An International Journal of Optimization and Control: Theories & Applications
ISSN: 2146-0957 eISSN: 2146-5703
Vol.16, No.3, pp.928-947 (2026)
https://doi.org/10.36922/IJOCTA025480219

RESEARCH ARTICLE

TrustFCL: Task-grained model assessment and aggregation for secure federated continual learning

Xiaoning Wu[1], Siqi Du[1], Ke Qiu[1], Shilin Wen[2], Haiting Hou[1], Yuxiao Liu[1], Jianxin Zhao[1], Chi Harold Liu[1], and Rui Han[1*]

[1]Department of Computer Science, Beijing Institute of Technology, Beijing, China
[2]North Automatic Control Technology Institute, Taiyuan, Shanxi, China
568712145@qq.com, siqiqia@sina.com,3220231321@bit.edu.cn, 18811530859@163.com, houhaiting123@126.com, 3220215156@bit.edu.cn, jianxin.zhao@bit.edu.cn, chiliu@bit.edu.cn, hanrui@bit.edu.cn

ARTICLE INFO

Article History:
Received: November 30, 2025
Revised: February 13, 2026
Accepted: February 14, 2026
Published Online: April 28, 2026

Keywords:
Federated continual learning
Task-grained aggregation
Distributed optimization
Multi-agent control
Security

AMS Classification 2010:
26A33; 34A08; 35H15; 34K50 47H10; 60H10

ABSTRACT

Federated learning (FL) enables decentralized model training across devices, while federated continual learning (FCL) continuously adapts to evolving tasks (varying in categories, data distributions, or problem domains). Currently, FL security issues, such as model poisoning and privacy leaks, are undermining trust and reliability, prompting the development of defense methods. However, with evolving tasks, when a poisoning attack occurs, traditional FL defense methods only evaluate whether the latest model from each client is poisoned. Therefore, existing approaches face two limitations: (i) defense methods based on static tasks ensure security by restricting parameter mutations, which undermines the model's ability to adapt to new tasks and (ii) knowledge preservation mechanisms (e.g., parameters or gradients), crucial for alleviating forgetting, also introduce new vulnerabilities. Poisoned historical knowledge could be reused to launch attacks that are persistent and difficult to trace. To enable secure and adaptive FCL, this paper proposes TrustFCL, which proposes inter-task reliability and task knowledge reliability as both optimization constraints and control feedback. These reliabilities guide both local anti-forgetting training and global decentralized secure aggregation through a dynamic review committee. Furthermore, TrustFCL stores task reliability information on blockchain to prevent tampering and utilizes the consensus mechanism for privacy-preserving model aggregation. Evaluation results show that TrustFCL reduces the accuracy degradation by 37.1% compared to existing defense methods, and achieves a 17.8% improvement over FCL baselines.

1. Introduction

Federated learning (FL)[1] enables a collaborative training of a global model across distributed devices while preserving data locally. However, this privacy protection also creates trust issues among FL nodes, introducing security and traceability challenges. Security threats in FL can be broadly categorized as utility-centric and privacy-centric threats.[2] Utility-centric threats (e.g., data/model poisoning, backdoors)[3,4] directly compromise model accuracy, whereas privacy-centric attacks[5] exploit shared updates to reconstruct sensitive information. Traditional defenses equate anomalous model updates with malicious activity. However, the threats become more complex in federated continual learning (FCL), which combines FL with evolving tasks. Typically, a task is composed of multiple

*Corresponding Author

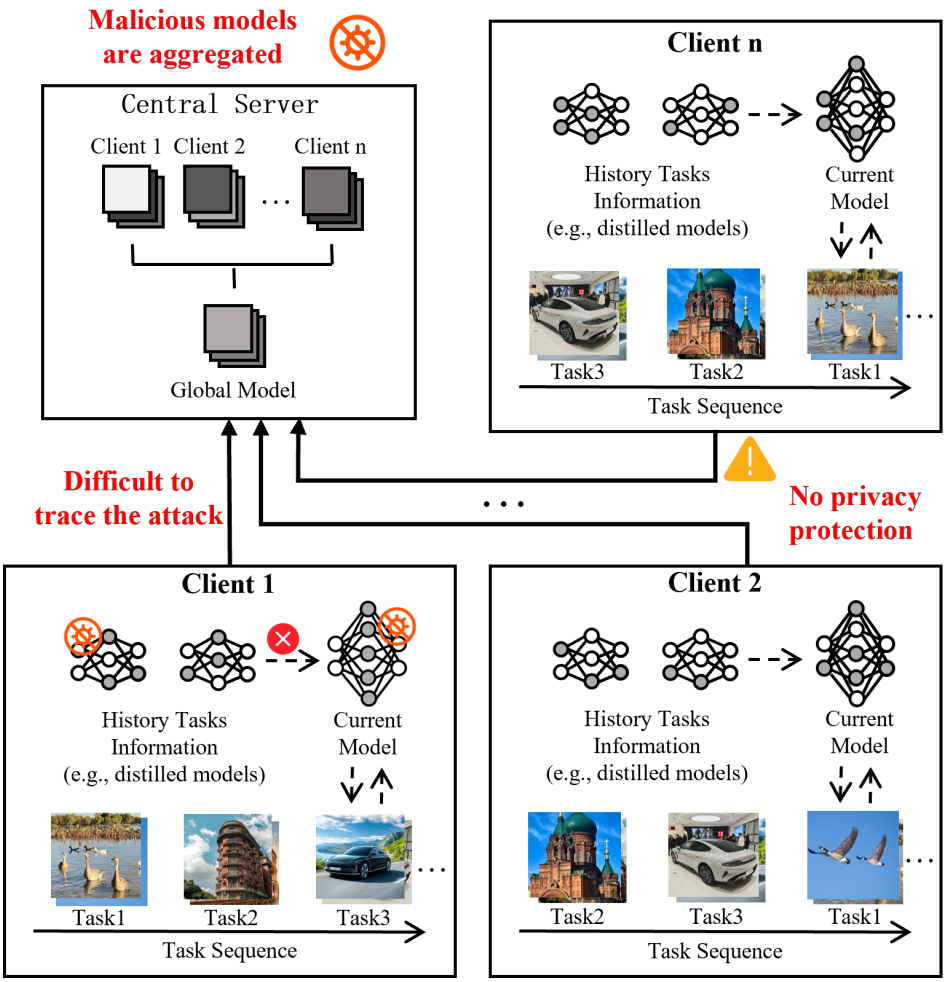

Figure 1. An example scenario of federated continual learning under attacks

classes/objects (e.g. different animals or vehicles).

Therefore, FCL requires models to incorporate knowledge preservation mechanisms to mitigate catastrophic forgetting.[6]

Scenario. In FCL, updates from new tasks may be falsely flagged as anomalous, while reused history knowledge could harbor undetected backdoors.[7] As shown in Figure 1, each client maintains a private task sequences, leading to inherently heterogeneous model update patterns. This inherent variability creates a detection blind spot for the central server. The server faces difficulty in distinguishing benign updates from new tasks and malicious updates. Consequently, malicious nodes can exploit this ambiguity to induce catastrophic forgetting or implant hidden backdoors. However, current FL defense methods only evaluate whether the latest model is poisoned, without considering interference or poisoning in historical task knowledge. Therefore, traditional defense approaches exhibit the following challenges when applied to FCL:

(i) Static defenses undermine adaptability With evolving tasks, conventional FL defenses hinder the model's ability to adapt to new and distinct tasks. When an honest client learns a new task, its model parameters change to adapt to the new data. From a static perspective, this update appears anomalous. For example, FLTrust[8] maintains a root dataset on the server side for evaluation, but this method cannot adapt to dynamically updating tasks. Defense methods based on clients' historical behavior[9] or trust scores[10] suffer from the same limitation. Besides, FCL across non-independent and identically distributed (non-IID) tasks causes natural model variations. These variations' similarity to poisoning attacks limits traditional detection methods relying on update consistency.

(ii) Catastrophic forgetting solutions introduce new vulnerabilities. To mitigate catastrophic forgetting, current FCL approaches create additional attack surfaces. Reusing old samples or dynamically expanding the network[13] offer knowledge preservation mechanisms. The storage and transmission of historical task

information (such as samples, gradients, or model parameters) provide new attack targets and increase privacy risks. Furthermore, these FCL methods create new attack vectors for malicious attacks. For example, class-aware gradient compensation loss (GLFC)[14] introduces a proxy server. However, existing defenses approaches are insensitive to threats correlated with task evolving.

This paper proposes TrustFCL, a task-grained optimization method for trusted FCL, integrating inter-task reliability and task knowledge reliability. By framing model aggregation as a constrained optimization problem with reliability constraints, TrustFCL not only mitigates catastrophic forgetting but also enables client evaluation at the task level. Building on this, we leverage blockchain-based federated learning (BC-FL) technology to securely store and verify critical task-level updates. This blockchain also integrates a dynamic review committee mechanism which ensures reliable model assessment and aggregation. The core contributions are as follows:

i A Reliability-based optimization method for local training. To mitigate catastrophic forgetting, we reformulate the local continual training process as an optimization problem with reliability constraints. With evolving tasks, historical task gradients serve as stability constraints. The strength of these constraints is dynamically adjusted by inter-task reliability, imposing restrictions for historical tasks deemed more vulnerable to interference.

ii A Control mechanism for robust federated aggregation. We design a reliability-based control mechanism for reliable model aggregation. A dynamically formed review committee acts as a distributed controller. It utilizes reliabilities to evaluate updates, adjust participant reputation via an integral feedback rule, filter potential malicious inputs, and execute weighted aggregation.

By integrating task reliability assessment with blockchain technology, TrustFCL ensures stable performance under malicious attacks and guarantees privacy security in FCL. The experimental results evaluated on CIFAR-100 datasets shows improved defense against malicious attacks. As the number of malicious clients increases, TrustFCL exhibits the smallest performance degradation and reduces the degradation by 37.1% compared to baselines. Compared to baseline methods in FCL scenarios, TrustFCL achieves the highest accuracy of 57%, representing an average improvement of 17.8%. For similar task aggregation processes, TrustFCL is 80.7% faster than conventional approaches. These experimental results confirm that TrustFCL works well in FCL scenarios, including security threats, catastrophic forgetting, and computational efficiency.

2. Background & related work

2.1. Federated learning

Federated learning leverages private data from multiple clients within a distributed architecture to collaboratively train a global model while preserving data privacy. Depending on the architectural design, FL can be categorized into centralized and decentralized paradigms. As a classical algorithm, FedAvg[1] aggregates local model parameters through a weighted averaging approach. Decentralized federated learning (DFL) eliminates the need for a central server. Gossip-based communication,[15] a classic decentralized paradigm used in FL, involves nodes selecting neighbors—either randomly or according to predefined rules—for information exchange.

2.2. Federated continual learning

In real-world scenarios, new categories and domains may continuously emerge. FL suffers catastrophic forgetting, which reduces performance for past categories. To combat forgetting, the concept of FCL was proposed. Current FCL approaches include experience replay, knowledge distillation, regularization, and framework modifications.[16] Nevertheless, FCL still faces many challenges, including: privacy concerns, computation burden, non-IID data, and so on.[17] The latest FCL methods still struggle to simultaneously resolve multiple challenges while maintaining usable performance. The balance between plasticity and forgetting is outlined as the stability-plasticity dilemma. Reusing old samples or dynamically expanding the network[13] can easily resolve this dilemma but incurs extra overhead. FedKD[18] proposed an adaptive mutual distillation framework to lower communication costs. However, FedKD does not address the non-IID data problem. GLFC[14] introduces a proxy server to alleviate the non-IID data issue but increases communication overhead.

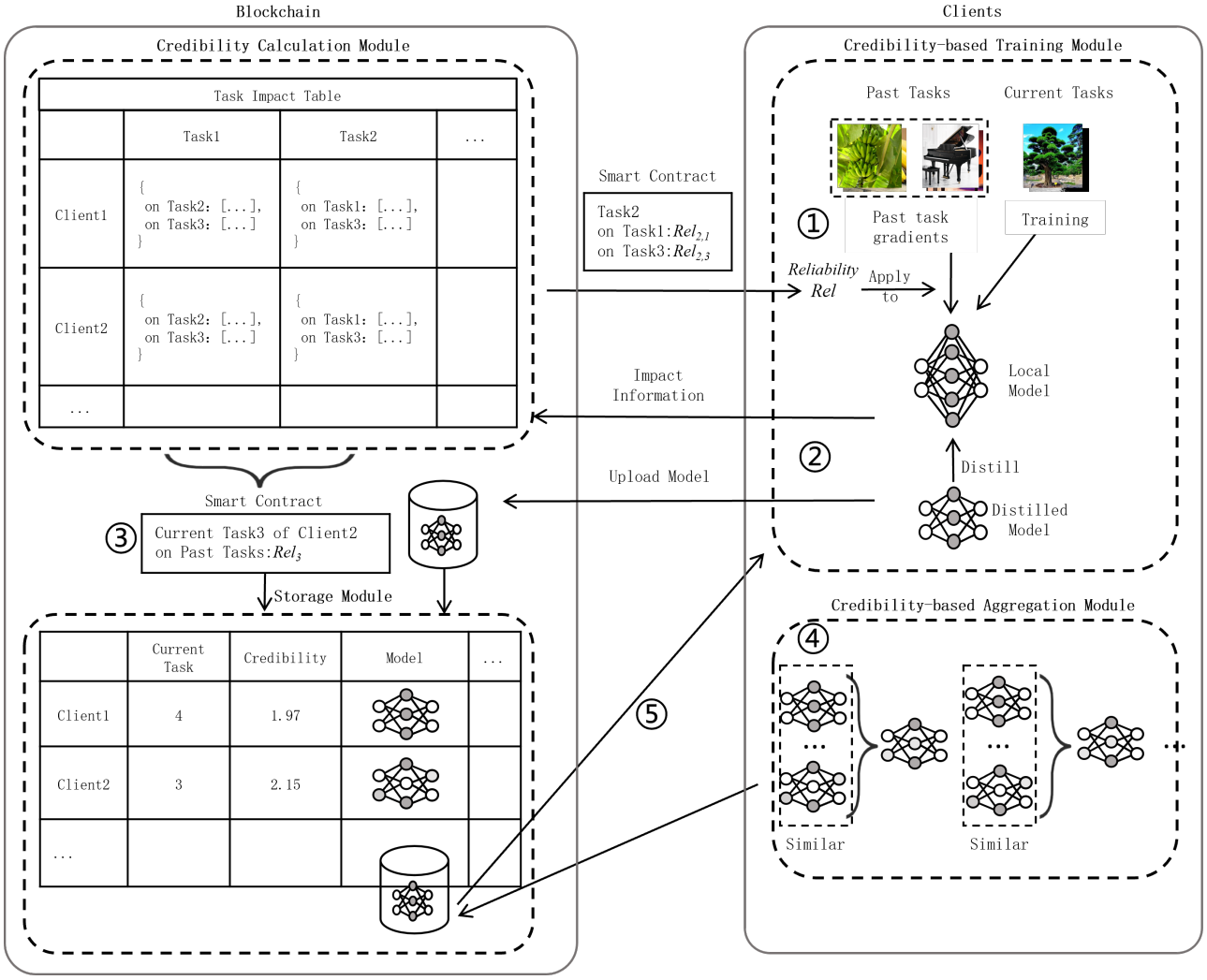

Figure 2. The architecture of TrustFCL

2.3. Blockchain-based federated learning

In practical deployments, FL needs to consider different data sources and face threats from both the client and server sides. Therefore, blockchain is chosen to provide a decentralized shared ledger with immutability, traceability, and decentralized trust. Current related research mainly includes:

i The architecture of blockchain. In fully decentralized BC-FL systems,[19–21] all nodes participate and need high computational and storage capabilities, while partially decentralized BC-FL systems[22,23] sacrifices some transparency for efficiency.

ii Reputation mechanisms. In BC-FL, blockchain serves as a reliable third-party ledger, recording key information about each node's reputation.[24–26]

iii Blockchain with other FL security methods. BC-FL is compatible with most defense methods such as secure aggregation[27] to further enhance robustness. BC-FL can incorporate these defense technologies into the system through smart contracts or designs specific consensus algorithms to prevent malicious activities.[28]

2.4. Federated learning defense

In FL, defending against malicious attacks is crucial for ensuring model security and reliability. Existing defense strategies are primarily developed from client-side and server-side perspectives. On the client side, approaches such as objective regularization (e.g., FedProx,[29]) optimizer modifications (e.g., Byzantine-robust SGD,[30]) and differential privacy mechanisms have been adopted. On the server side, defense efforts focus on improving the global aggregation process through robust aggregation

algorithms (e.g., geometric median,[11] Krum,[12] and client selection mechanisms based on trust scores,[10] contribution,[31] historical behavior,[9] or similarity metrics.[32] Furthermore, hybrid defense frameworks integrating technologies such as blockchain and secure multi-party computation are gaining attention. However, challenges remain in terms of generalization under non-IID data and coordination between client and server strategies.

Based on the aforementioned developments, it is evident that research on system reliability in the field of FCL is still in its early stages. Chen et al.[33] employ multi-task heads to mitigate catastrophic forgetting and enhance security by locally generating adversarial examples. Recent work by Chen et al.[34] ensure system integrity by combining malicious model detection with an improved Shapley value method to navigate the fairness–accuracy tradeoff. TrustFCL leverages blockchain reliability to execute an integral feedback rule, dynamically balancing participant reputation and model performance in evolving tasks. However, numerous unresolved security issues persist. Current defense methods based on the training phase do not account for forgetting and are incompatible with FCL, leading to significant declines in accuracy. In contrast, blockchain inherently possesses security attributes that do not interfere with the training process, flexibly enhancing the security of FCL. To date, no systematic research integrating blockchain with FCL has emerged. The potential applications of blockchain technology in this area remains untapped.

3. Method

3.1. Overview

We designed TrustFCL to continually train sequences of different tasks on decentralized federated clients safely. Blockchain was incorporated into TrustFCL to fulfill the requirements for traceability and evaluation mechanisms in FCL, and to share task-granularity information. Task-granularity refers to the distinct knowledge representation of a task i defined by its data distribution $\mathcal{D}_i$.

Additionally, blockchain's characteristics of tamper resistance, traceability, and distributed consensus provided support for FCL reputation system. As shown in the Figure 2, TrustFCL consists of four modules: reliability calculation module, training module, aggregation module, and storage module. Each client has its private task sequence and abstracts the model parameters of the locally trained model on a task as "task knowledge." The design of TrustFCL has three objectives:

(i) Lightweight and scalable. With limited communication resources, TrustFCL employed knowledge distillation to extract key knowledge and uploaded the distilled model as task knowledge to the blockchain. After the aggregation process, the newly aggregated distilled model was retrieved from the blockchain and used in subsequent training steps. Another lightweight design of TrustFCL is mainly reflected in the fact that only model metadata (such as hash, task identification, and reliability evaluation) was stored on-chain instead of the original parameters, and the consensus overhead was limited to local nodes through a committee mechanism.

(ii) Catastrophic forgetting prevention. On the client side, incremental training was performed on the local model that carries knowledge of old tasks. During this process, inter-task impact information shared by other clients on the blockchain was used to guide the model's gradient adjustment. This adjustment ensured the retention of critical old knowledge while learning new tasks.

(iii) Reliable transmission of task knowledge. Blockchain distributed ledger technology was adopted to store inter-task impact information, ensuring traceability and data integrity. Building on the blockchain, a multi-client consensus mechanism was used for model aggregation, ensuring transparency and verifiability.

3.2. Reliability-based calculation module

This module used the inter-task impact information shared by clients and executable code (smart contracts) on the blockchain to calculate inter-task reliability and task knowledge reliability. The results served as a basis for decision-making in subsequent local training and aggregation phases. The inter-task impact information included three types: positive impact (p), negative impact (n), and uncertain impact (u). During local training, clients tested the current local model using 10% sampled data from historical tasks. The rise and fall of accuracy determine the type of impact that the current task i has on historical task j. If the accuracy rate of the model on the historical task sampling set increases by more than the threshold ϵ after the update, it is classified as a positive impact (Positive); if it decreases by more than ϵ, it is classified as a negative impact (Negative); otherwise, it is regarded as uncertain (Uncertain). Clients recorded the number of impacts from task

i to task j as $n^p_{i\to j}$, $n^n_{i\to j}$, and $n^u_{i\to j}$, with n_{total} representing the sum of these three values.

Inter-task reliability measures the extent to which one task affects the knowledge retention of another task. It guides gradient updates during local client training, preventing the learning of new tasks from causing "catastrophic forgetting" of old tasks. The reliability $Rel_{i\to j}$ of task i (the current task) relative to task j (a historical task) is calculated as follows:

$$u_{i\to j} = \frac{n^u_{i\to j}}{n_{\text{total}}} \tag{1}$$

$$b_{i\to j} = (1 - u_{i\to j}) \cdot \frac{n^p_{i\to j}}{n^p_{i\to j} + n^n_{i\to j}} \tag{2}$$

$$Rel_{i\to j} = b_{i\to j} + \beta \cdot u_{i\to j} \tag{3}$$

Among these, $u_{i\to j}$ represents the uncertainty, $b_{i\to j}$ represents the confidence level, and $\beta \in [0, 1]$ is the coefficient representing the degree of uncertainty effect on reputation.[35] A larger value of $Rel_{i\to j}$ indicates that the influence of task i on task j is more positively inclined, suggesting that task i has less overlap with the knowledge of task j. The reliability of the current task relative to other tasks was selected as the reliability of the most recently uploaded task knowledge by the client. This value was used to assess the quality and reliability of the distilled model (i.e., task knowledge) submitted by a client. The review committee used this value to evaluate and filter the models before model aggregation, thereby identifying and defending against poisoning attacks.

From a control-theoretic perspective, $Rel_{i\to j}$ can be interpreted as a robustness margin. In a multi-agent system with persistent disturbances (i.e., poisoning attacks), ensuring $Rel_{i\to j}$ stays above a threshold is equivalent to maintaining the system's practical finite-time stability.[37] Drawing on the robust principles,

3.3. Reliability-based training module

As in Figure 3, the objective of training module was to execute the training of a new task while actively preserving the knowledge state of previously learned tasks.

First, a historical task replay mechanism was activated to reconstruct stability constraints. Samples from previous tasks were drawn from a local buffer, and their corresponding gradient set $\mathcal{G} = \{g_1, g_2, \ldots, g_{i-1}\}$ was obtained via backpropagation. Each g_k encodes the direction that reduces loss on the k-th old task. During training of the new task T_i using knowledge distillation, a raw gradient g_i was computed. Instead of applying it directly, a reliability-modulated gradient rectification step was performed, formulated as the following quadratic programming problem (solved using CVXOPT version 1.2.6):

$$\begin{aligned} \tilde{g} = \arg\min_{\tilde{g}} \frac{1}{2}\|\tilde{g} - g_i\|^2 \\ \text{subject to} \quad \tilde{g}^\top g_k \geq \left(\tfrac{1}{2} - Rel_{k\to i}\right)\|g_i\|\|g_k\|, \\ \forall g_k \in \mathcal{G}. \end{aligned} \tag{4}$$

Finally, the local model was updated using $\tilde{g}$, and the same replayed data generated inter-task impact factors (p, n, u). These factors were emitted as system feedback for the subsequent global reliability aggregation, closing the federated control loop.

3.4. Reliability-based aggregation module

As shown in Figure 4, the reliability-based aggregation module was governed by a decentralized review committee, a dynamically selected subset of clients. The review committee functions analogous to a multi-agent controller for reliability assessment and consensus-based aggregation. Meanwhile, inter-task similarity was computed to identify similar knowledge for each task.

The system operated in iterative rounds, each orchestrated by a review committee selected via a verifiable random function weighted by client reputation R_c, computed as $R_c = b + \beta \cdot u$. This committee executed three key functions:

i Model evaluation. Each member evaluated submitted model updates $\{W_i\}$ by computing pairwise task similarity $\text{Sim}_{i,j}$ using public samples and multi-class cross-entropy:

$$\text{Sim}_{i,j} = 1 - \mathcal{L}_{\text{MCE}}(Z_i, Z_j) \tag{5}$$

where Z_i, Z_j are outputs of W_i, W_j.

ii Reputation update. Each client started with an initial reputation $R_c = 5$. The committee adjusted this based on reliability rankings: R_c decreased by 1 if a client's update ranked in the bottom one-third for three consecutive rounds, and increased by 1 (capped at 5) if it remained above this threshold for three consecutive rounds. Nodes with $R_c = 0$ were flagged as malicious and excluded from committee elections and model aggregation. To ensure decentralization, committee members independently executed this logic, with

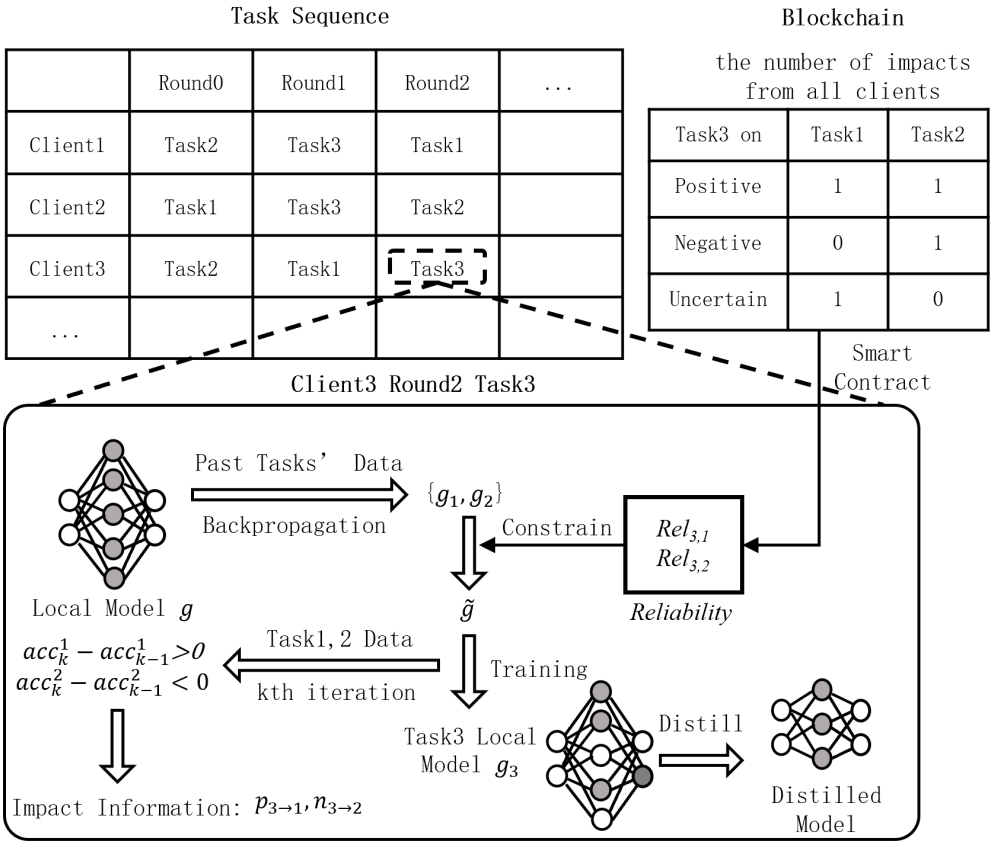

Figure 3. Training module

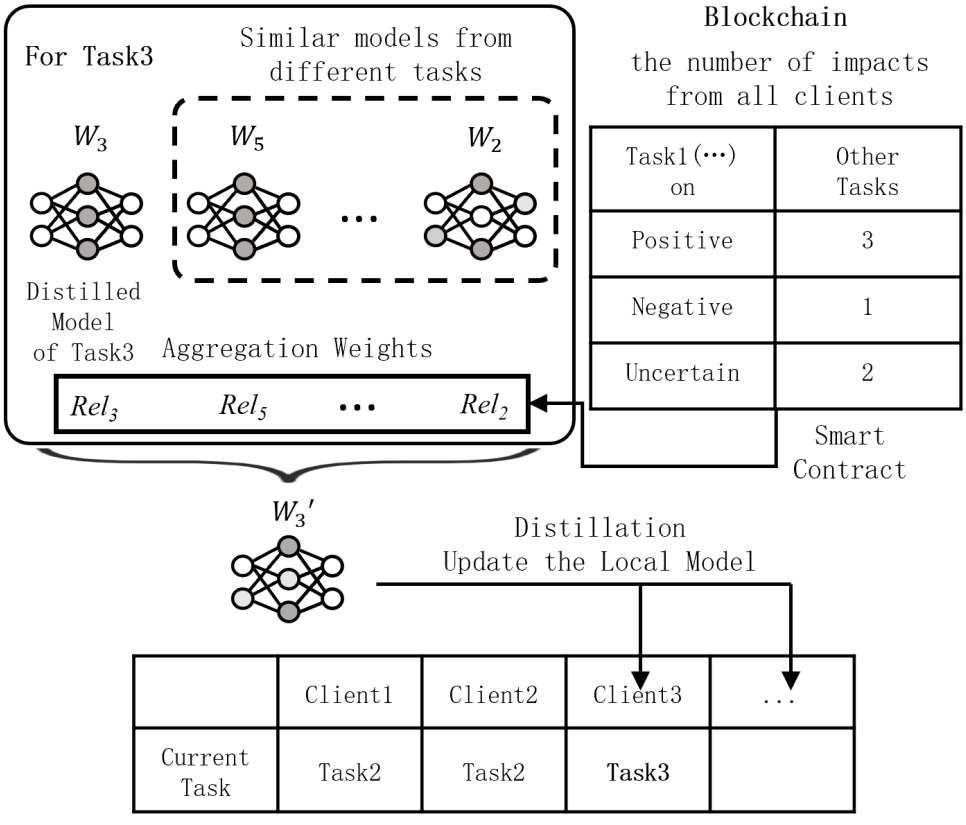

Figure 4. Aggregation module

the leader's results verified via smart contract-based consensus.

iii Consensus decision (aggregation and verification). For each task i, the committee selected the top-k most similar tasks, $\text{List}_i = \{d_1, \ldots, d_k\}$ based on $\text{Sim}_{i,j}$. The aggregated model W_i' was computed via reliability-weighted averaging:

$$W_i' = \frac{Rel_i \cdot W_i + \sum_{j=1}^{k} Rel_{d_j} \cdot W_{d_j}}{Rel_i + \sum_{j=1}^{k} Rel_{d_j}} \quad (6)$$

The committee was reconfigured each round via weighted random sampling, establishing a closed-loop control architecture. This reliability-weighted aggregation and knowledge selection jointly balanced model plasticity with stability through a constrained optimization objective.

Furthermore, this decentralized review committee was specifically designed to defend against long-term, cumulative poisoning attacks.[38] Unlike traditional FL defenses that focus on instantaneous anomalies, TrustFCL leverages the integral feedback rule in reputation updates. By comparing the current task knowledge W_i with the top-k most similar historical models $\{W_{d_j}\}$ stored on the blockchain, the committee can detect subtle "gradient drifts" that deviate from the benign knowledge evolution trajectory.

(iv) Communication efficiency and scalability. Unlike traditional model-grained FL that exchanges full weight matrices, TrustFCL significantly improves scalability by exchanging only task-specific parameters. This reduction in bandwidth consumption and block size ensures that the control channel remains unsaturated even as the number of networked agents increases. This design allows TrustFCL to maintain a stable communication overhead in dense edge environments.

3.5. Storage module and chang'an chain

This module consisted of a blockchain and was responsible for storing model information, inter-task impact information, task knowledge reliability, the aggregated models, and members' information of the review committee.

TrustFCL utilized Chang'an Chain to implement two key functions: (i) intelligent scheduling contract: deploys smart contracts that support FCL, records model information, maintains or dynamically adjusts client reputation, and selects review committeeand (ii) data authentication: leverages Chang'an Chain Software Development Kit (SDK) to store model update-related data on the blockchain. The tamper-resistant nature of the blockchain ensures data consistency and enables tracing of malicious nodes that submit poisoned data.

Specifically, the immutable and traceable nature of blockchain allows the system to roll back to an undamaged historical state. This capability provides an accountable infrastructure that enables the FCL system to comply with privacy regulations. The Chang'an Chain provides each client with a dedicated key-value store for their model data, evaluation records, and interaction history. For the upload process, a connection to the Chang'an Chain was first established via the SDK. The client authenticated its identity using its identification and key. Each upload operation undergoes permission verification through the smart contract, ensuring that only the client can modify its own data. For the query process, the smart contract implemented access control for data queries. Query requests were initiated via the SDK, and the smart contract verified the legitimacy of the request.

4. Evaluation

In this section, we conducted an experimental evaluation of the BC-FCL system and validated its committee-based consensus mechanism. We analyzed whether the system achieved improvements in the security of model data and the aggregation process.

4.1. Experimental settings

4.1.1. Testbed and deep neural network models

TrustFCL deployed 10 clients and 1 server on a cloud server equipped with 5 Neural Processing Units (Ascend 910B3). The cloud server ran on a Linux system (openEuler 22.03, aarch64 architecture), with 895 GB of memory, a 110-core Kunpeng-920 processor, and an ARM64 architecture. For computer vision (CV) and natural language processing (NLP) tasks, different deep neural network (DNN) models were deployed on the clients. The CV task models included two 6-convoluted neural network (CNN) models, two 10-CNN models, two DenseNet models, two ResNet-18 models, and two 6-layer ViT models. The NLP task models included two TextCNN models, two recurrent neural network models, three long short-term memory-series models, and three BERT models. For experiments involving 50 and 100 clients, the

Table 1. Training configuration.

Task type	Models	Datasets	Number of tasks
CV	6-CNN, 10-CNN, ResNet18, WideResNet, SixLayerViT	CIFAR-100	10
		Mini-ImageNet	10
		Tiny-ImageNet	20
NLP	TextCNN, RNN, Bert, LSTM, MoELSTM	ASC	19
		DSC	10

Abbreviations: ASC: Aspect sentiment classification; CNN: Convolutional neural network; CV: Computer vision; DSC: Document sentiment classification; LSTM: Long short-term memory; MoELSTM: Mixture of experts long short-term memory; NLP: Natural language processing; RNN: Recurrent neural network.

number of each CV task model was scaled up by 5 and 10, respectively.

4.1.2. Blockchain deployment

For the blockchain implementation, we employed the Chang'an Chain deployed on cloud servers. Chang'an Chain natively supports multi-replica deployment on a single server. Each blockchain replica listened for requests through distinct ports on the server. We deployed 10 blockchain replicas, each corresponding to one client. Client programs interacted with their respective blockchain replicas using unique user IDs and cryptographic keys.

4.1.3. Datasets and tasks

The CV datasets included CIFAR-100,[39] Mini-ImageNet,[?] and Tiny ImageNet.[41] The NLP datasets included aspect sentiment classification and document sentiment classification.[42] Following the setup of FedRep,[43] we configured the distribution of tasks and datasets to ensure heterogeneity across clients. Each client was assigned a random sequence of tasks, with each task containing 2 to 5 classes. Each class accounted for 5% to 10% of the training samples. Table 1 summarizes the datasets and model information for different applications.

4.1.4. Comparison of federated learning baselines

We implemented and compared our approach with 13 baselines, which can be categorized into three groups:

i FL methods for heterogeneous models:
- FedMD[1]: This method updates the distillation model using a public dataset during model aggregation.
- FedKD[44]: This method designs multiple distillation loss functions for different network architectures across clients.
- FedKEMF[43]: This method merges all teacher models during aggregation and leverages a public dataset to refine a superior global model on the server.
- FedGKT[45]: This method proposes an alternating minimization variant that trains edge models and transfers knowledge to a server model via distillation.

ii FCL methods:
- GLFC[14]: This method constructs multiple distillation losses to prevent forgetting and distills stable inter-class relationships across tasks.
- FedWEIT[18]: This method divides model weights into base and adaptive weights, storing the latter for easy retrieval of historical information.
- FedKNOW[46]: This method uses quadratic programming to compute gradients for recovering past task information.
- FedViT[47]: This method retains training samples from previous tasks and uses them to update the model for new tasks.
- TFCL[48]: This method adopts knowledge aggregation and model decomposition strategies to enable joint learning of similar tasks within clients.
- AFFCL[49]: This method utilizes generative models to estimate data distributions, combining feature replay and knowledge integration.

iii Clustered FL methods:
- CFL[50]: This method partitions clients into different clusters based on cosine similarity of gradients, then aggregates clients within the same cluster.

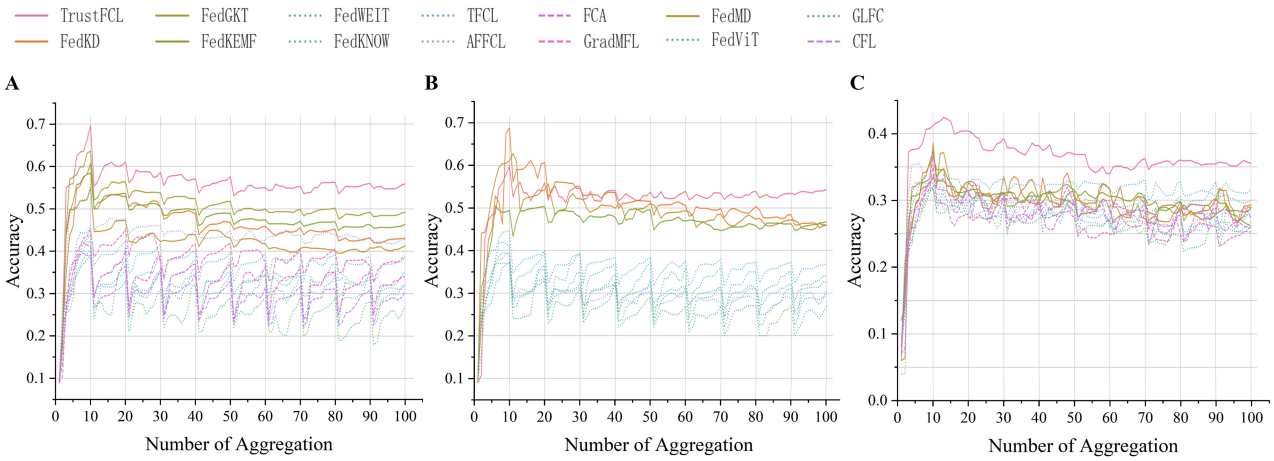

Figure 5. Accuracy comparison. TrustFCL vs. baselines on computer vision dataset. (A) CIFAR-100. (B) Mini-ImageNet. (C) TinyImageNet
Abbreviations: FL: Federated learning; FCL: Federated continual learning.

- FCA[51]: This method estimates cluster identities of clients and optimizes model parameters for each cluster accordingly.
- GradMFL[52]: This method introduces hierarchical clustering to organize clients and facilitates knowledge transfer across different levels.

4.1.5. Comparison of federated learning defense baselines

We selected seven defense algorithms in FL for comparison:

i Weak DP[53]: A privacy-preserving method that adds minimal noise to gradients, offering basic protection against small-scale attacks but often lowers model accuracy.
ii FL-wbc[54]: This technique filters malicious models by detecting persistent anomalous gradients and injecting targeted noise to neutralize attacks before aggregation.
iii SLSGD[55]: This method aggregates only the update direction, using the majority sign per parameter. It effectively resists outliers at the cost of slower convergence.
iv Krum[12]: This algorithm selects the gradient most consistent with others by choosing the one with the smallest total distance to all peers as the secure update.
v Coordinate-wise trimmed mean[56]: This method removes the highest and lowest values in each parameter dimension before aggregation to filter outliers.
vi Coordinate-wise median[56]: This median-based method aggregates each parameter dimension independently, providing effective defense against malicious clients.
vii CClip[57]: This method stabilizes updates by organizing gradients into buckets, averaging them, and applying clipping constraints.

4.1.6. Attacks of malicious nodes

For the attack strategies employed by malicious nodes, we selected two poisoning attacks from FedSecurity[58]:

i zero-gradient attack: this attack sets all gradient parameters to zero to halt progress and delay convergence,and
ii random-gradient attack: this attack injects controlled noise into gradients to misdirect updates, slowing convergence and reducing accuracy.

4.1.7. Hyper-parameters and evaluation metrics

The accuracy metric is the top-1 accuracy. In the FL scenario, the reported accuracy for task T_m is the average accuracy of all m learned tasks. For training iterations, FedKEMF, FedKD, and our proposed TrustFCL each underwent 5 local iterations and 5 global knowledge integration iterations per round. In contrast, FedMD consisted of 5 training iterations on the public dataset and 5 on the private dataset. All other methods were configured with 10 local training iterations per round. For each task, the number of communication rounds was fixed at

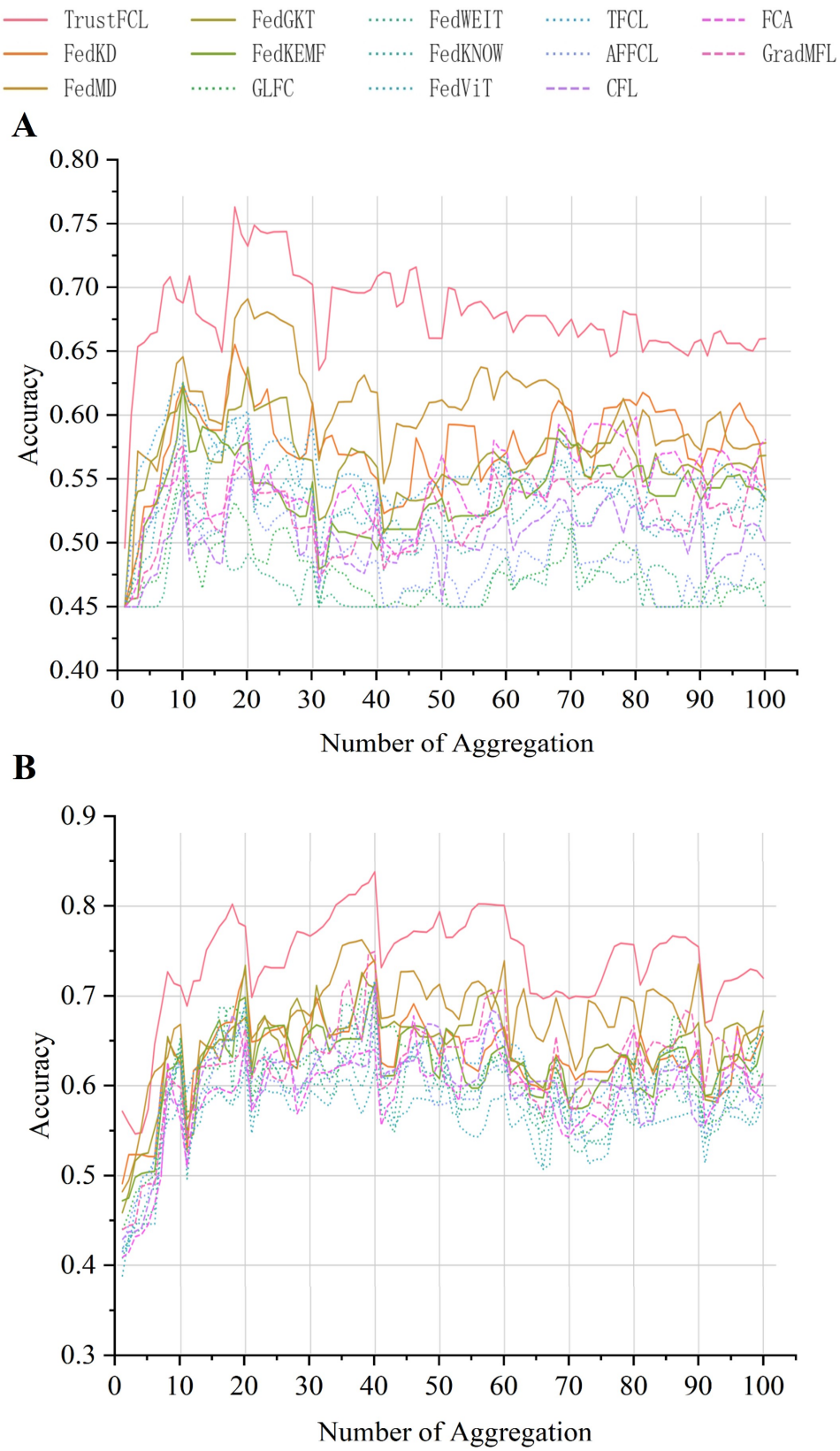

Figure 6. Accuracy comparison. TrustFCL vs. baselines on natural language processing dataset. (A) Aspect sentiment classification. (B) Document sentiment classification
Abbreviations: FL: Federated learning; FCL: Federated continual learning.

10. Specifically, clients performed multiple local iterations followed by an aggregation, repeating this cycle for 10 rounds. Regarding the committee mechanism, the committee size was scaled with the total number of clients: 3 for 10 clients, 11 for 50 clients, and 21 for 100 clients.

For baseline hyperparameter optimization, the search space boundaries were defined as (0.5×, 2×) of the original evaluated values. For FCA, the initial cluster count was set to 3. For TrustFCL, the search spaces for the number of candidate models (h) and the top-k most similar models during aggregation were defined as 10, 20, 40 and 5, 10, 15, respectively.

4.2. Comparative evaluations

This section evaluates the model accuracy of TrustFCL compared to 13 baseline techniques on both CV and NLP datasets. We also compared the time consumption for searching similar task knowledge between TrustFCL and FCL GradMFL.

4.2.1. Accuracy comparison

As shown in Figure 5, the six FCL methods (GLFC, FedWEIT, FedKNOW, FedViT, TFCL, AFFCL), which are designed to aggregate models with identical network architectures, achieved the lowest accuracy. The three clustered FL methods (CFL, FCA, GradMFL), which aggregate local models from similar tasks, exhibited lower accuracy in scenarios involving a large number of tasks or complex tasks. Figures 5(A), 5(B), and 6(A) show that FedKD, FedMD, FedGKT, and FedKEMF achieved higher accuracy. This is because they utilize knowledge distillation to aggregate models with heterogeneous network structures. The proposed TrustFCL uses knowledge distillation to integrate heterogeneous knowledge and explicitly mitigates negative transfer. Therefore, TrustFCL achieved the highest final accuracy across all datasets. Among them, on CIFAR-100, TrustFCL attained a accuracy of 57%, representing an average improvement of 17.8% compared to baselines.

4.2.2. Similar task search time

Table 1 illustrates the search time of TrustFCL and CFL GradMFL for similar task knowledge. The search covered 10 tasks across varying total volume of stored knowledge. The result shows that TrustFCL reduced search time by a factor of 5. This improvement organized tasks in an index tree, which provided logarithmic time complexity during queries. This approach reduced search time compared to the traversal-based method used by CFL GradMFL. The time savings are considerable in large knowledge bases or over many aggregation rounds.

Table 2. Search time for task knowledge (seconds).

Knowledge volume	**TrustFCL**	**CFL GradMFL**
100	0.13	0.15
500	0.22	0.63
1000	0.34	1.17
1500	0.47	1.85
2000	0.56	2.31
2500	0.65	3.22
3000	0.74	3.83

Abbreviations: CFL: Clustered federated learning; FL: Federated learning; FCL: Federated continual learning; GradMFL: Gradient memory-based federated learning.

4.2.3. Latency induced by blockchain

To quantify the impact of blockchain, we evaluated the system's response latency and throughput across varying network scales (Figure 7). The results indicate that the consensus latency remained consistently below one second when the number of participating nodes was under 15, which is marginal compared to the local gradient training time in FCL. As the node count reached 17, the system throughput acheived a stable saturation plateau, demonstrating sufficient capacity to handle concurrent periodic weight synchronization requests. This empirical evidence confirms that the security-enhanced blockchain backbone introduced acceptable overhead while maintaining high system scalability for decentralized learning tasks.

4.3. Client scalability assessment

This subsection explores TrustFCL's scalability with respect to client population and its impact on the FCL system. We compared the accuracy of TrustFCL, FedWEIT, and AFFCL with client numbers set to 50 and 100.

Figure 8 presents the accuracy of TrustFCL, FedWEIT, and AFFCL when trained on the CIFAR-100 dataset with 50 and 100 clients. Compared with the results in Figure 5(A), it shows that as the number of clients increased, the accuracy of all methods declined. This occurs because a fixed training dataset was divided among more clients, reducing the data per client. Notably, TrustFCL maintained higher accuracy

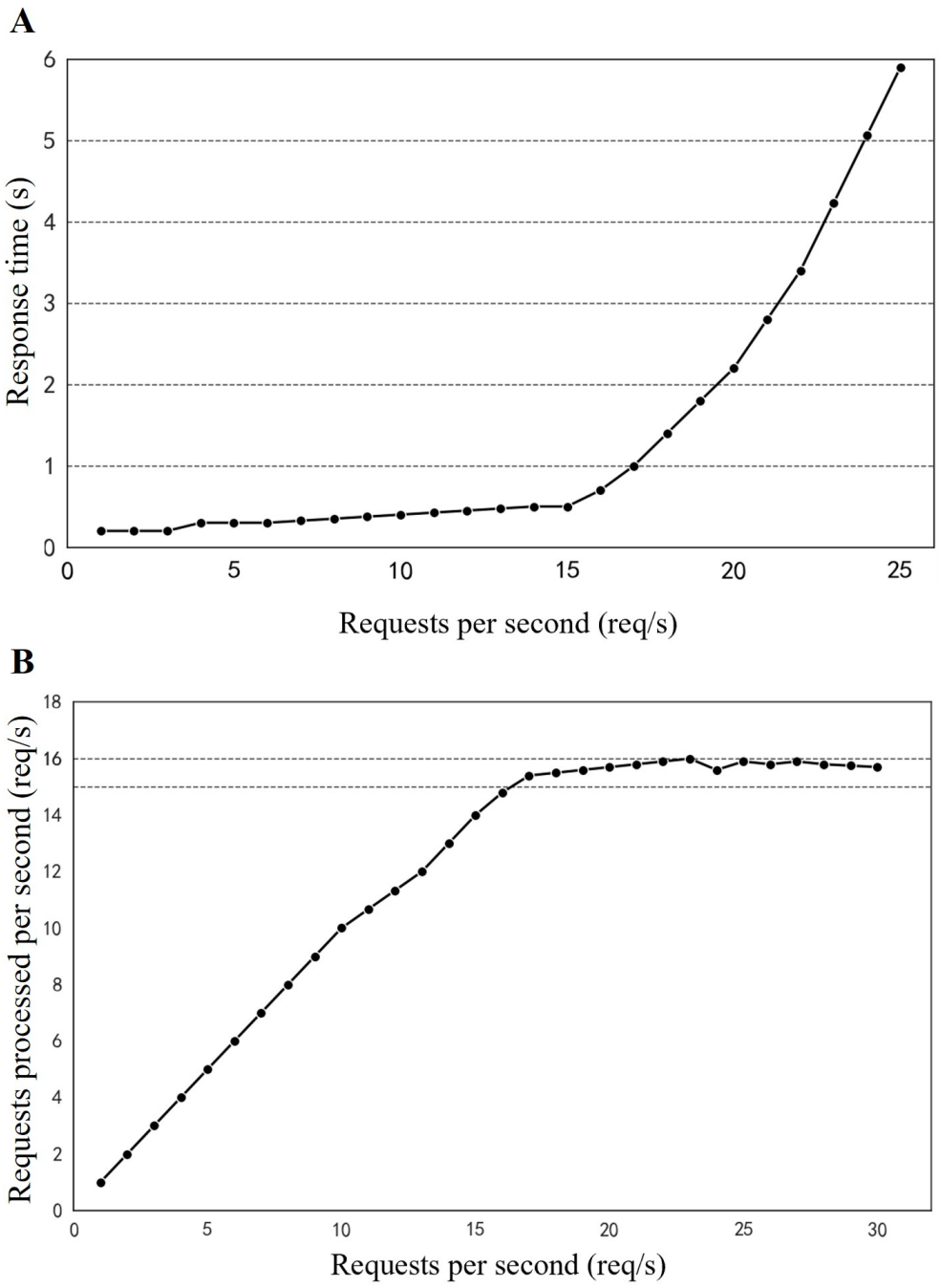

Figure 7. Evaluation of Blockchain-induced overhead. (A) Response speed test. (B) Throughput test.

than the other two methods as the number of clients increases. Furthermore, when the client count was large, AFFCL demonstrated better performance compared to FedWEIT. This can be attributed to AFFCL's use of generative replay technology.[49]

4.4. Systemic reliability assessment

In this section, we evaluated the FCL system under the presence of malicious clients on CIFAR-100. The identification and adjustment of potentially malicious model submissions were managed by the committee mechanism in TrustFCL. To experimentally validate the resilience of the committee mechanism against malicious attacks, we implemented the committee mechanism alongside other defense algorithms on FedKD[44] and conducted comprehensive experimental evaluations.

4.4.1. Setting of malicious node

Ten honest clients participated in the training process of the FL system. Malicious nodes were incrementally added beyond these 10 honest clients. Malicious nodes submitted incorrect

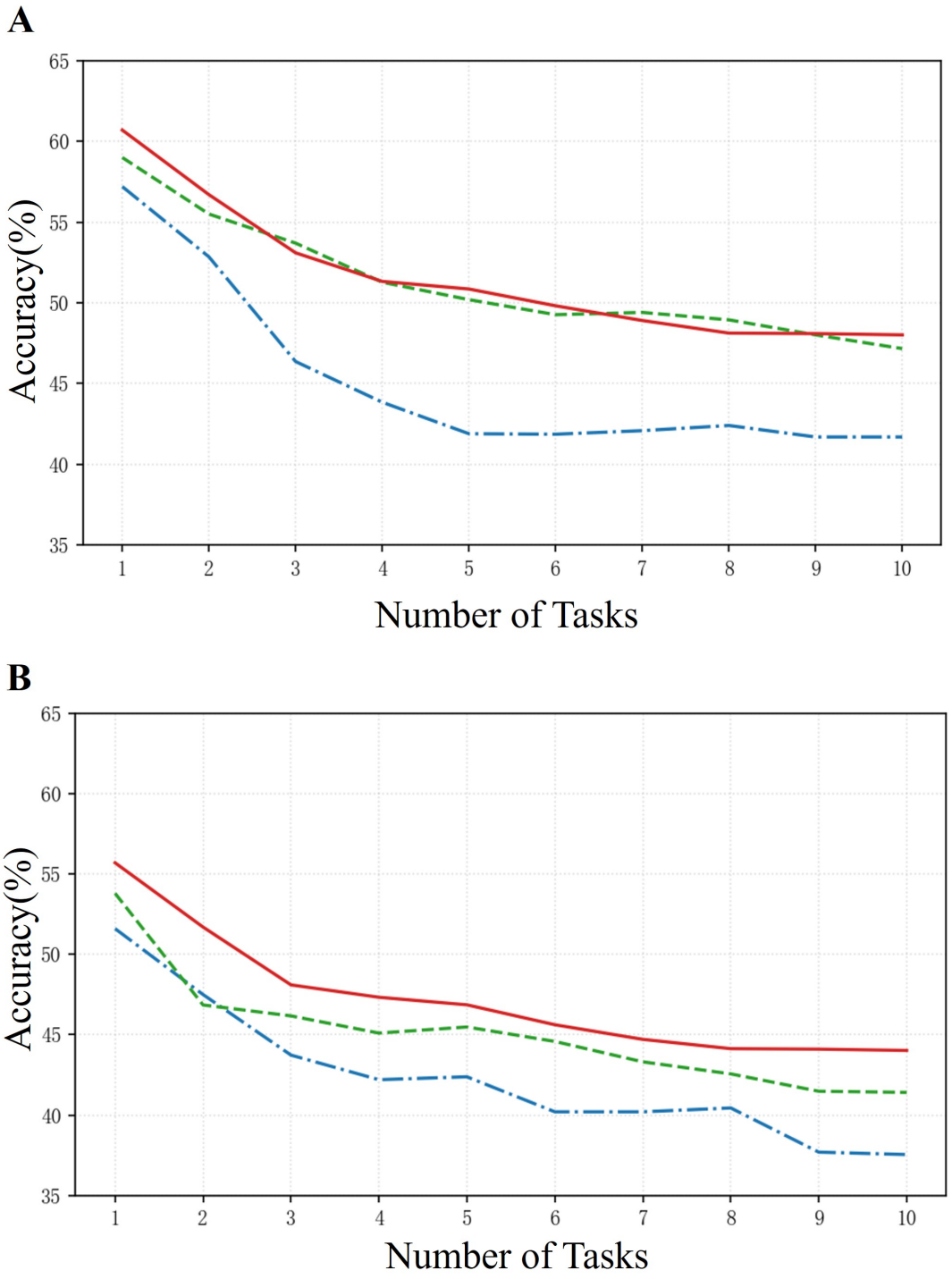

Figure 8. Accuracy with 50 and 100 clients. (A) 50 clients. (B) 100 clients.

models during the distillation model submission phase to mislead aggregation. For calculating the average accuracy of clients after training, only the accuracy of the 10 honest clients was used.

4.4.2. Comparison under random gradient attacks

As shown in Figure 9(A), as malicious nodes increased from 0 to 5, the accuracy of systems without TrustFCL's committee mechanism dropped from 53.6% to 48.7%. In contrast, systems with the committee mechanism showed strong resilience, declining only to 51.8%. Figure 9(B) illustrates the accuracy of the FedKD FL system under random gradient attacks. With malicious nodes increasing from 0 to 5, the system without the committee mechanism showed a 10.2% accuracy drop. By contrast, the system

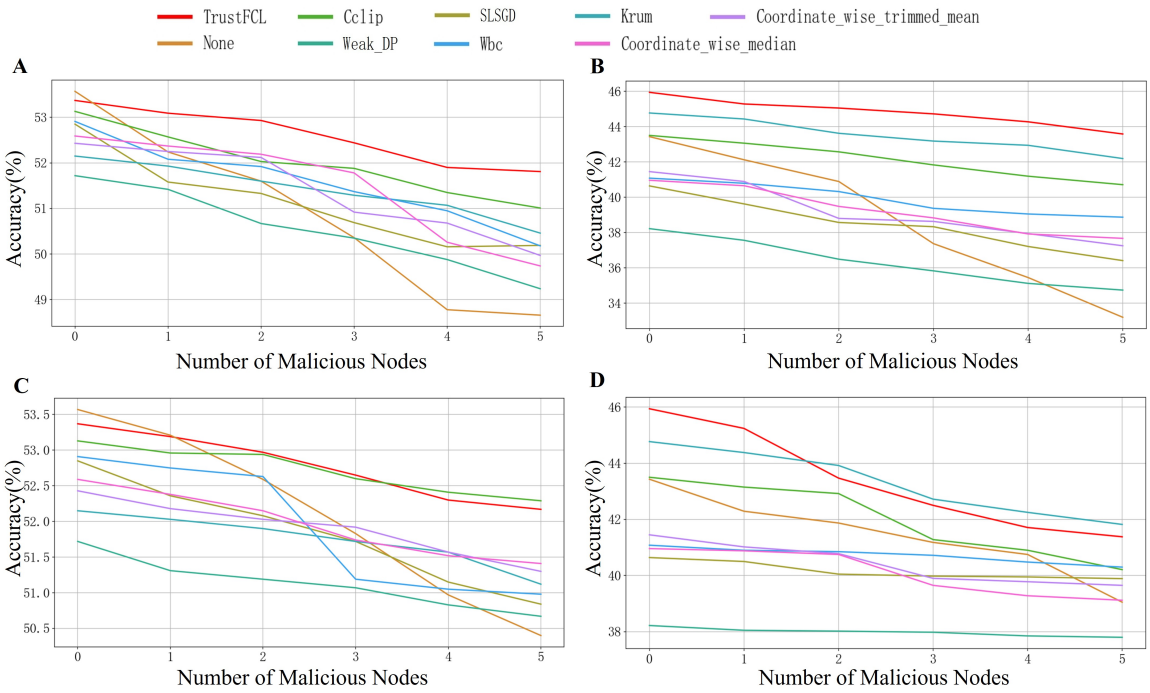

Figure 9. Accuracy comparison. TrustFCL vs. baselines under two attacks. (A) Accuracy in TrustFCL under random gradient attack. (B) Accuracy in FedKD under random gradient attack. (C) Accuracy in TrustFCL under zero-gradient attack. (D) Accuracy in FedKD under zero-gradient attack.

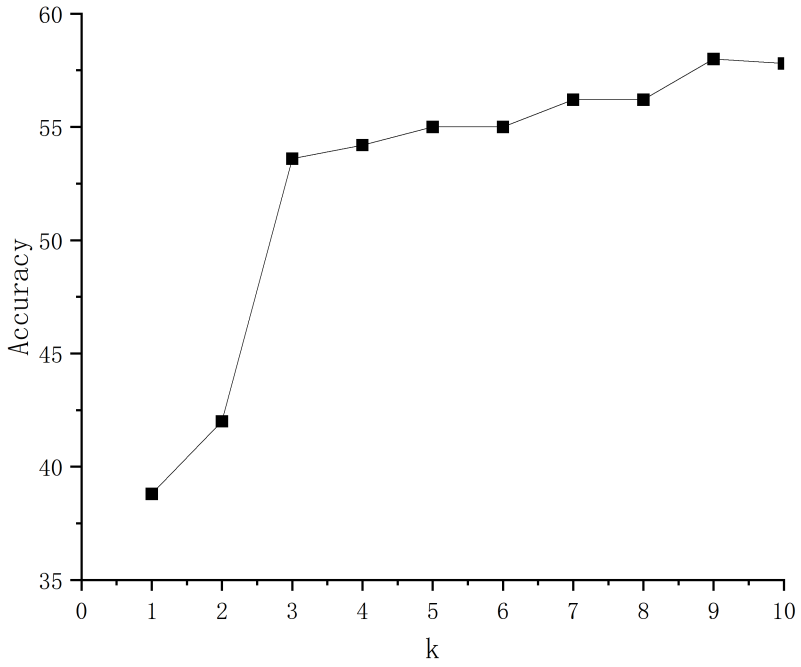

Figure 10. Impact of committee size k on system robustness

with the mechanism experienced a smaller decline of 2.3%. This demonstrates that TrustFCL's committee mechanism mitigated the negative impact of malicious nodes on model aggregation and subsequent training.

When comparing TrustFCL with other defense algorithms, TrustFCL achieved the highest accuracy in Figure 9(A) and (B). In contrast, weak DP introduced noise into the global model, which reduces accuracy while providing limited attack protection, resulting in the lowest accuracy. Other methods such as Krum and CClip performed relatively well. Krum aggregated only similar models, while CClip effectively clipped outlier values, contributing to their higher accuracy rates.

4.4.3. Comparison under random gradient attacks

Figure 9(C) and (D) present the accuracy of the TrustFCL and FedKD system respectively under zero-gradient attacks. The zero-gradient attacks caused less accuracy loss than the random-gradient attacks, as it only delays convergence rather than misleading the model. For instance, under random gradient attacks with no defense (Figure 9(A), TrustFCL's accuracy decreased by 4.9% as malicious nodes increase from 0 to 5. In contrast, under zero-gradient attacks without defense (Figure 9(C), the accuracy drop was only 3.2%.

When comparing TrustFCL with other defense algorithms, weak DP yielded the lowest accuracy, as it introduced noisy updates but failed to filter zero-gradient attacks. In some cases, weak DP's accuracy fell below that of having no defense mechanism. In Figure 9(D), with 5 malicious nodes, weak DP achieved 37.8% accuracy, compared to 39.05% without any defense. Other defense methods demonstrated effective resistance to zero-gradient attacks. Among them, TrustFCL's committee mechanism and Krum achieved the highest accuracy.

Overall, under malicious attacks, TrustFCL showed improved accuracy compared to unprotected FCL and most FL defense methods. As the number of malicious clients increased, TrustFCL exhibited the smallest performance degradation. The average performance degradation of other defense methods was 1.59 times that of TrustFCL, meaning TrustFCL reduced the degradation by 37.1% compared to baselines.

Table 3. Performance (final accuracy %) on Task 1 under targeted degradation attack (targeting Task 1 during Task 3 training).

Method	**Task 1 (Clean)**	**Task 1 (Attack)**
Krum	48.0	35.2
CClip	51.8	40.8
TrustFCL	52.8	49.0

Abbreviation: FCL: Federated continual learning.

4.4.4. Resistance to targeted degradation attack

To further evaluate the robustness of TrustFCL, we simulated a targeted forgetting attack on the CIFAR-100 dataset. In this scenario, 2 out of 10 clients were designated as malicious attackers. These attackers submitted poisoned updates specifically designed to degrade the performance of Task 1 while appearing normal on the current task (Task 3). Table 3 shows that baseline methods (Krum and CClip) exhibited a sharp decline (above 10%) in Task 1 accuracy. In contrast, TrustFCL maintained stable performance (only drop 3.8%). This is because our committee-based assessment identified the negative impact on historical task buffers, effectively filtering out these targeted attacks through the defense mechanism.

4.4.5. Ablation study

We conducted an ablation study with 10 clients,varying k from 1 to 10. The results in Figure 10 reveal that $k = 3$ constitutes a critical threshold, effectively mitigating fundamental malicious threats. While increasing k offers marginal security gains, it triggered excessive latency in resource constrained edge clusters. Consequently, $k = 3$ achieved the optimal trade-off between security guarantees and communication efficiency.

5. Conclusion

This paper designed and implemented TrustFCL, a blockchain-based trusted FCL system. By integrating distributed blockchain technology and a committee mechanism, the system effectively mitigated issues of catastrophic forgetting and trustworthiness in dynamic task environments. Experiments indicate that the system achieved improvements in both attack resistance and accuracy compared to baseline methods. This work delivers a reliable privacy-preserving solution for machine learning in edge computing

environments. Future enhancements via meta-learning and anomaly detection could further improve adaptability.

Acknowledgments

None.

Funding

This paper was supported by National Natural Science Foundation of China (Grant No. 62272046, 62132019, 62302039,61872337), and the Special Program for High-Quality Development of the Ministry of Industry and Information Technology (No. CEIEC-20240), a Cooperative Project with the Northern Institute of Automatic Control Technology.

Conflict of interest

The authors declare they have no competing interests.

Author contributions

Conceptualization: Rui Han, Chi Harold Liu, Shilin Wen
Formal analysis: Haiting Hou, Yuxiao Liu, Shilin Wen
Investigation: Xiaoning Wu, Siqi Du, Ke Qiu
Methodology: Xiaoning Wu, Siqi Du, Ke Qiu
Writing – original draft: Xiaoning Wu, Rui Han, Jianxin Zhao
Writing – review & editing: Siqi Du, Chi Harold Liu

Availability of data

Not applicable.

AI tools statement

All authors confirm that no AI tools were used in the preparation of this manuscript.

References

1. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. Paper presented at: proceedings of the 20th International Conference on Artificial Intelligence and Statistics; April 20-22, 2017; Fort Lauderdale, FL, USA. Accessed March 17, 2026. https://proceedings.mlr.press/v54/mcmahan17a.html
2. Usynin D, Ziller A, Makowski M, et al. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. *Nat Mach Intell.* 2021;3(9):749-758. https://www.doi.org/10.1038/s42256-021-00390-3
3. Zhang J, Chen B, Cheng X, Binh HTT., Yu S. PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet Things J.* 2020;8(5):3310-3322. https://www.doi.org/10.1109/jiot.2020.3023126
4. Fang M, Cao X, Jia J, Gong NZ. Local model poisoning attacks to Byzantine-robust federated learning. In: Proc of the 29th USENIX Security Symposium (USENIX Security 20); August 12-14, 2020; Virtual. USENIX Association; 2020:1605-1622. Available from https://www.usenix.org/system/files/sec20-fang.pdf
5. Ren H, Deng J, Xie X. Grnn: generative regression neural network—a data leakage attack for federated learning. *ACM Trans Intell Syst Technol.* 2022;13(4):1-24. https://www.doi.org/10.1145/3510032
6. Robins A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connect Sci.* 1995;7(2):123-146. https://www.doi.org/10.1080/09540099550039318
7. Wang H, Sreenivasan K, Rajput S, et al. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. arXiv. Preprint online July 9, 2020. https://www.doi.org/10.48550/ARXIV.2007.05084
8. Cao X, Fang M, Liu J, Gong NZ. Fltrust: Byzantine-robust federated learning via trust bootstrapping. Paper presented at: The Network and Distributed System Security Symposium 2021; February 21-25, 2021; Virtual. Accessed March 17, 2026. https://www.ndss-symposium.org/ndss-paper/fltrust-byzantine-robust-federated-learning-via-trust-bootstrapping/
9. Mao Y, Yuan X, Zhao X, Zhong S. Romoa: Robust model aggregation for the resistance of federated learning to model poisoning attacks. *Proc Eur Symp Res Comput Secur.* 2021;476-496. https://www.doi.org/10.1007/978-3-030-88418-5_23
10. Geng G, Cai T, Yang Z. Better safe than sorry: Constructing byzantine-robust federated learning with synthesized trust. *Electronics.* 2023;12(13):2926. https://www.doi.org/10.3390/electronics12132926
11. Chen Y, Su L, Xu J. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proc ACM Meas Anal Comput Syst.* 2017;1(2):1-25. https://www.doi.org/10.1145/3154503
12. Blanchard P, El Mhamdi EM, Guerraoui R, Stainer J. Machine learning with adversaries: Byzantine tolerant gradient descent. *Adv Neural Inf Process Syst.* 2017;30:118-128. https://www.doi.org/10.5555/3294771.3294783
13. Nikishin E, Oh J, Ostrovski G, et al. Deep reinforcement learning with plasticity injection. *Adv Neural Inf Process Syst.* 2023;36:37142-37159. https://www.doi.org/10.5555/3666122.3667736

14. Dong J, Wang L, Fang Z, et al. Federated class-incremental learning. arXiv. Preprint online March 22, 2022. https://www.doi.org/10.48550/arXiv.2203.11473
15. Hegedűs I, Danner G, Jelasity M. Gossip learning as a decentralized alternative to federated learning. *Proc IFIP Int Conf Distrib Appl Interoperable Syst.* 2019;74-90. https://www.doi.org/10.1007/978-3-030-22496-7_5
16. Hamedi P, Razavi-Far R, Hallaji E. Federated continual learning: concepts, challenges, and solutions. *Neurocomputing.* 2025;651(C). https://www.doi.org/10.1016/j.neucom.2025.130844
17. Birashk A, Khan L. Federated continual learning for task-incremental and class-incremental problems: A survey. *Expert Syst Appl.* 2025;129278. https://www.doi.org/10.1016/j.eswa.2025.129278
18. Yoon J, Jeong W, Lee G, Yang E, Hwang SJ. Federated continual learning with weighted inter-client transfer. Paper presented at: proceedings of the 38th International Conference on Machine Learning; July 18-24, 2021; Virtual. Accessed March 17, 2026. https://proceedings.mlr.press/v139/yoon21b.html
19. Li J, Shao Y, Wei K, et al. Blockchain assisted decentralized federated learning (BLADE-FL): Performance analysis and resource allocation. *IEEE Trans. Parallel Distrib Syst.* 2021;33(10):2401-2415. https://www.doi.org/10.1109/tpds.2021.3138848
20. Li J, Shao Y, Wei K, et al. Blockchain Assisted Decentralized Federated Learning (BLADE-FL): Performance Analysis and Resource Allocation. *IEEE Trans Parallel Distrib Syst.* 2022;33(10):2401-2415. https://www.doi.org/10.1109/tpds.2021.3138848
21. Singh S, Rathore S, Alfarraj O, Tolba A, Yoon B. A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology. *Future Gener Comput Syst.* 2022;129:380-388. https://www.doi.org/10.1016/j.future.2021.11.028
22. Lu Y, Huang X, Zhang K, Maharjan S, Zhang Y. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Trans Veh Technol.* 2020;69(4):4298-4311. https://www.doi.org/10.1109/TVT.2020.2973651
23. Ma C, Li J, Shi L, et al. When federated learning meets blockchain: A new distributed learning paradigm. *IEEE Comput Intell Mag.* 2022;17(3):26-33. https://www.doi.org/10.1109/mci.2022.3180932
24. Qi J, Lin F, Chen Z, Tang C, Jia R, Li M. High-quality model aggregation for blockchain-based federated learning via reputation-motivated task participation. *IEEE Internet Things J.* 2022;9(19):18378-18391. https://www.doi.org/10.1109/jiot.2022.3160425
25. Gao L, Li L, Chen Y, Xu CZ, Xu M. FGFL: a blockchain-based fair incentive governor for Federated Learning. *J. Parallel Distrib Comput.* 2022;163:283-299. https://www.doi.org/10.1016/j.jpdc.2022.01.019
26. Guo S, Zhang K, Gong B, et al. Sandbox computing: A data privacy trusted sharing paradigm via blockchain and federated learning. *IEEE Trans Comput.* 2022;72(3):800-810. https://www.doi.org/10.1109/tc.2022.3180968
27. Li Z, Yu H, Zhou T, et al. Byzantine resistant secure blockchained federated learning at the edge. *IEEE Netw.* 2021;35(4):295-301. https://www.doi.org/10.1109/mnet.011.2000604
28. Abdel-Basset M, Moustafa N, Hawash H. Privacy-preserved cyberattack detection in Industrial Edge of Things (IEoT): A blockchain-orchestrated federated learning approach. *IEEE Trans Ind Inform.* 2022;18(11):7920-7934. https://www.doi.org/10.1109/tii.2022.3167663
29. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. *Proc Mach Learn Syst.* 2020;2:429-450. https://proceedings.mlsys.org/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html
30. Roberts L, Smyth, E. A simplified convergence theory for Byzantine resilient stochastic gradient descent. *EURO J Comput Optim.* 2022;10:100038. https://www.doi.org/10.1016/j.ejco.2022.100038
31. Chen B, Fan L, Zeng X, Gu M, Zhou, J. A contribution-aware federated framework for electric vehicle batteries health estimation. *IEEE Internet Things J.* 2024. https://www.doi.org/10.1109/jiot.2024.3524005
32. Gupta A, Luo T, Ngo MV, Das SK. Long-Short History of Gradients Is All You Need: Detecting Malicious and Unreliable Clients in Federated Learning. In: Lecture Notes in Computer Science. Springer Nature Switzerland. 2022:445-46 https://www.doi.org/10.1007/978-3-031-17143-7_22
33. Chen C, Li P, Sakurai K, et al. A multi-head federated continual learning approach for improved flexibility and robustness in edge environments. *Int J Netw Comput.* 2024;14(2):123-144. https://www.doi.org/10.15803/ijnc.14.2_123
34. Chen L, Zhao D, Tao L, et al. A credible and fair federated learning framework based on blockchain. *IEEE Trans Artif Intell.* 2024;6(2):301-316. https://www.doi.org/10.1109/tai.2024.3355362
35. Kang J, Xiong Z, Niyato D, Xie S, Zhang J. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet Things J.* 2019;6(6):10700-10714. https://www.doi.org/10.1109/jiot.2019.2940820

36. Zouari F, Mahmud M. Neural Network-Based Robust Adaptive Output Feedback Control for MIMO Time-Varying Delay Systems. Paper presented at: the Global Conference on Applications of Artificial Intelligence and Data Science 2024; July 22-26, 2024; Munich, Germany, and virtually. Accessed March 17, 2026. https://www.doi.org/10.1007/978-3-031-98498-3_5
37. Boulkroune A, Boubellouta A, Bouzeriba A, Zouari F. Practical finite-time fuzzy synchronization of chaotic systems with non-integer orders: Two chattering-free approaches. *J Syst Sci Syst Eng.* 2025;34(3):334-359. https://www.doi.org/10.1007/s11518-024-5635-7
38. Jiang Y, Shen J, Liu Z, Tan CW, Lam KY. Towards efficient and certified recovery from poisoning attacks in federated learning. *IEEE Trans Inf Forensics Secur.* 2025. https://www.doi.org/10.1109/tifs.2025.3533907
39. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. *Univ Toronto*; 2009. Available from https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf
40. Vinyals O, Blundell C, Lillicrap T, et al. Matching networks for one shot learning. *Adv Neural Inf Process Syst.* 2016;29. https://www.doi.org/10.5555/3157382.3157504
41. Le Y, Yang X. Tiny imagenet visual recognition challenge. *Stanford Univ CS231N*; 2015. Available from https://cs231n.stanford.edu/reports/2015 /pdfs/yle_project.pdf
42. Ke Z, Liu B, Wang H, Shu L. Continual learning with knowledge transfer for sentiment classification. *Proc Jt Eur Conf Mach Learn Knowl Discov Databases.* 2020;683-698. https://www.doi.org/10.1007/978-3-030-67664-3_41
43. Collins L, Hassani H, Mokhtari A, Shakkottai S. Exploiting shared representations for personalized federated learning. Paper presented at: proceedings of the 38th International Conference on Machine Learning. 2021;18-24; Virtual. Available from https://proceedings.mlr.press/v139/collins21a.html
44. Deng Y, Kamani MM, Mahdavi M. Adaptive personalized federated learning. arXiv. Preprint. Available from https://www.doi.org/10.48550/arXiv.2003.13461
45. He C, Annavaram M, Avestimehr S. Group knowledge transfer: Federated learning of large cnns at the edge. *Adv Neural Inf Process Syst.* 2020;33:14068-14080. https://www.doi.org/10.5555/3495724.3496904
46. Luopan Y, Han R, Zhang Q, Liu CH, Wang G, Chen LY. Fedknow: Federated continual learning with signature task knowledge integration at edge. *Proc IEEE 39th Int Conf Data Eng.* 2023;341-354. https://www.doi.org/10.1109/icde55515.2023.00033
47. Zuo X, Luopan Y, Han R, et al. FedViT: federated continual learning of vision transformer at edge. *Future Gener Comput Syst.* 2024;154:1-15. https://www.doi.org/10.1016/j.future.2023.11.038
48. Wang Q, Liu B, Li Y. Traceable federated continual learning. *Proc IEEE/CVF Conf Comput Vis Pattern Recognit.* 2024;12872-12881. https://www.doi.org/10.1109/cvpr52733.2024.01223
49. Wuerkaixi A, Cui S, Zhang J, et al. Accurate Forgetting for Heterogeneous Federated Continual Learning. arXiv. Preprint online Feb 20, 2025. https://www.doi.org/10.48550/arXiv.2502.14205
50. Sattler F, Müller KR, Samek W. Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints. *IEEE Trans Neural Netw Learn Syst.* 2020;32(8):3710-3722. https://www.doi.org/10.1109/tnnls.2020.3015958
51. Ghosh A, Chung J, Yin D, Ramchandran K. An Efficient Framework for Clustered Federated Learning. *IEEE Trans Inform Theory.* 2022;68(12):8076-8091. https://www.doi.org/10.1109/tit.2022.3192506
52. Tong G, Li GB, Wu J, Li J. Gradmfl: gradient memory-based federated learning for hierarchical knowledge transferring over non-iid data. *Proc Int Conf Algorithms Archit Parallel Process.* 2021;612-626. https://www.doi.org/10.1007/978-3-030-95384-3_38
53. Sun Z, Kairouz P, Suresh AT, McMahan HB. Can you really backdoor federated learning? arXiv. Preprint online November 18, 2019. https://www.doi.org/10.48550/arXiv.1911.07963
54. Sun J, Li A, DiValentin L, Hassanzadeh A, Chen Y, Li H. Fl-wbc: enhancing robustness against model poisoning attacks in federated learning from a client perspective. *Adv Neural Inf Process Syst.* 2021;34:12613-12624. https://www.doi.org/10.5555/3540261.3541226
55. Xie C, Koyejo O, Gupta I. SLSGD: secure and efficient distributed on-device machine learning. Paper presented at: joint European Conference on Machine Learning and Knowledge Discovery in Databases 2019; September 16-20, 2019; Würzburg, Germany. Accessed March 17, 2026. https://link.springer.com/chapter/10.1007/978-3-030-46147-8_13
56. Yin D, Chen Y, Kannan R, Bartlett P. Byzantine-robust distributed learning: towards optimal statistical rates. Paper presented at: Proceedings of the 35th International Conference on Machine Learning; July 10-15, 2018; Stockholmsmässan, Stockholm, Sweden. Accessed March 17, 2026. https://proceedings.mlr.press/v80/yin18a
57. Karimireddy SP, He L, Jaggi M. Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing. Paper presented at: International Conference on Learning Representations; April 25-29, 2022; Virtually.

Accessed March 17, 2026. https://openreview.net/forum?id=jXKKDEi5vJt

58. Han S, Buyukates B, Hu Z, et al. Fedsecurity: a benchmark for attacks and defenses in federated learning and federated llms. Paper presented at: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; August 25-29, 2024; Barcelona, Spain. Accessed March 17, 2026. https://www.doi.org/10.1145/3637528.3671545

Xiaoning Wu *a PhD student in Beijing Institute of Technology.*
https://orcid.org/0009-0001-8193-861X

Siqi Du *is a PhD student in Beijing Institute of Technology.*
https://orcid.org/0009-0000-9242-6241

Ke Qiu *is a Master student in Beijing Institute of Technology.*
https://orcid.org/0009-0007-5257-0434

Shilin Wen *is a Master student in Beijing Institute of Technology.*
https://orcid.org/0009-0007-5257-0434

Haiting Hou *a Master student in Beijing Institute of Technology.*
https://orcid.org/0009-0005-6971-9261

Yuxiao Liu *Include a short (maximum 150 words) biography of each author.*

Jianxin Zhao *is a PhD student in Beijing Institute of Technology.*
https://orcid.org/0000-0002-4147-9797

Chi Harold Liu *is currently a Full Professor and the Vice Dean with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing.Dr. Liu is a fellow of IET, and a fellow of Royal Society of the Arts.*
https://orcid.org/0000-0002-0252-329X

Rui Han *is a Professor in Beijing Institute of Technology.*
https://orcid.org/0000-0001-6894-1921

An International Journal of Optimization and Control: Theories & Applications
(https://accscience.com/journal/ijocta)