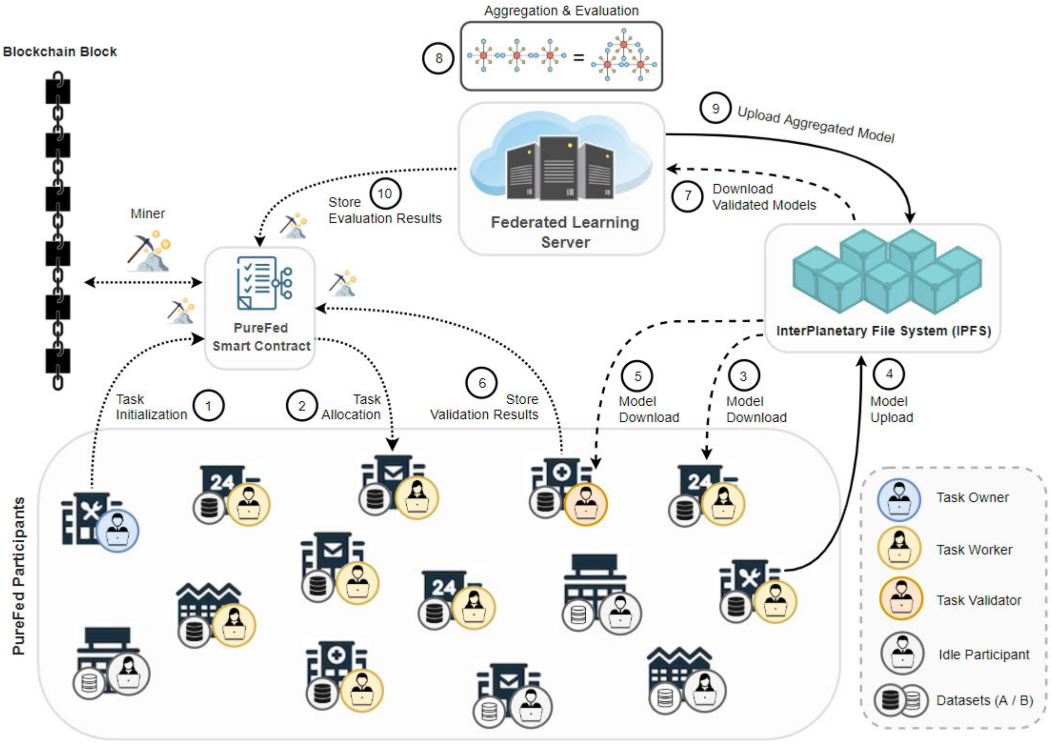

FIGURE 1. The general overview of the proposed PureFed framework.

that preserves participant data is needed. Furthermore, the efficiency of the framework is addressed by the lightweight smart contract gas costs, while trustworthiness is achieved by utilizing the properties of the blockchain network.

III. PROPOSED SYSTEM

This section covers the conceptualization, overall system design of PureFed, and security enhancements for encryption and digital signature to ensure the trustworthiness of the proposed system, including the proposed incentive and punishment mechanisms.

A. PROBLEM FORMULATION

FL has emerged as a promising paradigm for training machine learning models across decentralized and heterogeneous devices. In its current state, FL relies on a central server to initiate tasks, limiting the true potential of distributed learning. This limitation hinders the scalability, efficiency, and innovation potential of FL, particularly when dealing with diverse participant profiles. We identify two primary categories of FL participants:

1) HIGH COMPUTING POWER, LIMITED DATA

The first category comprises participants with ample computing power but a scarcity of data for AI model training. Models trained on such limited datasets tend to exhibit reduced accuracy when tested in slightly different environments, reflecting the narrow scope of their training data.

2) LIMITED COMPUTING POWER, VAST DATASETS

The second category consists of participants with limited computing resources but access to extensive datasets. While these participants may produce more accurate models, the trade-off is longer training times.

This duality in participant capabilities presents a challenge in achieving efficient and accurate federated learning outcomes. A comprehensive framework is required to bridge the gap between these participant types, fostering collaborative model training and addressing the trustworthiness of the FL process. Our objective is to formulate a novel FL framework, named PureFed that addresses these challenges. We aim to empower participants to initiate specific tasks within the FL process, thereby enabling a more distributed and collaborative approach as illustrated in Fig. 1. Moreover, we recognize the importance of trustworthiness in FL and seek to ensure the transparency and security of every event during training.

In summary, our problem formulation revolves around the need for an efficient and trustworthy FL framework that accommodates participants with varying computing resources and dataset sizes. The framework should enable broader collaboration while preserving data privacy and model ownership, ultimately leading to improved FL performance and innovation potential.

B. PUREFED FRAMEWORK OVERVIEW

The proposed PureFed framework can be divided into four key components, as illustrated in Fig. 2. These components

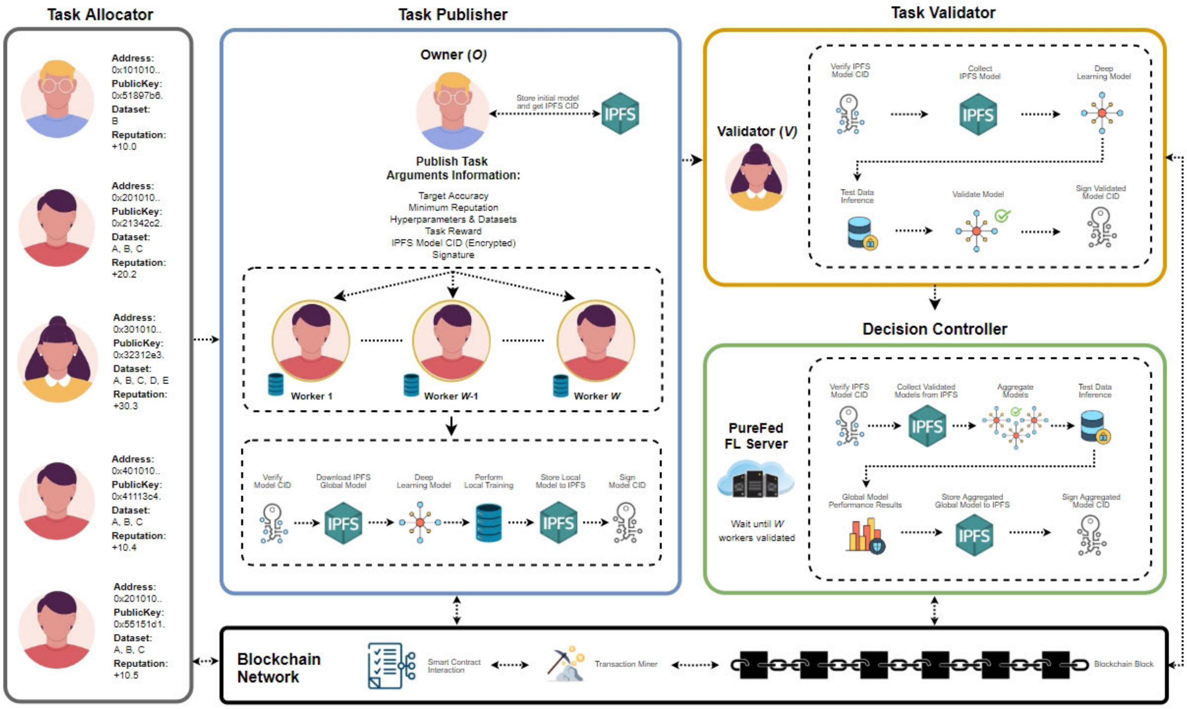

FIGURE 2. PureFed framework design enabling collaborative and trustworthy FL system.

have been designed to facilitate smart contract interactions, ensuring that every action related to task creation and participation is recorded on the blockchain network.

1) TASK PUBLISHER

The task publisher is responsible for creating tasks within the PureFed framework, a feature accessible to any registered individual. Task creation involves specifying various parameters, including model parameters and the estimated reward, which is distributed upon task completion by the task Owner (O).

2) TASK ALLOCATOR

The task allocator's role is to assign specific workers and validators based on the requirements set by the task publisher. In this research, we explore an adaptive allocation approach, considering factors such as reputation and dataset sizes to determine the most suitable participants for each task.

3) TASK VALIDATOR

The task validator is tasked with evaluating the model before the global aggregation process takes place. Task validator is performed by a participant that is selected as a validator (V) by task allocator. Their responsibilities include ensuring that the worker has stored a valid model and verifying the originality of the model provided by the worker.

4) DECISION CONTROLLER

The decision controller is responsible for overseeing the global model aggregation process, utilizing the validated models provided by V. Following the global model aggregation, an evaluation process is conducted to assess the overall model performance and verify whether the predetermined conditions are satisfied or not.

This division of responsibilities within the PureFed framework aims to streamline the FL process, enhance transparency, and ultimately improve the efficiency and effectiveness of collaborative AI model training.

C. PUREFED SEQUENCE DIAGRAM AND MODULES

To further illustrate the process of the PureFed framework, Fig. 3 depicts a sequence diagram for the collaborative and trustworthy FL system utilizing smart contracts and blockchain networks. There are four modules in PureFed, as follows:

1) PARTICIPANTS

The first module of PureFed is participants. In general, participants (P) are any individual who registers to the PureFed framework and is able to initiate and participate in a task to train an AI model. There are three roles of participants.

1) Owner (O) is the one who initiates the FL task and provides learning information along with reward details. The task is initiated with the following parameters:

$$O_{T(i)} = [T_{N(i)}, T_{I(i)}, T_{A(i)}, T_{R(i)}, T_{P(i)}, T_{M(i)}] \,. \qquad (1)$$

where $O_{T(i)}$ represent task i initialized by O. $T_{N(i)}$ denotes the task name, followed by $T_{I(i)}$ as the incentive or task reward and $T_{A(i)}$ as the target accuracy for the collaborative learning determined by O. $T_{R(i)}$ is

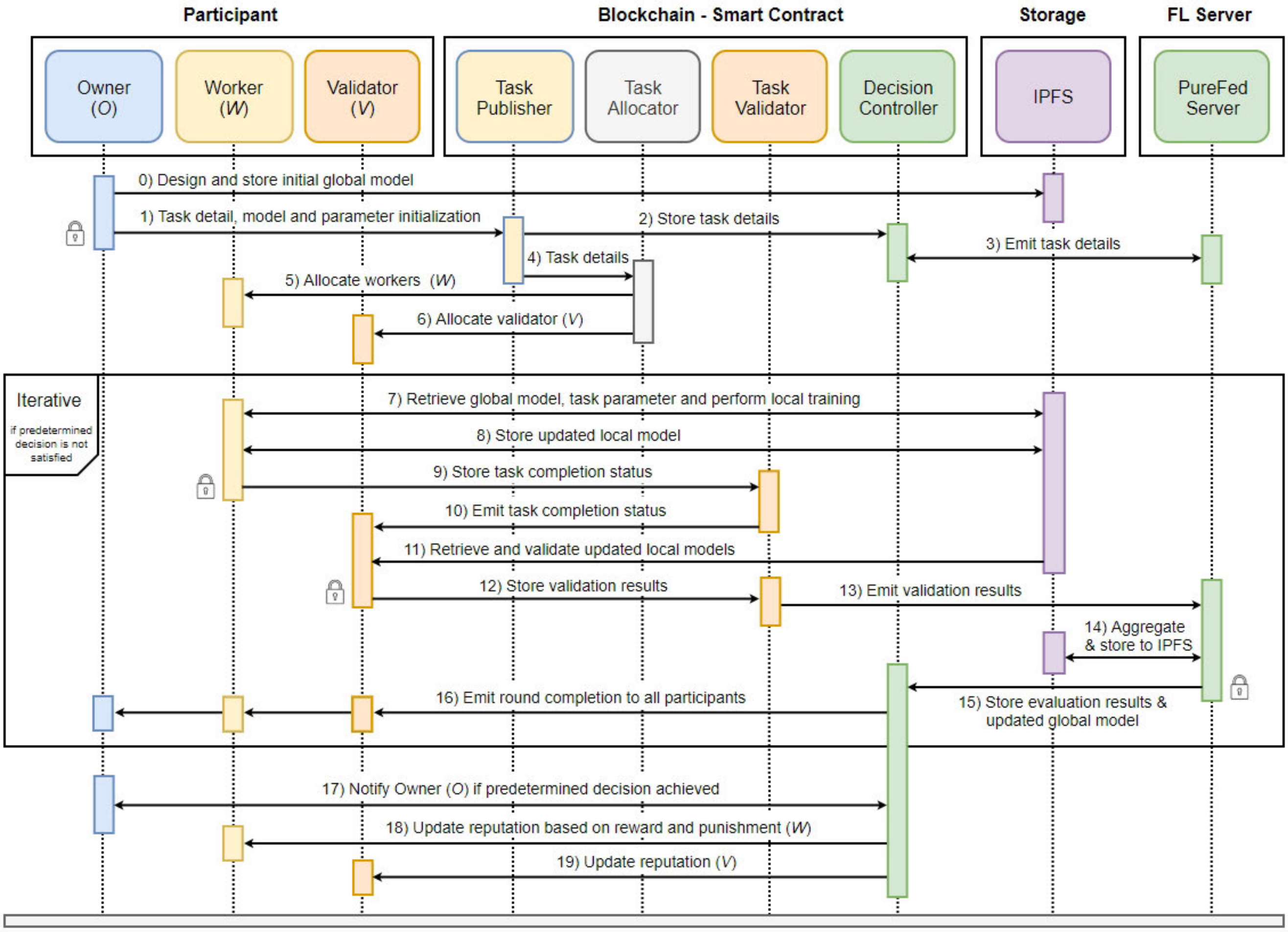
Participant
Blockchain - Smart Contract
Storage
FL Server
Owner (O)
Worker (W)
Validator (V)
Task Publisher
Task Allocator
Task Validator
Decision Controller
IPFS
PureFed Server
0) Design and store initial global model
1) Task detail, model and parameter initialization
2) Store task details
3) Emit task details
4) Task details
5) Allocate workers (W)
6) Allocate validator (V)
Iterative
if predetermined decision is not satisfied
7) Retrieve global model, task parameter and perform local training
8) Store updated local model
9) Store task completion status
10) Emit task completion status
11) Retrieve and validate updated local models
12) Store validation results
13) Emit validation results
14) Aggregate & store to IPFS
15) Store evaluation results & updated global model
16) Emit round completion to all participants
17) Notify Owner (O) if predetermined decision achieved
18) Update reputation based on reward and punishment (W)
19) Update reputation (V)

FIGURE 3. Sequence diagram of the proposed PureFed framework.

the minimum reputation requirement for P to join the learning process, where $T_{P(i)}$ is the hyperparameter configuration and $T_{M(i)}$ is the encrypted and signed model identifier. It is worth mentioning that the task owner is recorded automatically in the blockchain network and utilizes the access control design of smart contracts.

2) Worker (W) is responsible for performing local training based on their datasets. The local training process can be determined as follows:

$$W_{Model(w),T(i)} = Train(T_{M(i)}, T_{P(i)}, W_{data(w)}). \quad (2)$$

where $W_{Model(w),T(i)}$ denotes local model of $W_{(w)}$ for task $T_{(i)}$, $W_{data(w)}$ represents the local data owned by $W_{(w)}$. After the training process is completed, $W_{(w)}$ submits its local model after encrypting it and inserting a digital signature to prove the originality of the model.

3) Validator (V) is a participant responsible for verifying each worker model by testing the model performance and inserting a dual signature mechanism.

Finally, once the output of the FL process fulfills the predetermined condition, task rewards $T_{I(i)}$ are distributed accordingly to V and W based on their contributions, which are calculated for each federation round, while O receives the final AI model.

2) BLOCKCHAIN AND SMART CONTRACT

The second module is blockchain and smart contract, as shown in Fig. 2. According to the sequence diagram, there are four main functions in the smart contract developed for the PureFed framework. It is worth mentioning that PureFed uses the Ethereum blockchain network with a Proof of Authority (PoA) consensus mechanism to ensure fast transaction speed. The task publisher is responsible for receiving task information from O and forwarding it to the task allocator and decision controller. Once the task information is received, the task allocator initiates the assignment of workers and validators based on the provided details. This allocation depends on P' reputations and dataset sizes.

Initially, P is designated as W if their dataset size surpasses the predetermined threshold set by O. Once the list of workers, denoted as $W = \{1, 2, \ldots, w\}$, is established, their reputations are compared. Higher reputation scores increase the likelihood of a participant becoming V rather than W. In cases where two or more workers share the same reputation, the total number of datasets becomes the deciding factor in determining which participant becomes V. Likewise, a larger dataset size increases the chances of being allocated as a V. The task allocator assigns W and V based on Algorithm 1.

Algorithm 1 Pseudocode of Task Allocator

```
Input: Minimum reputation $T_{R(i)}$, Participants List $P$
Output: Set of workers and validator $W, V$
Initialize list of $W, V$
// Determine Worker ($W$) //
for each $p$ in $P$ do
    if $P_{(p,R)} \geq T_{R(i)}$ then
        $W \Leftarrow P_{(p)}$
    end
end
// Determine Validator ($V$) //
Perform TimSort $W$ based on reputation
for each $w$ in $W$ do
    if $W_{R(w,i)} \leq W_{R(w+1,i)}$ then
        if $W_{R(w,i)} \leq W_{R(w+1,i)}$ then
            $V \Leftarrow W[w+1]$
        end
        else
            $V \Leftarrow W[w]$
        end
    end
    else
        $V \Leftarrow W[0]$
        break
    end
end
Return allocation list of $W, V$
```

Following the allocation of W and V, local training is performed by each participant in W. After all training processes are reported as completed, V initiates the verification process by evaluating each model's performance. If a model is deemed valid, V inserts a validation signature for the respective worker. Only models validated by V are selected for global aggregation by the aggregator.

Subsequently, after the aggregation is completed, the decision controller checks if the performance of the aggregated model satisfies the predetermined conditions set by O. The training process concludes if the conditions are met, and rewards are distributed accordingly to W and V. However, if the conditions are not satisfied, W performs an additional federation round, repeating the same process until the performance results meet the specified requirements.

3) STORAGE

The third element is off-chain data storage. In this work, we have chosen to utilize the InterPlanetary File System (IPFS) to store every AI model. IPFS provides a distributed data-sharing network, identifying content through cryptographic hashes, which are unique to each piece of content. We opted for IPFS over the InterPlanetary Name System (IPNS) due to its immutability. The primary goals of this work are trustworthiness and ensuring that all information remains immutable and traceable. As mentioned earlier, IPFS is employed to store both global and local AI models for the PureFed framework.

Once a file is uploaded to IPFS, it returns a Content Identifier (CID) in the form of a hash, referencing the file's location on the IPFS network. To enhance security and privacy, this CID is encrypted before being stored on the blockchain. This encryption safeguards against data leakage and ensures that only valid participants can access the model CID.

4) FL SERVER

The last element is the FL server, which serves as the model aggregator and evaluator. Initially, the aggregator collects all validated model CIDs from the blockchain network and verifies the digital signatures. Once the validation is completed, the model aggregation is conducted, followed by model evaluation. The aggregated model is stored in IPFS as the new global model, while the evaluation results are stored in the decision controller to determine the next step in the collaborative learning process.

D. ENCRYPTION AND DUAL SIGNATURE MECHANISMS

To establish secure data transfer between participants in the PureFed, we employ the following security mechanisms: (i) We use symmetric key encryption to encrypt and decrypt AI model parameters. The secret key is generated by O during the task initialization process and is shared within the blockchain network via a smart contract. Only valid W are granted access to the symmetric key for decryption. (ii) We employ a digital signature mechanism, specifically Elliptic Curve Digital Signature Algorithm (ECDSA), to verify the authenticity and integrity of AI models. ECDSA is a cryptographic algorithm based on Elliptic Curve Cryptography (ECC) designed for digital signatures. It requires an elliptic curve key pair i.e., private and public keys for operation. The ECDSA private key is used to generate digital signatures, while the public key is used to verify signature authenticity. These encryption and dual signature mechanisms are applied to all participants in PureFed, including the FL server. Algorithm 2 details the encryption and signing process, while Algorithm 3 illustrates the data integrity verification and decryption process to obtain the model CID.

1) OWNER

Initially, O generates the symmetric key used for encryption. This key is employed to encrypt the model CID obtained from IPFS. The encrypted model CID is then signed using the ECDSA private key to obtain the digital signature.

Algorithm 2 Pseudocode for Encryption and Signing

```
Input: Model CID $M_{(cid)}$, Symmetric Key $\kappa$, Private Key $k_{private}$
Output: Encrypted and Signed Model $T_M$
$c_m = Encrypt\ (M_{(cid)}, \kappa)$
$s_m = Sign\ (k_{private}, c_m)$
$T_m = c_m, s_m$
Return encrypted and signed Model $T_M$
```

Algorithm 3 Pseudocode for Verify and Decryption

```
Input: Encrypted and Signed Model $T_M$, Symmetric Key $\kappa$, Public Key $k_{public}$
Output: Model CID $M_{(cid)}$
$c_m, s_m = T_m$
$\hat{c_m} = Verify\ (k_{public}, s_m)$
if $c_m == \hat{c_m}$ then
    $M_{(cid)} = Decrypt\ (c_m, \kappa)$
    Return model CID $M_{(cid)}$
end
```

Both the encrypted data and the signature, denoted as $T_{M(i)}$, are stored on the blockchain network via a smart contract.

2) WORKER

Once allocated as W, the specific task's symmetric key becomes visible to W. Each W subsequently decrypts $T_{M(i)}$ and verifies that the model CID matches the digital signature. The local training process is then executed to obtain a new local model, which is subsequently uploaded to IPFS. The CID is encrypted and signed by each W before being stored on the blockchain network.

3) VALIDATOR

Similarly, V uses the symmetric key to decrypt the encrypted model CID and also verifies the W's digital signature. If the signature matches the encrypted data, the model is validated. After validation, V signs the encrypted model CID provided by W. Subsequently, the smart contract is updated with the validation results and the V's signature.

4) AGGREGATOR

As previously described, the FL server aggregates only validated models that have been verified by V. Before aggregation, the FL server decrypts the model CID and verifies both V and the W's signatures. Following verification, the new global model is evaluated and stored on IPFS. Similarly, the IPFS CID is encrypted and signed by the FL server before being updated on the blockchain network.

E. DYNAMIC AGGREGATION SCHEME

In the traditional FL system, the aggregation process involves combining all the models received by the FL server. Previous efforts to enhance FL accuracy have been explored in our prior work [13]. In this study, we take a different approach to align with a collaborative framework where any individual can participate in the training process. However, to ensure that the aggregated model represents the best aggregation results, we employ a dynamic aggregation scheme. This scheme takes into account three conditions:

- *Traditional*. In this condition, every model validated by V is aggregated to form a new global model without considering the conditions of the model. To some extent, despite the dataset and reputation requirements set by O, the chance of a model becoming an outlier compared to the others remains high.
- *Average Accuracy*. In this condition, the FL server only aggregates models that provide accuracy higher than the overall accuracy of all W in a specific FL round. The FL server evaluates each model and then establishes a threshold value based on the average accuracy.
- *Average Loss*. Similar to the average accuracy, the last condition considered in this work is average loss. Instead of calculating and comparing accuracy, the loss of all W is examined. Only certain validated models with a loss lower than the average loss are considered in the aggregation process.

The proposed dynamic aggregation scheme in PureFed has the capability to dynamically select the condition that offers superior model performance for each federation round. Through this approach, the desired target accuracy set by O can be attained more rapidly compared to relying solely on a traditional aggregation scheme.

F. REWARD AND PUNISHMENT MECHANISMS

PureFed is a collaborative FL framework that ensures the trustworthiness of participants during model training with an efficient process. To support the framework and incentivize participation, we have developed a reward mechanism in this work. The reward is calculated based on participants' contributions to the global model, which are recorded after each federation round. The FL server is responsible for calculating rewards based on the validated model's performance when tested with its datasets. The contribution of each participant is calculated as follows:

$$Contrib(w, r) = \frac{Acc(w, r)}{\sum_{i=0}^{W} Acc(r)}\,. \tag{3}$$

The contribution of W_w in round r is determined by the accuracy generated by W_w in that round divided by the total accuracy of all participants in round r. Based on these contributions, the reward for each round is also determined, with the number of rounds varying based on the task's complexity. The reward for each federation round in task i

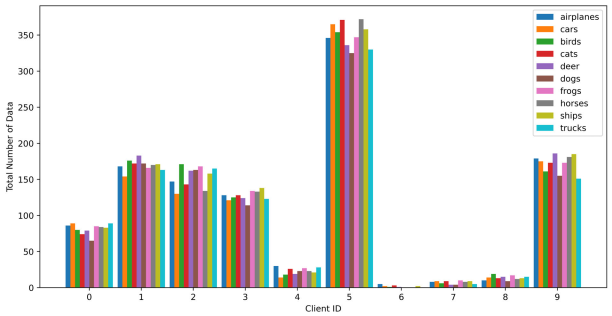

FIGURE 4. Dataset distribution for the first 10 clients follows a Non-IID distribution, divided based on the Dirichlet approach with $\alpha = 0.5$.

is calculated as follows:

$$Reward(i) = \frac{R(i)}{T_{I(i)} \times 0.9 \times W} \,. \tag{4}$$

where R is the total number of rounds required to fulfill predetermined conditions for a task, and $T_{I(i)}$ is the total reward provided by the task owner. According to Equation 4, 90% of the total reward is distributed among workers W, while the remaining 10% is allocated to the validator. Finally, the reward for each participant collaborating to build the AI model for task i is given by:

$$Reward(w, r) = Contrib(w, r) \times Reward(i) \,. \tag{5}$$

Rewards are distributed to each participant upon task completion. In addition to rewards, we have also considered a punishment mechanism to prevent malicious participation during model training. The punishment is determined by consecutive negative contributions from a participant. A contribution is considered negative if the performance evaluation result falls below the minimum accuracy threshold for a specific round. The minimum accuracy is calculated as follows:

$$MinAccuracy(r) = \frac{\overline{Acc\ (r)}}{W} \times \frac{100}{r} \,. \tag{6}$$

where $\overline{Acc\ (r)}$ is the average accuracy of all participants in round r, divided by the total number of workers W and the current round r. Under this condition, if a worker consistently provides negative contributions in three consecutive federation rounds, the worker is blacklisted from participating in the same task, and any positive contributions made in the previous round for that task are disregarded. Additionally, the reputation of each participant is updated after task completion, with positive contributions resulting in a $+1$ reputation, while negative contributions lead to a -1 reputation.

IV. PERFORMANCE EVALUATION AND DISCUSSION

This section outlines the evaluation of the proposed PureFed framework. We begin by presenting the simulation settings used in this study, followed by an exploration of federated evaluation in a collaborative scenario. We then provide a performance analysis to illustrate the framework's effectiveness.

TABLE 1. Parameter configurations for simulation work.

Parameter	Value
FL Framework	Flower v.1.3.0
Dataset	non-IID MNIST (Dirichlet α = 0.5)
Number of Workers	10 - 50
Number of Validator	1
Local Epoch	5
Target Accuracy	70%
Model Structure	[Conv2D, MaxPooling2D, Flatten,Dense, Dropout, Dense]
Optimizer (lr)	SGD (0.0001)

Additionally, we discuss the scalability analysis of PureFed and assess its trustworthiness.

A. SIMULATION CONFIGURATIONS

The proposed PureFed framework aims to enable efficient and trustworthy collaboration for the FL system. To evaluate the performance of the proposed framework, we vary the total number of participants joining PureFed, ranging from 10 to 50. Each participant is equipped with a subset of the MNIST dataset [25], distributed under non-IID conditions. The MNIST dataset is divided into 50 subsets, with the allocation of the first 10 illustrated in Fig. 4. The complete set of parameters used for the simulation is provided in Table 1. For the sake of simplicity, only one FL job is initialized by O.

Regarding blockchain implementation, we utilize an Ethereum-based network with a PoA consensus for both the local and public test networks. The interaction between Flower and the FL system is implemented using web3 libraries available in the Python language. For the local blockchain network, we adopt the Ganache library to create the environment, while Goerli is used for the public test network. Finally, the Remix integrated development environment is employed to develop, deploy, and test the smart contract of PureFed.

Furthermore, to assess the effectiveness of the proposed PureFed, we employ several performance metrics during evaluation. First, we calculate the model's accuracy using Equation 7. After achieving the desired target accuracy, we also compute the Precision, Recall, and F1-score using Equations 8, 9, and 10, respectively.

$$Accuracy\ (\%) = \frac{T_p + T_n}{T_p + T_n + F_n + F_p} \times 100 \,. \tag{7}$$

$$Precision\ (\%) = \frac{T_p}{T_p + F_p} \times 100 \,. \tag{8}$$

$$Recall\ (\%) = \frac{T_p}{T_p + F_n} \times 100 \,. \tag{9}$$

$$F1 - score\ (\%) = \frac{2 \times Precision \times Recall}{Precision + Recall} \,. \tag{10}$$

TABLE 2. Performance of PureFed framework under different aggregation schemes with 10 workers.

Aggregation Scheme	Accuracy = 70%		FL Round = **36**		
	FL Round	Time (ms)	Precision (%)	Recall (%)	F1-score (%)
FedAvg	96	786,594	19.51	22.92	21.08
Accuracy	78	540,474	25.34	28.13	26.66
Loss	**36**	**254,618**	**70.51**	**70.28**	**70.39**
Dynamic	43	307,842	52.14	54.87	53.47

where T_p is the true positive, T_n is the true negative, followed by F_p and F_n, which are the false positive and false negative, respectively. In terms of the effectiveness of blockchain implementation, transaction time and gas fees were considered. The transaction time is calculated based on the timestamp difference between the initial interaction with the smart contract after the transaction hash is obtained.

B. FEDERATED EVALUATION

Initially, the evaluation of the FL carried out in the PureFed framework is investigated using one task, with 10 workers and one validator. As detailed previously, the dataset used for the evaluation is MNIST, which contains $60,000$ images of handwritten digits. These data are divided into a total of 50 participants using the Dirichlet technique with $\alpha = 0.5$, following the non-IID condition. The first 10 participants and their datasets are selected as the first evaluation scenario. The evaluation is conducted under different aggregation schemes, such as FedAvg, accuracy, and loss thresholds, as well as the dynamic aggregation proposed in this work.

The results detailed in Table 2 show that the traditional FL aggregation strategy, the FedAvg technique, needs 96 rounds to achieve an accuracy of 70%. This is not sufficient to cope with the PureFed framework, which allows any participant to collaboratively join the FL task in an efficient manner. The second aggregation method is the accuracy threshold, which is determined based on the average accuracy. The results indicate that aggregation based on accuracy requires 78 FL rounds, which is relatively large compared to other aggregation methods like loss-based and dynamic-based approaches. The dynamic-based aggregation proposed in this work requires 43 rounds to achieve 70% accuracy, which is slightly higher compared to the loss-based aggregation that only takes 36 rounds. However, it's worth noting that as the total number of participants is limited to 10, the effectiveness of the proposed dynamic aggregation may not be appropriate.

Similar to the FL rounds, the larger the number of rounds required to train the model, the longer the training time to achieve 70% accuracy. The loss-based aggregation only requires $254,618$ ms, followed by dynamic aggregation with $307,842$ ms, accuracy-based and FedAvg with $540,474$ ms and $786,594$ ms, respectively.

TABLE 3. Gas costs for smart contract deployment and interactions.

Stage	Process Name	Entity	Gas Costs
Deployment	N/A	N/A	3816044
Initialization	Registration	Participants	52982
	Task Publish and Allocation	Owner	228049
Training	Retrieve Task Information	Workers	0
	Store Task Completion	Workers	70839
	Retrieve Workers Data	Validator	0
	Store Verification Results	Validator	57878
	Retrieve Validated Data	Aggregator	0
	Store Updated Model	Aggregator	43705
	Task Decision	Aggregator	35639
Finish	Reward Distribution	Aggregator	215390
	Task Termination	Aggregator	45455

TABLE 4. Gas cost comparison between PureFed with state-of-the-art framework design.

Process Name	Contract Deployment	Task Initialization	Task Training	Task Completion
LM [12]	10424901	509740	1411921	401771
PureFed	3816044	281031	208061	260845

In addition, the model performance in terms of Precision, Recall, and F1-score is also calculated. We used the lowest FL round to achieve the target accuracy of 70%, which is 36. The performance of the model using different aggregation schemes is calculated at round 36. Since the training process of the other aggregation schemes is not finished by round 36, the performance of the model is relatively low for FedAvg and accuracy-based approaches. It is worth noting that the performance of the dynamic approach is acceptable in round 36 of FL training with an F1-score of 53.47%, whereas the loss-based approach achieved 70.39%.

C. BLOCKCHAIN AND SMART CONTRACT EVALUATION

The second evaluation conducted in this work assesses the efficiency of the proposed smart contract in terms of gas costs required for deployment and interaction. Table 3 displays the detailed gas costs necessary for PureFed to facilitate collaborative FL processes. The gas cost for smart contract deployment is 3816044 gas. In general, the deployment cost for smart contracts is relatively high compared to the interaction cost after the contract has been deployed.

PureFed comprises three different stages: initialization, the training process (which occurs iteratively based on the target accuracy condition), and task termination. During initialization, two processes are required before a task can be trained. Registration incurs a gas cost of 52982, followed by publishing a task and allocating a worker and validator, which require 228049 gas. Once the task is created, the training

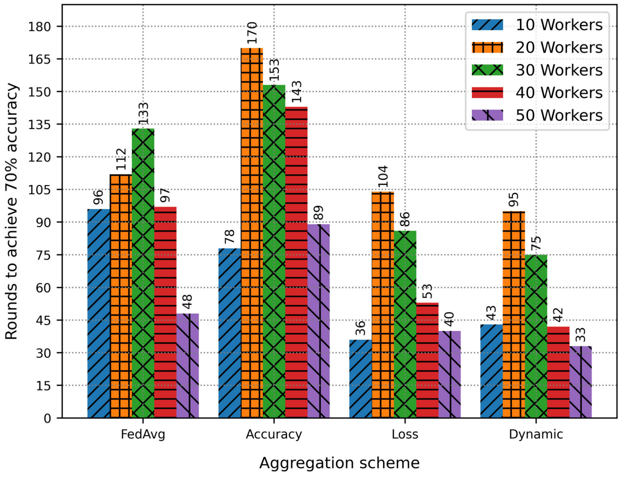

FIGURE 5. **Performance comparison of PureFed scalability under various aggregation schemes to attain a 70% accuracy threshold.**

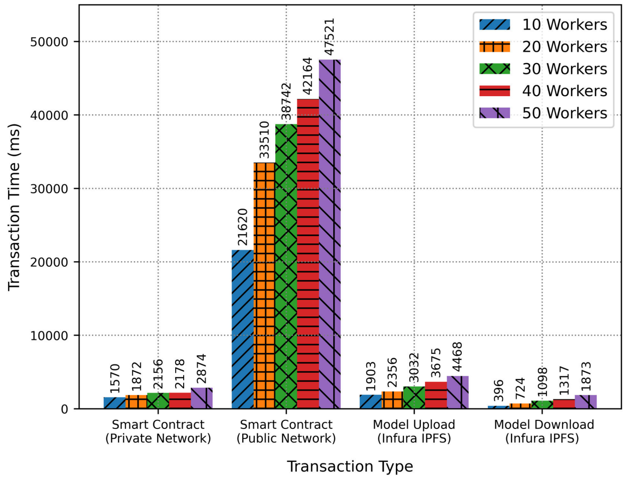

FIGURE 6. **Performance comparison of PureFed scalability using smart contracts and IPFS transaction times.**

process begins, and the only gas cost incurred by workers is to modify the blockchain record (e.g., store task completion and verification results). Functions to retrieve current information can be performed without any gas cost (e.g., retrieving task information and validated data). Lastly, after the task achieves the desired accuracy, reward distribution and task termination are executed. Reward distribution requires 215390 gas, whereas task termination necessitates 45455 gas.

To provide a better understanding of the efficiency of the proposed PureFed compared to state-of-the-art designs in the collaborative market, Table 4 displays a gas cost comparison between PureFed and existing studies, such as LM [12]. As observed, from contract deployment to task initialization, training, and completion, the proposed PureFed achieves lower gas costs while providing the same functionality. For instance, a total of 63.39% gas cost efficiency is achieved from smart contract deployment. Moreover, PureFed preserves user privacy without sharing the owner's private dataset. PureFed's efficiency is achieved by not calculating the incentive mechanism on the blockchain, as done in LM. Iteratively computing incentives via the blockchain network is inefficient, so PureFed calculates and records the incentive on the aggregator side. The reward is distributed after the learning process is completed, resulting in higher efficiency.

D. SCALABILITY ANALYSIS

To demonstrate the performance of PureFed in handling collaborative learning with an increasing number of participants, we evaluated five different variations of worker sizes, each with one task and one validator. Scalability analysis is performed for both federated and blockchain evaluation.

1) SCALABILITY OF FEDERATED LEARNING

For the federated evaluation, the aggregation schemes were assessed based on FedAvg, accuracy-based, loss-based, and dynamic-based methods proposed in the PureFed framework. Fig. 5 illustrates these five worker variations with four aggregation schemes. It is evident that the accuracy-based aggregation performs the poorest compared to the other techniques. For example, it takes a total of 170 FL iterations to achieve 70% accuracy when there are 20 workers, whereas FedAvg only requires 112 iterations, followed by loss-based and dynamic-based with 104 and 95 rounds, respectively.

In general, as the number of workers increases, the total rounds required to achieve the desired accuracy decrease. However, an increase in the number of workers from 10 to 20 leads to an increase in the total number of rounds required to achieve 70% accuracy. To the best of our knowledge, this phenomenon is caused by the dataset distribution among workers 11 to 20, which is highly unbalanced and affects the aggregation results, ultimately increasing the number of rounds. When more than 50% of workers have unbalanced datasets, the training process takes a longer time.

Despite the impact of data distribution on PureFed's performance, the proposed dynamic aggregation technique demonstrates better performance by reducing the total rounds required to meet the predetermined accuracy set by the Owner. For instance, in a task involving 50 workers, dynamic aggregation only needs 33 rounds to achieve 70% accuracy, whereas the loss-based method requires 40 rounds, followed by FedAvg and accuracy-based methods with 48 and 89 rounds, respectively.

2) SCALABILITY OF BLOCKCHAIN NETWORK

In the context of blockchain scalability, this work investigates two conditions related to transaction time. Firstly, we examine the interaction time for smart contracts in both private and public blockchain networks. The private network is established on the Ethereum blockchain, utilizing Ganache with the PoA consensus algorithm, while the public network is tested on top of Goerli testnets. Transaction time is calculated by subtracting the difference between the timestamp after the transaction is confirmed and the

TABLE 5. A comparative analysis between the proposed framework and state-of-the-art.

Reference Number	Framework Type	Privacy Preserving	Incentive Mechanism	Main Computing	Distributed Storage	Total Entities	Scalability Evaluation
[26]	Computing Services	No	Static	Off-chain	IPFS	3	Yes
[24]	Distributed Computing	Yes	Static	Off-chain	N/A	2	No
[27]	Distributed Computing	No	No Incentive	On-chain	IPFS	3	No
[12]	Both	No	Dynamic	Off-chain	IPFS	3	No
PureFed (Ours)	Both	Yes	Dynamic	Off-chain	IPFS	4	Yes

timestamp of the initial interaction. As illustrated in Fig. 6, the average transaction time in the private network is significantly better compared to the public network. For example, with a total of 30 workers, the private network only takes an average of 2156 ms to interact with the smart contract, whereas the public network requires 38742 ms. It is evident that the private network outperforms the public blockchain network. The private network necessitates participants to have knowledge of other entities in the blockchain, making it a closed network. In the context of open collaboration, as suggested in PureFed, the public network is more suitable, as transaction mining can be carried out by any entity. However, if the goal is fast communication and low-latency networking, the application of PureFed with a private network is also feasible by configuring additional information of the network entities.

Secondly, we depict the average transaction time required for model upload and download to or from the IPFS network, as shown in Fig. 6. It is worth noting that the IPFS network utilized in this study is based on Infura IPFS. The results reveal that model upload requires more time compared to model download. For instance, with a total of 50 clients, the average time required for model upload to IPFS is 4468 ms, whereas model download only takes 1873 ms on average. The presented results indicate that an increase in the number of participants for both smart contract and IPFS interactions leads to higher transaction times.

E. TRUSTWORTHINESS ANALYSIS

This paper focuses on efficient collaboration and trustworthiness in the FL system using a blockchain network. Analyzing the trustworthiness of a system or framework can be done using several parameters. The proposed PureFed was assessed using four common approaches and considerations for evaluating its trustworthiness.

1) SECURITY MEASURES

The first aspect of evaluating the trustworthiness of PureFed is security measures. This aspect includes encryption and access control mechanisms to ensure data protection. Based on this aspect, the PureFed framework satisfies the requirements. First, PureFed preserves users' private datasets without sharing any user data information with the worker or validator. Second, data transfer between the Owner, Worker, Validator, and Aggregator is done using symmetric key encryption followed by dual signature mechanisms powered by ECDSA. Therefore, sensitive information, in this case, model parameters, is completely secure.

2) CONSISTENCY

The second aspect is consistency. PureFed is built on top of smart contracts designed to execute predetermined functions with given roles and authority. Therefore, PureFed can deliver a dependable framework to support collaborative learning.

3) TRANSPARENCY

Blockchain networks have become popular due to their transparency and immutable concept. Any information or transaction performed on a blockchain network can be traced, thanks to the block structure of blockchain networks that allows this traceability. PureFed adopted blockchain to store any event performed for collaborative learning, such as task creation, task submission, as well as task completion. Therefore, the transparency of PureFed is guaranteed.

4) RELIABILITY

Last but not least is reliability. A trustworthy framework needs to be reliable and fault-tolerant. A framework can be classified as reliable if it meets the expectations of the users. In this case, the PureFed framework is designed with a dynamic aggregation scheme that selects the best-performing aggregation technique for each federation round. By utilizing this, the performance expected by the task owner can be achieved over time.

F. COMPARATIVE ANALYSIS

Based on the presented results in this section, the proposed PureFed can achieve satisfactory performance in FL evaluation, blockchain, and its scalability, as well as trustworthiness analysis. To better understand the differences between the proposed PureFed and state-of-the-art research in collaborative FL, we present a comparative analysis between PureFed and previous work. Table 5 provides details of four different pieces of research related to this work, with seven different aspects in comparison.

First, in terms of framework type, the proposed framework is capable of delivering both computing services and distributed computing, whereas several research projects only consider one of them [24], [26], [27]. In the FL process, an incentive mechanism is crucial to attract collaborative learning; therefore, a dynamic incentive mechanism that adapts over time is required. The dynamic incentive mechanism is provided in LM [12], which is also considered in PureFed.

The main difference between PureFed and other works is in privacy preservation. Most of the frameworks require task owners to share their information [12], [26], [27], which is not aligned with the FL concept and also results in larger communication overhead. In PureFed, the framework ensures that all participant datasets are secure and not recorded by any entities in the system. Additionally, existing frameworks only consider three participants (e.g., owner, worker, validator), while PureFed considers four, with the addition of the FL server that calculates incentives every federation round as well as aggregates the verified model. Lastly, PureFed also evaluates actual scalability analysis by increasing the total number of workers participating in a task, whereas the existing studies did not. It can be said that PureFed enables collaborative learning with privacy preservation and reliable performance, as demonstrated by the given scalability analysis results.

V. CONCLUSION

This paper introduces PureFed as an efficient, collaborative, and trustworthy FL framework. PureFed empowers participants to initiate FL tasks or participate in others. It leverages blockchain networks, utilizing smart contracts with their traceability and immutability. The proposed PureFed incorporates symmetric key encryption and dual digital signature mechanisms using ECDSA to ensure the integrity of the shared model in the framework. To achieve rapid model convergence, a dynamic aggregation scheme is designed, selecting the best-aggregated model from three different techniques: FedAvg, accuracy-based, and loss-based. Additionally, a dynamic incentive and punishment mechanism are proposed to attract more collaborators and ensure the trustworthiness of the PureFed framework.

Based on extensive performance evaluation, the proposed PureFed framework is 63.39% and 67.72% more efficient compared to its counterparts in terms of smart contract deployment and interaction gas costs. Furthermore, the scalability analysis of PureFed indicates that the proposed framework can adapt and deliver satisfactory performance, such as achieving target accuracy in fewer rounds. In a comparative analysis, PureFed is shown to provide significant additional features in privacy preservation, a crucial aspect of FL systems. For future work, investigating the impact of malicious workers and implementing decentralized applications (dApps) to support the PureFed framework will be explored.

REFERENCES

[1] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data–AI integration perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1328–1347, Apr. 2021.

[2] D. Zha, Z. Pervaiz Bhat, K.-H. Lai, F. Yang, Z. Jiang, S. Zhong, and X. Hu, "Data-centric artificial intelligence: A survey," 2023, *arXiv:2303.10158*.

[3] A. Bailly, C. Blanc, É. Francis, T. Guillotin, F. Jamal, B. Wakim, and P. Roy, "Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models," *Comput. Methods Programs Biomed.*, vol. 213, Jan. 2022, Art. no. 106504.

[4] Y. Gong, G. Liu, Y. Xue, R. Li, and L. Meng, "A survey on dataset quality in machine learning," *Inf. Softw. Technol.*, vol. 162, Oct. 2023, Art. no. 107268.

[5] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2021.

[6] 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*. Westminster, U.K.: PMLR, 2017, pp. 1273–1282.

[7] L. Jai Vinita and V. Vetriselvi, "Federated learning-based misbehaviour detection on an emergency message dissemination scenario for the 6G-enabled Internet of Vehicles," *Ad Hoc Netw.*, vol. 144, May 2023, Art. no. 103153.

[8] M. Adi Paramartha Putra, S. Maliah Rachmawati, M. Abisado, and G. A. Sampedro, "HFTL: Hierarchical federated transfer learning for secure and efficient fault classification in additive manufacturing," *IEEE Access*, vol. 11, pp. 54795–54807, 2023.

[9] Q. Dou et al., "Federated deep learning for detecting COVID-19 lung abnormalities in CT: A privacy-preserving multinational validation study," *Npj Digit. Med.*, vol. 4, no. 1, p. 60, Mar. 2021, doi: 10.1038/s41746-021-00431-6.

[10] U. Majeed, L. U. Khan, A. Yousafzai, Z. Han, B. J. Park, and C. S. Hong, "ST-BFL: A structured transparency empowered cross-silo federated learning on the blockchain framework," *IEEE Access*, vol. 9, pp. 155634–155650, 2021.

[11] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. Data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[12] L. Ouyang, Y. Yuan, and F.-Y. Wang, "Learning markets: An AI collaboration framework based on blockchain and smart contracts," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14273–14286, Aug. 2022.

[13] M. A. P. Putra, A. R. Putri, A. Zainudin, D.-S. Kim, and J.-M. Lee, "ACS: Accuracy-based client selection mechanism for federated industrial IoT," *Internet Things*, vol. 21, Apr. 2023, Art. no. 100657.

[14] L. Zhao, J. Jiang, B. Feng, Q. Wang, C. Shen, and Q. Li, "SEAR: Secure and efficient aggregation for Byzantine-robust federated learning," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 5, pp. 3329–3342, Sep. 2022.

[15] X. Ma, J. Zhang, S. Guo, and W. Xu, "Layer-wised model aggregation for personalized federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2022, pp. 10082–10091, doi: 10.1109/CVPR52688.2022.00985.

[16] H. Zeng, T. Zhou, Y. Guo, Z. Cai, and F. Liu, "FedCav: Contribution-aware model aggregation on distributed heterogeneous data in federated learning," in *Proc. 50th Int. Conf. Parallel Process.*, Aug. 2021, pp. 1–20, doi: 10.1145/3472456.3472504.

[17] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis," *Sci. Rep.*, vol. 12, no. 1, p. 1953, Feb. 2022, doi: 10.1038/s41598-022-05539-7.

[18] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2650–2654.

[19] R. N. Alief, M. A. Paramartha Putra, A. Gohil, J.-M. Lee, and D.-S. Kim, "FLB2: Layer 2 blockchain implementation scheme on federated learning technique," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2023, pp. 846–850.

[20] A. Qammar, A. Karim, H. Ning, and J. Ding, "Securing federated learning with blockchain: A systematic literature review," *Artif. Intell. Rev.*, vol. 56, no. 5, pp. 3951–3985, May 2023, doi: 10.1007/s10462-022-10271-9.

[21] C. Korkmaz, H. E. Kocas, A. Uysal, A. Masry, O. Ozkasap, and B. Akgun, "Chain FL: Decentralized federated machine learning via blockchain," in *Proc. 2nd Int. Conf. Blockchain Comput. Appl. (BCCA)*, Nov. 2020, pp. 140–146.

[22] S. Ko, K. Lee, H. Cho, Y. Hwang, and H. Jang, "Asynchronous federated learning with directed acyclic graph-based blockchain in edge computing: Overview, design, and challenges," *Expert Syst. Appl.*, vol. 223, Aug. 2023, Art. no. 119896.

[23] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "FLChain: A blockchain for auditable federated learning with trust and incentive," in *Proc. 5th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2019, pp. 151–159.

[24] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

[25] L. Deng, "The MNIST database of handwritten digit images for machine learning research [Best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[26] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. H. Deng, "CrowdBC: A blockchain-based decentralized framework for crowdsourcing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 6, pp. 1251–1266, Jun. 2019.

[27] G. J. Mendis, Y. Wu, J. Wei, M. Sabounchi, and R. Roche, "A blockchain-powered decentralized and secure computing paradigm," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 2201–2222, Oct. 2021.

MADE ADI PARAMARTHA PUTRA (Member, IEEE) received the Ph.D. degree in IT convergence engineering from the Kumoh National Institute of Technology, Gumi, South Korea, in 2024. He is currently a full-time Lecturer of informatics engineering with Primakara University, Bali, Indonesia, and also the Director of the Postgraduate Studies, in 2024. His research interests include named data networks (NDN), the real-time Internet of Things, federated learning optimization, blockchain, and energy efficient architecture. He served as a reviewer for high-impact journals, including IEEE Access, IEEE Journal, and IEEE Transactions on Communications.

NYOMAN BOGI ADITYA KARNA (Senior Member, IEEE) received the Ph.D. degree in electrical engineering and computer science from Bandung Institute of Technology, West Java, Indonesia, in 2018. He has been a full-time Lecturer with the School of Electrical Engineering, Telkom Higher School of Technology (now Telkom University), West Java, since 1999. His research interests include the intelligent IoT, cybersecurity, and the Internet of Drone Things.

REVIN NAUFAL ALIEF received the bachelor's degree in telecommunication engineering from Telkom University, Indonesia, in 2020. He is currently pursuing the master's degree in IT convergence engineering with the Kumoh National Institute of Technology, Gumi, South Korea. He is also a Researcher with the Network System Laboratory, Kumoh National Institute of Technology. His research interests include blockchain, real-time systems, and machine learning.

AHMAD ZAINUDIN (Member, IEEE) received the B.Eng. and M.Eng. degrees in telecommunication engineering and electrical engineering from the Electronic Engineering Polytechnic Institute of Surabaya, Indonesia, in 2011 and 2014, respectively, and Ph.D. degree in electronic engineering from the Networked Systems Laboratory, Kumoh National Institute of Technology (KIT), Gumi, South Korea. He joined the Division of Telecommunication Engineering, Department of Electrical Engineering, Electronics Engineering Polytechnic Institute of Surabaya, as a Lecturer, in 2012. He has been a full-time Researcher with the Networked Systems Laboratory, KIT, since September 2021. His research interests include intrusion detection systems, the industrial IoT, federated learning, blockchain, and metaverse applications.

DONG-SEONG KIM (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2003. From 1994 to 2003, he was a full-time Researcher with ERC-ACI, Seoul National University. From March 2003 to February 2005, he was a Postdoctoral Researcher with the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. From 2007 to 2009, he was a Visiting Professor with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He was the Dean of IACF, from 2019 to 2022. He is currently the Director of the KIT Convergence Research Institute and the ICT Convergence Research Center (ITRC and NRF advanced research center program), supported by Korean Government at the Kumoh National Institute of Technology, and the Director of NSLab Company Ltd. His research interests include the real-time IoT and smart platforms, industrial wireless control networks, networked embedded systems, fieldbus, metaverse, and blockchain. He is a Senior Member of ACM.

JAE-MIN LEE (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2005. From 2005 to 2014, he was a Senior Engineer with Samsung Electronics, Suwon, South Korea. From 2015 to 2016, he was a Principle Engineer with Samsung Electronics. Since 2017, he has been an Associate Professor with the School of Electronic Engineering and the Department of IT-Convergence Engineering, Kumoh National Institute of Technology, Gyeongbuk, South Korea. His current research interests include the smart IoT convergence application, industrial wireless control network, UAV, metaverse, and blockchain.

GABRIEL AVELINO SAMPEDRO (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Mapúa University, Manila, Philippines, in 2018, and the Ph.D. degree in IT convergence engineering from the Kumoh National Institute of Technology, in 2023. Currently, he is the CEO of Philippine Coding Camp, a training institute that promotes technology education in collaboration with the Animo Laboratory Foundation, De La Salle University. His research interests focus on real-time systems, embedded systems, robotics, machine learning, and blockchain.