TrustedMix: Mixup with TEE for improving the data distribution heterogeneity of federated learning

Hongzhan Ma*†, Muyan Shen*†, Yu Qin*
*Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
†University of Chinese Academy of Sciences, Beijing 100049, China
Email: mahongzhan21@mails.ucas.ac.cn

***Abstract*—Data heterogeneity (Non-IID) remains a critical challenge in federated learning (FL), often leading to model divergence and performance degradation. We propose TrustedMix, a novel framework that leverages Trusted Execution Environments (TEEs) to facilitate secure, privacy-preserving data sharing for distribution rectification. In TrustedMix, clients offload encrypted subsets of local data to a server-side TEE enclave, which computes the global label distribution and performs cross-client sample mixing within its hardware-isolated boundary. These synthesized samples are redistributed to clients to augment local training, effectively mitigating the impact of data skewness. Evaluated on CIFAR-10 and Tiny-ImageNet benchmarks, TrustedMix consistently outperforms state-of-the-art methods including FedAvg, FedProx, and FedMix. Specifically, in extreme non-IID scenarios ($\alpha = 0.1$), the test accuracy of TrustedMix increases to up to 89.0%, significantly surpassing the performance of FedMix while maintaining robust data confidentiality and minimal communication overhead.**

***Index Terms*—Federated learning, Trusted Execution Environments, TEE-assistance, Privacy-Preserving**

I. Introduction

Distributed learning has become one of the most prominent challenges due to the emergence of edge and cloud computing. As a widely used distributed learning paradigm, federated learning (FL) enables clients to leverage their private data for collaborative model training while maintaining data privacy by avoiding direct access to individual client data [1], [2].

The most prevalent FL algorithm is FedAvg [2], which averages local models during each communication round and assigns weights based on the volume of client training data until convergence is achieved. However, since each client operates in a distinct environment, the collected data is often non-independent and identically distributed (Non-IID), a phenomenon referred to as data heterogeneity.

Previous studies have shown that neural network models are highly sensitive to variations in data distribution [3]. Consequently, the heterogeneity of client data significantly undermines the accuracy and convergence rate of FL [4]. Thus, mitigating data heterogeneity remains one of the major challenges facing the FL community.

Several variants of FedAvg aim to enhance the stability of the FL process to reduce the impact of data heterogeneity. For instance, FedProx [5] improves training stability by incorporating a proximal term into the client's loss function, while Legate et al. [6] proposed a re-weighted softmax method that reduces the contribution of labels with insufficient data. In addition to modifying the objective function, strategies such as enhancing the generalization of local models [7] and employing cross-training [8] have also been implemented to tackle this issue. However, these studies primarily focus on altering model update mechanisms without effectively rectifying the underlying data heterogeneity, which often leads to suboptimal training performance.

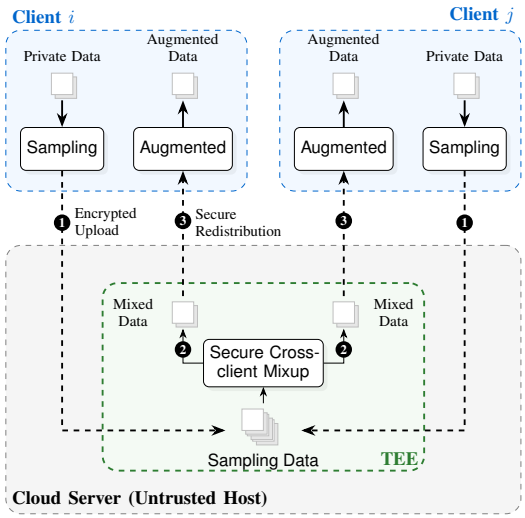

Fig. 1. Workflow of TrustedMix: (1) Secure sample transmission from clients to TEE; (2) Privacy-preserving data mixing within the TEE enclave; (3) Redistribution of augmented samples to enhance local training.

Recently, hardware-based Trusted Execution Environments (TEEs) have gained widespread adoption in machine learning [9]. TEEs are capable of securely storing data and executing authenticated code within the secure memory of untrusted devices. Several solutions leveraging TEEs have been proposed to safeguard FL against membership inference attacks (MIA) and data reconstruction attacks (DRA), while ensuring training integrity [10], [11]. However, limited research has focused on leveraging TEEs to directly mitigate client data heterogeneity while simultaneously preserving data privacy.

Inspired by the Mixup strategy [12], we propose **TrustedMix**, a novel framework that leverages TEEs to facilitate secure cross-client data augmentation to mitigate data heterogeneity. Specifically, each client shares a confidential portion of its local data with a server-side TEE, which subsequently performs secure data mixing and redistributes augmented samples back to the clients. Consequently, clients can utilize the received

samples to rectify local data skewness. Fig. 1 illustrates the architectural overview of our framework. TrustedMix enables clients to minimize data heterogeneity while maintaining strict privacy guarantees, representing a significant advancement over prior data augmentation solutions in FL [12], [13].

We evaluated TrustedMix using standard FL benchmarks, demonstrating its robustness across varying degrees of data heterogeneity. Experimental results indicate that TrustedMix achieves a superior convergence rate and effectively suppresses performance degradation in highly non-IID scenarios, thereby enhancing the overall utility of FL systems.

The main contributions of this work are summarized as follows:

- We propose TrustedMix, a TEE-based federated learning framework that enables secure cross-client data exchange. Our approach provides robust privacy protection while maintaining high computational efficiency even under constrained network conditions.
- We implement the TrustedMix framework and validate its effectiveness on multiple benchmark datasets. The results demonstrate that our method significantly outperforms existing baselines in scenarios characterized by extreme data heterogeneity.

The source code and experimental configurations of TrustedMix are publicly available at https://github.com/Sinope-Nanto/TrustedMix.

II. Related work and Background

This section establishes the technical foundation for TrustedMix by reviewing the hardware security primitives of TEEs and the existing federated learning (FL) paradigms for addressing data heterogeneity.

A. Trusted Execution Environment

A Trusted Execution Environment (TEE) provides a hardware-isolated secure area within the processor to ensure the confidentiality and integrity of sensitive data and code. AMD SEV-SNP [14] is one of the most prominent TEE implementations, facilitating the deployment of Secure Virtual Machines (SVMs) in untrusted cloud environments. SVMs utilize remote attestation to provide verifiable proof to a remote challenger that the workload is executing within a genuine TEE environment, while simultaneously generating measurements to verify the software stack's state during the boot process.

The memory of an SVM is hardware-encrypted, shielding its contents from unauthorized access by the hypervisor or cloud administrators. Beyond runtime protection, SEV-SNP offers data sealing mechanisms to safeguard the confidentiality of data in persistent storage. Furthermore, the Reverse Map Table (RMP) and associated metadata mechanisms enforce strict memory integrity, protecting SVMs against memory aliasing and tampering. Together, these hardware features ensure a robust root of trust for executing privacy-sensitive workloads like TrustedMix.

Algorithm 1 Generate Mixing Matrix $\mathbf{A}_k$

Require: Dimension of target samples n'_k
Ensure: Doubly stochastic mixing matrix $\mathbf{A}_k \in \mathbb{R}^{n'_k \times n'_k}$

Initialize: $a^k_{ij} \sim \text{Uniform}(0,1)$ for all $i,j \in \{1,\dots,n'_k\}$
for $j = 1$ **to** n'_k **do**
  $S_j \leftarrow \sum_{m=1}^{n'_k} a^k_{mj}$
  for $i = 1$ **to** n'_k **do**
    $a^k_{ij} \leftarrow a^k_{ij} / S_j$
  end for
end for
$\mathbf{\Delta} \leftarrow \mathbf{0} \in \mathbb{R}^{n'_k}$ {Error vector for column sums}
for $i = 1$ **to** $n'_k - 1$ **do**
  $R_i \leftarrow \sum_{m=1}^{n'_k} a^k_{im}$
  for $j = 1$ **to** n'_k **do**
    $\delta \leftarrow a^k_{ij} - (a^k_{ij} / R_i)$
    $\Delta_j \leftarrow \Delta_j + \delta$
    $a^k_{ij} \leftarrow a^k_{ij} / R_i$
  end for
end for
for $j = 1$ **to** n'_k **do**
  $a^k_{n'_k j} \leftarrow a^k_{n'_k j} + \Delta_j$
end for
return $\mathbf{A}_k$

B. Federated Learning

Federated Learning (FL) was pioneered by McMahan et al. [1], [2] to enable decentralized model training across multiple clients without aggregating their raw data. The most representative algorithm in this field is FedAvg [2], a synchronous approach that updates the global model by aggregating local model parameters after each communication round. To address the performance degradation caused by data and system heterogeneity, FedProx [5] was proposed, which introduces a proximal term to the local objective function to stabilize the training process.

Subsequent research has integrated various paradigms into FL to further mitigate the impact of non-IID data, including continual learning [6], transfer learning [15], knowledge distillation [8], Generative Adversarial Networks (GANs) [16], and traditional data augmentation [13]. Notably, TrustedMix is orthogonal to these algorithmic enhancements as it operates at the data-input level rather than modifying the optimization objective. By leveraging hardware-based security for data-level rectification, our solution can be seamlessly combined with these methods to achieve synergistic performance improvements.

III. Threat Model and Assumptions

We consider a standard federated learning (FL) framework where multiple clients perform local training and transmit model parameters to a central server for aggregation [17], [18], [19]. In this context, we identify two primary types of adversaries: (i) malicious clients participating in the training process, and (ii) the untrusted server host (e.g., a cloud service provider).

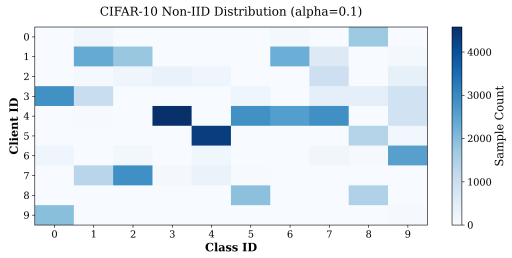

Fig. 2. Visualization of label distribution across 10 clients for CIFAR-10 ($\alpha = 0.1$). The heatmap illustrates the sample counts per class for each client, where the color intensity reflects the degree of data skewness in the simulated Non-IID environment.

We assume these adversaries are *honest-but-curious*; specifically, they strictly follow the FL protocol and TEE-based operations but attempt to infer or access the private data of other clients through the information they receive. Attacks targeting the hardware implementation of TEEs, such as side-channel attacks [20], physical hardware tampering [21], and denial-of-service (DoS) attacks, are considered beyond the scope of this paper.

To ensure secure execution, we assume that a root of trust is established between the TEE and each client via remote attestation. Based on this trust, a secure communication channel is created (e.g., utilizing the Diffie-Hellman (DH) key exchange protocol [22]) to ensure data confidentiality during transit.

Furthermore, we assume that the untrusted host does not obstruct the normal operation of the TEE enclave. Should the host refuse to provide necessary resources or terminate the enclave, it would impact the system's availability rather than its confidentiality or integrity. Consistent with standard TEE security research, such DoS-style availability attacks are not addressed in this study.

IV. The TrustedMix Framework

In this section, we detail the design of TrustedMix, a hardware-software co-design framework for mitigating data heterogeneity, and provide a analysis of the system's privacy and security guarantees.

A. System Overview

The TrustedMix framework facilitates secure data-level rectification through four primary stages: local data sampling, secure sample mixing, data augmentation, and local training with global aggregation. The overall workflow is illustrated in Fig. 1.

At the commencement of each communication round, each client randomly selects a subset of its private dataset. These samples are encrypted and transmitted to the server-side TEE via a secure channel. Upon receiving the encrypted subsets, the TEE decrypts them within its hardware-isolated enclave. It then partitions the aggregate shared dataset based on each client's specific distribution needs and performs a secure cross-client Mixup operation. The resulting mixed samples are then re-encrypted and redistributed to the respective clients.

After receiving the mixed samples, each client performs data augmentation according to the augmentation ratio specified by the server. These augmented samples, combined with the original local dataset, serve as the input for training the local model. Once the local training is completed, the server collects the updated model parameters from all participating clients and performs global aggregation (e.g., via FedAvg) to update the global model.

B. Data Sampling and Secure Mixup

Before the training process commences, each client $k \in \{1, \ldots, K\}$ uploads its local label distribution vector $\boldsymbol{\rho}_k = (n_k^1, \ldots, n_k^c)$ to the server-side TEE, where n_k^i denotes the number of samples associated with label i at client k. To rectify the data skewness, the server calculates a target distribution vector $\boldsymbol{\rho}'_k = (n_k^{1\prime}, \ldots, n_k^{c\prime})$ within the enclave, such that:

$$\begin{aligned} m &= \arg\max_i \frac{n_k^i}{n^i}, \\ n_k^{i\prime} &= \left\lfloor \frac{n_k^m n^i}{n^m} - n_k^i \right\rfloor, \end{aligned} \tag{1}$$

where n^i represents the total number of samples with label i across all participating clients.

In each communication round, client k randomly selects $\lfloor \gamma n_k \rfloor$ samples and transmits them to the TEE, where γ is a predetermined sampling ratio. Upon receiving these subsets, the TEE generates a shared dataset $\mathbf{M}_k$ tailored to the distribution $\boldsymbol{\rho}'_k$ for each client k. Each column vector in $\mathbf{M}_k$ represents an individual shared sample.

Subsequently, the TEE generates a mixing matrix $\mathbf{A}_k = \{a_{ij}^k\}_{n'_k \times n'_k}$ according to Algorithm 1, and computes the augmented dataset $\mathbf{M}'_k = \mathbf{M}_k \mathbf{A}_k$, which is then returned to client k. As demonstrated in Algorithm 1, each row and column of $\mathbf{A}_k$ sums to unity. This ensures that every sample in $\mathbf{M}_k$ is incorporated with consistent weighting. Furthermore, for any synthesized sample in $\mathbf{M}'_k$, the components of its label vector remain on the probability simplex (i.e., they sum to 1). This mathematical property guarantees that $\mathbf{M}'_k$ preserves the target label distribution of $\mathbf{M}_k$, thereby effectively mitigating the data heterogeneity across clients.

C. Local Training and Aggregation

In general, upon receiving the mixed samples $\mathbf{M}'_k$ from the server, client k incorporates these samples with its local dataset to form the training set for the current round. However, if the local dataset size n_k is substantial, or if the local label distribution deviates significantly from the global distribution, the required number of supplementary samples n'_k may become excessively large, leading to high communication overhead.

To optimize communication efficiency in such scenarios, the server transmits an augmentation ratio λ_k to client k and scales the transmitted samples according to $\boldsymbol{\rho}'_k/\lambda_k$. After receiving the downsampled $\mathbf{M}'_k$, client k first restores the required data volume by performing local data augmentation based on λ_k. This augmentation process is compatible with various established schemes, such as geometric transformations

TABLE I
TEST ACCURACY ON CIFAR10 UNDER VARIOUS HETEROGENEOUS ENVIRONMENTS.

α	FedAvg	FedMix	FedProx	TrustedMix
0.1	84.1	88.2	85.2	**89.0**
0.3	90.2	91.5	90.0	**91.8**
0.5	89.5	91.7	90.5	**92.0**
iid	92.7	**93.0**	91.5	92.7

TABLE II
TEST ACCURACY ON CIFAR10 UNDER VARIOUS LOCAL EPOCHS(E). $\alpha = 0.1$.

Local Epochs$E$	FedAvg	FedMix	TrustedMix
1	81.1	83.7	**84.4**
5	85.4	89.1	**90.5**
10	84.1	88.2	**89.0**
20	83.5	86.0	**87.5**

[23], noise injection [24], or color space transformations [25]. The resulting augmented dataset is then utilized alongside local data for model training.

Upon completing the local training phase, client k transmits its local model parameters Θ_k back to the server. The server then performs global aggregation to update the global model parameters $\Theta = \sum_{k=1}^{K} \frac{n_k}{n} \Theta_k$, where $n = \sum n_k$ represents the total number of samples across all clients. The updated global parameters Θ are subsequently redistributed to all clients to initiate the next communication round.

D. Security Analysis

In the TrustedMix framework, the privacy of client data is guaranteed through hardware-isolated processing and cryptographic mixing. During the training process, the raw private data of each client is processed exclusively within the server-side TEE enclave. Participating clients only receive the synthesized samples, which are linear combinations of multiple source images. Critically, since the mixing matrix $\mathbf{A}_k$ remains strictly within the TEE, it is computationally infeasible for a malicious client to solve the inverse problem and reconstruct the original private samples of others.

To address the computational overhead of generating $\mathbf{A}_k$ and performing matrix multiplication when the number of offloaded samples n'_k is large, we implement a block-based mixing strategy. Specifically, the sample set $\mathbf{M}_k$ is partitioned into blocks, and the mixing operation is applied independently within each block of size $B_M \times B_M$.

Our empirical analysis indicates that the block size B_M significantly influences the balance between privacy and computational overhead. We observe that a smaller block size, such as $B_M = 8$, is sufficient to effectively obscure the structural features and semantic information of the original images. By restricting the mixing to these localized blocks, the complex global pixel correlations are broken down into fragmented combinations, making it impossible to identify or recover individual source samples. Consequently, we adopt $B_M = 8$ as the default parameter for our framework, as it ensures robust privacy protection while maintaining high computational efficiency.

Furthermore, the data distribution of each client is shared exclusively with the TEE via a secure channel. The TEE computes the global distribution internally and redistributes shared data accordingly. While a client may infer the global label distribution from the received samples, it cannot access the specific distribution or metadata of any other individual client. Thus, TrustedMix provides robust protection for both the raw private data and the sensitive label distribution of all participants.

V. EXPERIMENTS

In this section, we conduct a comprehensive experimental evaluation of TrustedMix to verify its effectiveness in mitigating data heterogeneity. We describe the experimental setup, evaluate the performance against multiple state-of-the-art baselines on representative benchmarks, and provide a detailed analysis of the system's robustness and operational overhead.

A. Experimental Settings

a) Datasets and Architecture: We evaluate TrustedMix on two benchmarks: CIFAR-10 [26] and TinyImageNet [27]. To simulate realistic heterogeneous environments, we employ a Dirichlet distribution Dir(α) to partition the data among $K = 10$ clients. As illustrated in Fig. 2, we set the concentration parameter $\alpha = 0.1$ to create a highly skewed and non-IID label distribution, which serves as a challenging baseline for evaluating robustness to data heterogeneity.

Regarding the model architectures, we utilize two representative deep networks to verify the scalability of TrustedMix. For the CIFAR-10 experiments, we employ a VGG16 architecture [28], which consists of thirteen 3×3 convolutional layers followed by three fully connected layers, providing a baseline for high-parameter density. For both CIFAR-10 and Tiny-ImageNet, we further adopt the ResNet50 [29] architecture. The use of residual learning and a significantly deeper 50-layer structure allows us to evaluate TrustedMix's performance on complex feature extraction tasks and its communication efficiency in million-parameter deployments.

b) Implementation Details: Experiments are conducted with a full client participation fraction $C = 1.0$ and $E = 10$ local epochs per round. We use SGD with a learning rate $\eta = 0.01$. Training spans $R = 100$ rounds and batchsize is $B = 128$. The algorithm-specific hyper-parameters are configured to match optimal baseline performances: for FedProx, the proximal term is $\mu = 0.1$ (0.01 for TinyImageNet); for FedMix, the mixing ratio λ is set to 0.05, and 0.1 for CIFAR-10 and TinyImageNet, respectively. For our TrustedMix, the sampling ratio is consistently fixed at $\gamma = 0.1$ across all datasets. For Tiny-ImageNet with ResNet50, we apply random cropping and horizontal flipping as local augmentation to match the offloaded sample volume. To ensure the reliability of the reported accuracy, all experiments are conducted with

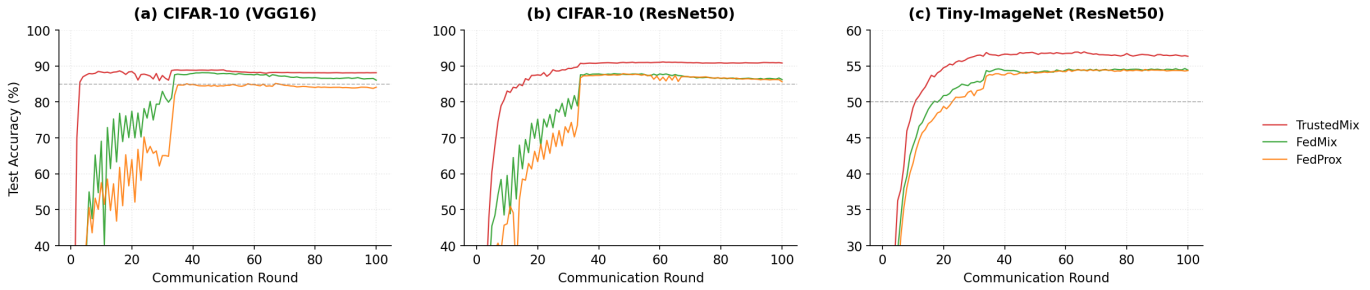

Fig. 3. Learning curves of TrustedMix and baseline methods across different model architectures and datasets. (a) CIFAR-10 (VGG16) (b) CIFAR-10 (ResNet50) (c) Tiny-ImageNet (ResNet50). The horizontal dashed lines indicate benchmark accuracy levels, highlighting the superior convergence efficiency of TrustedMix under extreme non-IID conditions ($\alpha = 0.1$).

randomized client partitioning and data shuffling to simulate diverse non-IID conditions.

B. Experimental Results and Analysis

We compare TrustedMix against three representative baselines and an ideal scenario with centralized global data: 1) FedAvg [2]: The standard federated aggregation baseline; 2) FedProx [5]: A state-of-the-art method addressing system and data heterogeneity; 3) FedMix [13]: A data-augmentation-based FL approach.

We evaluate the learning performance of TrustedMix on CIFAR-10 and TinyImageNet using VGG16 and ResNet50 architectures, respectively. As illustrated in Fig. 3, TrustedMix consistently outperforms all baselines in terms of both convergence rate and final accuracy. Notably, even under extreme label skewness ($\alpha = 0.1$), our framework achieves a rapid ascent in accuracy during the initial 20 rounds.

a) Robustness to Data Heterogeneity: We investigate the impact of varying degrees of data distribution skew on TrustedMix using the CIFAR-10 dataset with the VGG16 architecture, keeping other parameters consistent with the main experiments. By fixing these parameters and varying the concentration parameter α of the Dirichlet distribution, we simulate different Non-IID levels. Additionally, we include an IID training setup as a reference to evaluate the baseline performance.

As shown in Table I, TrustedMix maintains a superior performance margin over all existing baselines across all settings. Notably, the performance gain of TrustedMix is most pronounced in high-heterogeneity scenarios (i.e., smaller α values).

In the IID case, the performance differences among various models are marginal and primarily attributed to the inherent stochasticity during training. Furthermore, when the local data distribution of a client aligns with the global label distribution, the target rectification vector $\boldsymbol{\rho}'_k$ computed within the TEE naturally approaches zero. In such scenarios, TrustedMix remains idle and bypasses the mixing operation for these clients, thereby effectively degenerating into the standard FedAvg protocol to avoid unnecessary data perturbation. This observation confirms that the performance leap in Non-IID environments originates from TrustedMix's unique ability to effectively mitigate data distribution skew.

b) Impact of Local Training Intensity: We further study the impact of local training epochs (E) on the stability of TrustedMix. To accentuate the challenges posed by data heterogeneity, we conduct these experiments under a high-skew setting with $\alpha = 0.1$, a more stringent condition than the main experiments. In traditional FL, increasing E can accelerate convergence but often leads to local model drifting and overfitting when E is excessively large, particularly under such severe Non-IID settings.

According to the results in Table II, TrustedMix outperforms the baselines across all configurations of E. Interestingly, while baselines exhibit performance degradation at higher E values due to the intensified distribution skew, TrustedMix remains relatively stable. This stability is attributed to the fact that TrustedMix realigns the local training data with the global distribution within the TEE, thereby preventing local models from drifting toward biased local optima even under high heterogeneity.

C. Overhead Analysis

TrustedMix introduces marginal communication and temporal costs. To maintain bandwidth efficiency, we cap offloaded samples at 10% of the local dataset, subsequently restoring the required volume via local data augmentation (Section IV-C). Quantitative analysis demonstrates that the relative communication overhead is inversely proportional to the model scale. Specifically, while the overhead reaches 3.98% for the VGG16 architecture on CIFAR-10, it drops significantly to 0.58% for the ResNet50 model on Tiny-ImageNet. This trend highlights that in modern deployments featuring million-parameter architectures, the additional data payload becomes negligible. Crucially, since the TEE only processes a small fraction of the data, the bandwidth required for encrypted image offloading is orders of magnitude smaller than the standard gradient or weight exchange performed in each FL round.

From a temporal perspective, the encrypted exchange and TEE-based mixing operate in a pipelined fashion, effectively overlapping with local training computations. Given that deep model optimization remains the dominant computational bottleneck, TrustedMix avoids becoming a latency bottleneck and preserves the overall system throughput.

VI. Conclusion

This paper presented TrustedMix, a novel federated learning framework that leverages Trusted Execution Environments (TEEs) and the Mixup strategy to mitigate data heterogeneity while strictly preserving privacy. By facilitating secure, cross-client data mixing within hardware-isolated enclaves, TrustedMix effectively rectifies local data skewness. Experimental evaluations across multiple benchmarks demonstrate that TrustedMix significantly outperforms existing baselines, particularly in scenarios with extreme data heterogeneity, with marginal communication and temporal overhead.

Future work will explore the extensibility of TrustedMix to broader domains, such as detecting data poisoning attacks [30], [31] and addressing feature-level distribution shifts [18], [32]. Additionally, we aim to adapt the TrustedMix paradigm to more complex modalities, including multimodal learning and large-scale natural language processing, to further enhance the robustness and utility of privacy-preserving federated learning.

Acknowledgment

The Paper is supported by National Key R&D Program of China (2024YFE0211100).

References

[1] J. Konečnỳ, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

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

[3] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?" in *International conference on machine learning*. PMLR, 2019, pp. 5389–5400.

[4] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification. arxiv 2019," *arXiv preprint arXiv:1909.06335*, 2019.

[5] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[6] G. Legate, L. Caccia, and E. Belilovsky, "Re-weighted softmax cross-entropy to control forgetting in federated learning," in *Conference on Lifelong Learning Agents*. PMLR, 2023, pp. 764–780.

[7] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8397–8406.

[8] T. Liu, Z. Qi, Z. Chen, X. Meng, and L. Meng, "Cross-training with prototypical distillation for improving the generalization of federated learning," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2023, pp. 648–653.

[9] Z. Sun, R. Sun, C. Liu, A. R. Chowdhury, L. Lu, and S. Jha, "Shadownet: A secure and efficient on-device model inference system for convolutional neural networks," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1596–1612.

[10] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "Ppfl: Privacy-preserving federated learning with trusted execution environments," in *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, 2021, pp. 94–108.

[11] D. Feng, Y. Qin, W. Feng, W. Li, K. Shang, and H. Ma, "Survey of research on confidential computing," *IET Communications*, vol. 18, no. 9, pp. 535–556, 2024.

[12] H. Zhang, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[13] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, "Fedmix: Approximation of mixup under mean augmented federated learning," *arXiv preprint arXiv:2107.00233*, 2021.

[14] A. Sev-Snp, "Strengthening vm isolation with integrity protection and more," *White Paper, January*, vol. 53, pp. 1450–1465, 2020.

[15] G. Legate, N. Bernier, L. Page-Caccia, E. Oyallon, and E. Belilovsky, "Guiding the last layer in federated learning with pre-trained models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[16] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *arXiv preprint arXiv:1811.11479*, 2018.

[17] K. Bonawitz, "Towards federated learning at scale: Syste m design," *arXiv preprint arXiv:1902.01046*, 2019.

[18] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–44, 2023.

[19] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, and Q. Yang, "Federated learning for generalization, robustness, fairness: A survey and benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[20] M. Li, Y. Zhang, H. Wang, K. Li, and Y. Cheng, "{CIPHERLEAKS}: Breaking constant-time cryptography on {AMD}{SEV} via the ciphertext side channel," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 717–732.

[21] M. Lipp, A. Kogler, D. Oswald, M. Schwarz, C. Easdon, C. Canella, and D. Gruss, "Platypus: Software-based power side-channel attacks on x86," in *2021 IEEE Symposium on Security and Privacy (SP)*, May 2021. [Online]. Available: http://dx.doi.org/10.1109/sp40001.2021.00063

[22] F. Bao, R. H. Deng, and H. Zhu, "Variations of diffie-hellman problem," in *International conference on information and communications security*. Springer, 2003, pp. 301–312.

[23] E. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. Le, "Autoaugment: Learning augmentation policies from data," *arXiv: Computer Vision and Pattern Recognition*, May 2018.

[24] F. J. Moreno-Barea, F. Strazzera, J. M. Jerez, D. Urda, and L. Franco, "Forward noise adjustment scheme for data augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2018. [Online]. Available: http://dx.doi.org/10.1109/ssci.2018.8628917

[25] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proceedings of the British Machine Vision Conference 2014*, Jan 2014. [Online]. Available: http://dx.doi.org/10.5244/c.28.6

[26] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[27] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations,International Conference on Learning Representations*, Jan 2015.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[30] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[31] X. Lyu, Y. Han, W. Wang, J. Liu, B. Wang, J. Liu, and X. Zhang, "Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 9020–9028.

[32] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 2022, pp. 965–978.