SECUREHDC-FL: ADDRESSING DATA HETEROGENEITY IN ENCRYPTED FEDERATED HYPERDIMENSIONAL COMPUTING

Xiangyu Hu[1,2,3], *Jingwei Chen*[1,2,3], *Wenyuan Wu*[1,2,3], *Yong Feng*[1,2,3]

[1] Chongqing Institute of Green and Intelligent Technology, CAS
[2] Chongqing School, University of Chinese Academy of Sciences
[3] Chongqing Key Laboratory of Secure Computing for Biology
{huxiangyu, chenjingwei, wuwenyuan, yongfeng}@cigit.ac.cn

ABSTRACT

Federated learning in privacy-sensitive scenarios such as edge computing continues to face challenges from privacy leakage and performance degradation under non-IID data distributions. While homomorphic encryption(HE)-based federated hyperdimensional computing (HDC) has shown promise for privacy-preserving collaboration, prior studies mainly adopt simple aggregation schemes(e.g., uniform averaging). These operations are convenient in the ciphertext domain but fail to capture data heterogeneity, leading to poor performance under non-IID(non-independent and identically distributed) distributions. To address this limitation, we propose SecureHDC-FL, a secure federated HDC protocol that incorporates an HE-based dynamic weighted aggregation mechanism to enhance convergence and robustness while preserving privacy. Despite the computational complexity under encryption, our optimized design ensures practical efficiency. Both theoretical analysis and experimental results demonstrate that SecureHDC-FL achieves faster convergence (reducing training rounds by up to **50%**) and higher accuracy (up to **2.5%** improvement) compared with SOTA methods in non-IID settings, while incurring only acceptable overhead.

Index Terms— Hyperdimensional Computing, Federated Learning, Homomorphic Encryption, Privacy-Preserving Machine Learning, Edge Computing

1. INTRODUCTION

Hyperdimensional Computing (HDC) is a lightweight learning paradigm suitable for edge devices [1]. When combined with Federated Learning (FL) [2], it enables distributed training while preserving data privacy. However, model parameters may still leak information [3, 4, 5], prompting researchers to introduce Homomorphic Encryption (HE) [6, 7] for secure aggregation, which addresses privacy concerns at the cost of computational efficiency, alongside the persistent challenge of data heterogeneity.

Existing studies have separately explored the integration of FL with HDC (e.g., FedHD [8], FHDnn [9]) and the integration of HE with HDC (e.g., FHE-HD [10]). Rhychee-FL [11] was the first to unify all three, achieving privacy protection through encrypted HDC model aggregation. This line of work, however, still relies on simple averaging strategies leads to limited performance in non-IID environments [12], and it does not fully exploit the computational potential of fully homomorphic encryption. Table 1 summarizes the comparison between SecureHDC-FL and prior approaches.

To address these issues, we propose SecureHDC-FL, a secure federated HDC framework. Our core contributions are:

(1) We design an HE-friendly HDC-FL aggregation strategy that comprehensively considers data volume, model similarity, and historical information. Both theoretical analysis and experimental results demonstrate that this strategy achieves tighter convergence bounds and significantly accelerates convergence in non-IID scenarios.

(2) To enable efficient dynamic aggregation under CKKS scheme [7], we realize efficient ciphertext domain Softmax via client-server collaborative computation and polynomial approximation, and conduct in-depth optimizations of high-overhead encrypted matrix multiplication, providing flexible trade-offs between computational and communication costs across different network environments.

(3) SecureHDC-FL achieves up to **2**× faster convergence and up to **2.5%** higher accuracy compared to SOTA method in non-IID scenarios, while maintaining manageable overhead.

Table 1: Feature comparison of FL methods.

Method	**Edge Efficiency**	**HE Protection**	**Dynamic Aggregation**	**Non-IID Robustness**
DNN+FL [2, 13, 14]	○	○	◐	○
HDC+FL [8, 9]	●	○	○	◐
HE+FL [15, 16]	○	●	◐	○
Rhychee-FL [11]	●	●	○	○
SecureHDC-FL	●	●	●	●

● Supported; ○ Not supported; ◐ Partially supported.

J. Chen is the corresponding author.

This work was supported partly by the National Key R&D Program of China (2025YFA1017200), NSFC (12571553) and Natural Science Foundation of Chongqing (cstb2023yszx-jcx0008).

2. PROPOSED METHOD

In our protocol SecureHDC-FL, for a K-class classification problem, M clients locally train HDC models using their own sensitive data (where the i-th model ($i < M$) consists of K d-dimensional class hypervectors $\{L_{i,j}^{(t)} \in \mathbb{R}^d : j < K\}$), and then upload the results to the server after encrypting them with CKKS scheme [7]. The server performs dynamic weighted aggregation in the ciphertext domain to generate and return the global model $G_j^{(t)}{}_{j<K}$. , which is iterated over $t = 0, 1, \ldots$ until convergence. We assume that all participants are honest-but-curious, and the security is based on multi-key homomorphic encryption [17, 18].

2.1. Dynamic Weighted Aggregation

To address the challenge of non-IID data in federated HDC, we design a dynamic weighted aggregation mechanism. While Rhychee-FL [11] adopts simple strategies, SecureHDC-FL (our method) computes a composite weight for each client's local model, consisting of two key components: (1) *Model similarity weight*: compute the local-global cosine similarity for each class $S_{i,j}^{(t)} = \cos(L_{i,j}^{(t)}, G_j^{(t)})$ and normalize it across classes using Softmax; (2) *Data volume weight*: proportional to the number of local samples of the client.

These two weights are balanced by a hyperparameter α: $W = \alpha \cdot W_{\text{data}} + (1-\alpha) \cdot W_{\text{similarity}}$. The server then uses this weight to aggregate the encrypted local models: $G_{j,\text{agg}}^{(t+1)} \leftarrow \sum_{i=0}^{M-1} w_{i,j}^{(t)} \cdot L_{i,j}^{(t)}$. To enhance convergence stability, we introduce the exponential moving average (EMA) to update the global model: $G^{(t+1)} = \beta \cdot G_{\text{agg}}^{(t+1)} + (1-\beta) \cdot G^{(t)}$, where β is a hyperparameter controlling the update strength. The entire aggregation process mainly involves basic operations such as inner products and vector multiplications, which are highly compatible with the CKKS homomorphic encryption scheme. A small number of nonlinear operations can be approximated efficiently, thereby enabling efficient and secure execution in the ciphertext domain.

To theoretically support the effectiveness of this dynamic weighted aggregation strategy, we analyze its convergence. Under standard assumptions in federated learning [19] (e.g., L-smoothness and μ-strong convexity of the loss function), we prove that the proposed algorithm achieves an $\mathcal{O}(\frac{1}{T})$ convergence rate, where T denotes the number of communication rounds. More importantly, our analysis reveals the existence of an optimal hyperparameter α^* (In practice, α is treated as a hyperparameter and selected via grid search.), which precisely balances the variance introduced by data heterogeneity and the bias caused by model heterogeneity, thereby minimizing the convergence upper bound. The optimal solution takes the form: $\alpha^* = \frac{\beta(V_s - V_{ps}) + 3L(2-\beta)(A_s - A_p)}{\beta(V_p - 2V_{ps} + V_s)}$, where the terms V_s, V_{ps}, V_p and A_s, A_p quantify the variance and bias under different weighting strategies, respectively. This result theoretically explains why our dynamic method achieves a tighter convergence bound than fixed-weight strategies, and thus converges faster under non-IID settings. The explanations of the notation, detailed theorem statements and proofs are provided in the appendix. [1]

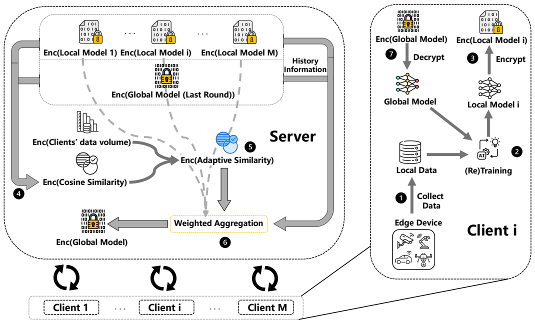

Fig. 1: System Architecture of SecureHDC-FL.

2.2. Efficient and Secure Implementation

To efficiently achieve aggregation in the ciphertext domain, we optimize two key components: the computation of Softmax normalization in the ciphertext domain, and the costly ciphertext matrix–vector multiplication.

Encrypted Similarity Weight Computation: To efficiently compute Softmax weights in the ciphertext domain, we need to address operations such as $\exp(\cdot)$, cosine similarity, and $\frac{1}{x}$, which are generally unsuitable for homomorphic encryption (HE). Exploiting the fact that each client has access to both its local model and the previous global model, we design a strategy that offloads computation between clients and the server: cosine similarity and its exponential operation $\exp(\cos(\cdot))$ are shifted to the client side and performed in the plaintext domain. As a result, the server only needs to handle the $\frac{1}{x}$ required for normalization in the ciphertext domain. We approximate $\frac{1}{x}$ using a second-order Chebyshev polynomial. Owing to the construction of the similarity measure, the input domain is restricted to an interval $[\frac{M}{e}, Me]$ related to the number of clients M, where this low-degree approximation suffices to achieve the desired precision. Theoretical analysis shows that the error bound is $\mathcal{O}(\frac{1}{M})$(derivation provided in the appendix), meaning the approximation accuracy improves as more participants are involved, thereby ensuring scalability and stability in large deployments.

Efficient Encrypted Matrix–Vector Multiplication: The most expensive step in the protocol is matrix–vector multiplication for each class j (i.e., encrypted model aggregation): $G_j^{(t+1)} = C_j^{(t)} \cdot W_j$, where $C_j^{(t)} \in \mathbb{R}^{d \times M}, d \gg M$. A straightforward, row-by-row implementation requires d independent ciphertext inner products, incurring prohibitive cost. To overcome this, we design two optimized schemes tailored for low-bandwidth and high-bandwidth environments.

[1] https://github.com/icassp20262336/CASSP2026-2336-Supplement

Table 2: Computational complexity and communication cost.

		Low Bandwidth	**High Bandwidth**
Computational complexity	#Mul	$O(KM)$	$O(KM)$
	#CMul	$O(KM)$	$O(KM)$
	#Rot	$O(KM \log d)$	$O(K \log \frac{d}{M} + KM)$
Communication cost (bits)	Upload	$\lceil \frac{dK}{\ell} \rceil \cdot 2N \log Q$	$M \cdot \lceil \frac{dK}{\ell} \rceil \cdot 2N \log Q$
	Download	$\lceil \frac{dK}{\ell} \rceil \cdot 2N \log Q'$	$M \cdot \lceil \frac{dK}{\ell} \rceil \cdot 2N \log Q'$

ℓ: CKKS slot number; N: ring dimension; Q, Q': ciphertext modulus.

Low-Bandwidth Mode: When minimizing communication overhead is critical, we adopt a column-wise strategy: clients tightly pack model parameters by columns before uploading. In this setting, the server reconstructs the matrix–vector multiplication as a linear combination of column vectors: $G_j^{(t+1)} = \sum_{i=0}^{M-1} w_{i,j}^{(t)} \cdot L_{i,j}^{(t)}$, where $L_{i,j}^{(t)}$ denotes the encrypted column vector uploaded by client i. This requires M independent ciphertext vector–scalar multiplications followed by ciphertext vector additions, whose efficiency has been demonstrated in [20, 21].

High-Bandwidth Mode: When latency reduction takes priority over communication efficiency, we apply the diagonal encoding technique [22], effectively trading communication for speed. Clients pre-encode their data using $F_{i,j,k}^{(t)} = L_{i,j}^{(t)} \circ g_{i-k \bmod M}$($g$ is mask vector). The server aggregates the pre-encoded vectors to obtain $d_k = \sum_i F_{i,j,k}^{(t)}$, then computes: $G_j^{(t+1)} = \sum_{k=0}^{M-1} d_k \odot \mathrm{Rot}_k(W_j)$.

Table 2 summarizes the per-round ciphertext computation and communication costs of the two modes. The high-bandwidth mode reduces the computational bottleneck—rotation operations (#Rot)—from $O(KM \log d)$ to $O(K \log \frac{d}{M} + KM)$, at the expense of M times higher communication overhead. This communication–computation trade-off provides deployment flexibility under varying network conditions.

The complete protocol flow is presented in Protocol 1. Its security relies on the standard assumptions of multi-key FHE [17, 18]. Under the honest-but-curious security model, the protocol can resist attacks from the server, external adversaries, and collusion between the server and a subset of clients, while achieving powerful secure dynamic aggregation with reasonable overhead.

3. EXPERIMENTS

3.1. Set Up

We evaluate SecureHDC-FL on four real-world datasets from different domains: MNIST, UCI-HAR, ISOLET, and CARDIO, and compare it against two baselines: a recent HE-based federated learning scheme, OECM [23],and the current SOTA HE-based scheme Rhychee-FL [11]. To simulate realistic scenarios, we construct two types of non-IID environments:

Protocol 1 (SecureHDC-FL)

```
Public parameters: Client count $M$, rounds $T$, hyperparameters $\alpha, \beta$.
Initialization: Server distributes initial global model $G^{(0)}$.
for $t = 0, 1, \ldots, T-1$ do
    Client $i$ (in parallel):
        Train local model $L_i^{(t)}$ using local data and $G^{(t)}$.
        Compute similarity scores for each class $j$:
        $S_{i,j}^{(t)} \leftarrow \exp(\cos(L_{i,j}^{(t)}, G_j^{(t)}))$.
        Encrypt and upload $\{L_i^{(t)}, (S_{i,j}^{(t)})_j, n_i^{(t)}\}$ to the server.
    Server (operating in the ciphertext domain):
        Compute similarity weights $S^{(t)}$ from all $\{S_{i,j}^{(t)}\}_j$ via
        polynomial approximation for division.
        Compute data quantity weights $N_{\text{norm}}^{(t)}$ from all $\{n_i^{(t)}\}$.
        Determine final aggregation weights:
        $W^{(t)} \leftarrow \alpha \cdot N_{\text{norm}}^{(t)} + (1-\alpha) \cdot S^{(t)}$.
        Securely aggregate local models for each class
        $j$(using bandwidth-specific efficient methods):
        $G_{j,\text{agg}}^{(t+1)} \leftarrow \sum_{i=0}^{M-1} w_{i,j}^{(t)} \cdot L_{i,j}^{(t)}$ .
        $G^{(t+1)} \leftarrow \beta \cdot G_{\text{agg}}^{(t+1)} + (1-\beta) \cdot G^{(t)}$.
        Broadcast the encrypted global model $G^{(t+1)}$ .
Output: The final decrypted global model $G^{(T)}$.
```

(1) **Dual Non-IID**, where label skew and data quantity skew are controlled respectively by a Dirichlet parameter γ and the standard deviation σ of a log-normal distribution [12, 24]; (2) **Additive Gaussian Noise**, as described in [12], but with extensions. Each client samples a noise mean $\mu_c \sim \mathcal{N}(0, \sigma)$, and then draws noise from $\mathcal{N}(\mu_c, \text{scale})$ that is added to its local data. Our protocol is implemented based on the CKKS scheme in Microsoft SEAL, with a ring dimension $N = 2^{14}$ and the ciphertext modulus q with $\log q \approx 320$ to reach a 128-bit security level. Preliminary experiments (Table 3) show that under the IID setting, our aggregation strategy achieves performance comparable to Rhychee-FL without sacrificing accuracy.

Table 3: Final Accuracy / Rounds to 90% under IID setting.

Method	MNIST	ISOLET	UCIHAR	CARDIO
SecureHDC-FL	0.967 / 2	0.943/5	0.947 / 3	0.939/7
Rhychee-FL	0.966 / 2	0.937/5	0.945 / 3	0.940/7

3.2. Experimental Results

Impact of Hyperparameters. Under medium-level non-IID settings ($\gamma = 0.5, \sigma = 0.5$), we explore the impact of the hypervector dimension d and the number of clients (Figure 2). Results show that the accuracy of SecureHDC-FL is insensitive to the number of clients, exhibiting good scalability. On MNIST, accuracy already exceeds 95% when $d = 1000$. For UCI-HAR, ISOLET, and CARDIO, performance peaks around $93\% - 95\%$. Across all datasets, dimension $d \geq 4000$ consistently yields near-optimal and stable performance.

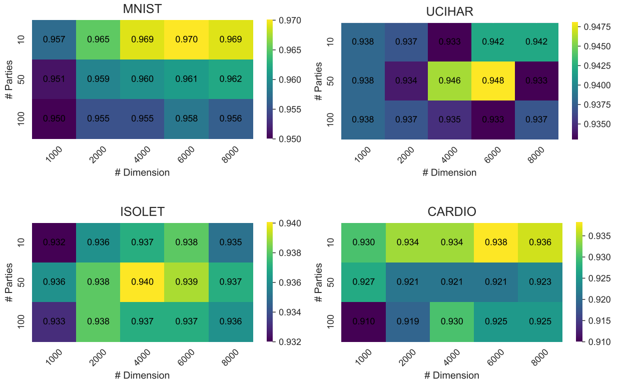

Fig. 2: Accuracy under Different Dimensions and Clients.

Data Imbalance. Under highly imbalanced non-IID settings ($\gamma = 0.1, \sigma = 0.9$), SecureHDC-FL demonstrates outstanding convergence efficiency across all four datasets (Figure 3). Across client counts ranging from 10 to 100, SecureHDC-FL consistently requires the fewest rounds to reach 90% accuracy; e.g., on MNIST, it converges nearly **2**× faster than Rhychee-FL and **3.5**× faster than OECM. In contrast to the baseline methods, which exhibit significantly slower convergence, SecureHDC-FL maintains its rapid and stable performance regardless of the number of clients.

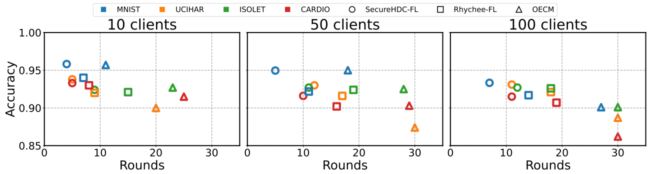

Fig. 3: Global Accuracy vs. Rounds (Data Imbalance)

Robustness to Feature Heterogeneity. To evaluate robustness against feature heterogeneity, we introduced non-IID additive Gaussian noise ($\sigma = 0.5$, scale $= 0.5$) across all four datasets. As shown in Figure 4, SecureHDC-FL consistently achieves faster convergence and higher final accuracy in these noisy conditions(e.g., under the 100-clients setting, it reaches 90% accuracy nearly **1.5**× faster than Rhychee-FL, while also achieving up to **2.5**% higher final accuracy). This result highlights its superior robustness against heterogeneous noise.

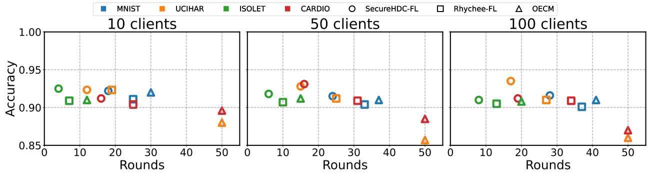

Fig. 4: Global Accuracy vs. Rounds (Feature Heterogeneity)

Ablation Study. The ablation under strong non-IID settings (nom-IID additive Gaussian noise, $\sigma = 0.5$, scale $= 0.5$; log-normal imbalance, $\gamma = 0.5$)(Table 4) confirms the necessity of the data-size weight N_{norm} and the similarity term $S^{(t)}$. Removing either component (W/O NUM and W/O COS) leads to significant declines in convergence speed and stability. Results suggest that N_{norm} ensures fairness in aggregation, while $S^{(t)}$ promotes semantic alignment, and their synergy is essential for achieving fast, accurate convergence.

Table 4: Ablation Experiment.

	Full Model		W/O NUM		W/O COS	
Dataset	Rounds	Accuracy	Rounds	Accuracy	Rounds	Accuracy
MNIST	19	0.921	28	0.905	25	0.914
UCIHAR	22	0.926	38	0.914	41	0.911
ISOLET	8	0.929	15	0.909	12	0.924
CARDIO	17	0.931	29	0.912	22	0.915

Computation and Communication Overhead. Table 5 summarizes server-side overheads. Computation time grows nearly linearly with client number, but is only marginally affected by the dimension d, and latency remains within practical ranges. Communication cost is proportional to d, with upload (UL) cost significantly higher than download (DL). This asymmetry is an inherent characteristic of the CKKS scheme, where server computations consume the modulus chain, yielding more compact ciphertexts in return. Results verify that SecureHDC-FL achieves predictable overhead and good scalability, making it suitable for practical deployment.

Table 5: Server-side computation and communication cost.

	Computation time (s)				Communication cost (MB)			
Dataset	**4K/50C**	**8K/50C**	**4K/100C**	**8K/100C**	**4K UL**	**8K UL**	**4K DL**	**8K DL**
MNIST	40.5	40.9	78.9	83.3	5.97	11.94	1.25	2.50
ISOLET	104.3	108.2	207.8	219.6	15.52	31.03	3.25	6.50
UCIHAR	24.6	25.6	48.1	50.1	3.58	7.16	0.75	1.50
CARDIO	13.3	13.5	25.7	27.6	2.39	3.58	0.50	0.75

4. CONCLUSION

This paper proposes the SecureHDC-FL protocol, which addresses the performance bottleneck of secure federated HDC under non-IID data through a novel dynamic weighted aggregation strategy and efficient homomorphic encryption. Experiments demonstrate that the method improves convergence speed by **2**× (avg. **1.6**×) and improves accuracy by up to **2.5%** (avg. **1.2%**), while ensuring privacy with reasonable overhead. This work provides a practical solution for building robust, efficient, and privacy-preserving edge federated learning systems. Our experiments are conducted in simulation, and future work will explore deployment on real edge devices for further validation.

5. REFERENCES

[1] Pentti Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, vol. 1, pp. 139–159, 2009.

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] Ligeng Zhu, Zhijian Liu, and Song Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[4] Hanchi Ren, Jingjing Deng, and Xianghua Xie, "GRNN: Generative regression neural network—a data leakage attack for federated learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–24, 2022.

[5] F. Boenisch, A. Dziedzic, R. Schuster, A.S. Shamsabadi, I. Shumailov, and N. Papernot, "When the curious abandon honesty: Federated learning is not private," in *Euro S&P*, 2023, pp. 175–199.

[6] Craig Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC '09*, 2009, pp. 169–178.

[7] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song, "Homomorphic encryption for arithmetic of approximate numbers," in *ASIACRYPT 2017*. 2017, pp. 409–437, Springer.

[8] Quanling Zhao, Kai Lee, Jeffrey Liu, Muhammad Huzaifa, Xiaofan Yu, and Tajana Rosing, "FedHD: Federated learning with hyperdimensional computing," in *MobiCom '22*, 2022, pp. 791–793.

[9] R. Chandrasekaran, K. Ergun, J. Lee, D. Nanjunda, J. Kang, and T. Rosing, "FHDnn: Communication efficient and robust federated learning for AIoT networks," in *DAC '22*, 2022, pp. 37–42.

[10] Yujin Nam, Minxuan Zhou, Saransh Gupta, et al., "Efficient machine learning on encrypted data using hyperdimensional computing," in *ISLPED '23*, pp. 1–6. IEEE, Piscataway, 2023.

[11] Yujin Nam, Abhishek Moitra, Yeshwanth Venkatesha, et al., "Rhychee-FL: Robust and efficient hyperdimensional federated learning with homomorphic encryption," in *DATE 2025*, 2025.

[12] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He, "Federated learning on non-iid data silos: An experimental study," in *ICDE 2022*. IEEE, 2022, pp. 965–978.

[13] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 7252–7261.

[14] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2020.

[15] Haokun Fang and Quan Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, pp. 94, 2021.

[16] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu, "BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning," in *USENIX ATC 2020*, 2020, pp. 493–506.

[17] Jing Ma, Si-Ahmed Naas, Stephan Sigg, and Xixiang Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5880–5901, 2022.

[18] Abdullah Al Omar, Xin Yang, Euijin Choo, and Omid Ardakanian, "Efficient privacy-preserving cross-silo federated learning with multi-key homomorphic encryption," *arXiv preprint arXiv:2505.14797*, 2025.

[19] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[20] Wen-Jie Lu, Shohei Kawasaki, and Jun Sakuma, "Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data," in *NDSS '17*. The Internet Society, 2017.

[21] Jingwei Chen, Linhan Yang, Chen Yang, et al., "Secure transformer-based neural network inference for protein sequence classification," Cryptology ePrint 2024/1851, 2024.

[22] Shai Halevi and Victor Shoup, "Algorithms in HElib," in *CRYPTO 2014*. 2014, pp. 554–571, Springer.

[23] Neveen Mohammad Hijazi, Moayad Aloqaily, Mohsen Guizani, Bassem Ouni, and Fakhri Karray, "Secure federated learning with fully homomorphic encryption for IoT communications," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4289–4300, 2023.

[24] Songyue Guo, Xu Yang, Jiyuan Feng, et al., "FedGR: Federated learning with gravitation regulation for double imbalance distribution," in *DASFAA 2023*. Springer, 2023, pp. 703–718.