HybridFL: Hybrid Approach Toward Privacy-Preserving Federated Learning

Sheraz Ali[1,4], Saqib Mamoon[2], Areeba Usman[3], Zain ul Abidin[3], and Chuan Zhao[1,4,5(✉)]

[1] School of Information Science and Engineering, University of Jinan, Jinan 250022, China
ise_zhaoc@ujn.edu.cn
[2] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[3] University of Sargodha, Sargodha, Pakistan
[4] Shandong Provincial Key Laboratory of Network-Based Intelligent Computing, University of Jinan, Jinan 250022, China
[5] Shandong Provincial Key Laboratory of Software Engineering, University of Jinan, Jinan 250022, China

Abstract. In this study, we introduce a novel Hybrid Federated Learning (HybridFL) approach aimed at enhancing privacy and accuracy in collaborative machine learning scenarios. Our methodology integrates Differential Privacy (DP) and secret sharing techniques to address inference risks during training and protect against information leakage in the output model. Drawing inspiration from recent advances, we present a HybridFL framework that combines the strengths of Homomorphic Encryption (HE) and Multi-Party Computation (MPC) to achieve secure computation without the computational overhead of pure HE methods. Our contributions include a privacy-preserving design for Federated Learning (FL) that ensures local data privacy through secret sharing while leveraging DP mechanisms for noise addition. The system offers resilience against unreliable participants and is evaluated using various machine learning models, including Convolutional Neural Networks (CNN), Multi-Layer Perceptrons (MLP), and linear regression. Furthermore, we address potential external threats by deploying predictive model outputs as robust services against inference attacks. Experimental results demonstrate improved accuracy and convergence speed, establishing the viability of HybridFL as an effective solution for collaborative machine learning with enhanced privacy guarantees.

Keywords: Machine Learning · Collaborative Machine Learning · Privacy-Preserving Federated Learning · Homomorphic Encryption · Secure Multiparty Computation · Secret Sharing · Differential Privacy

1 Introduction

In this chapter, we propose a hybrid approach to privacy-preserving collaborative ML in which we combine DP with secret sharing against inference attacks. Our approach enhances privacy and maintains the ML model's accuracy. Inspired by the hybrid scheme

J. Cai et al. (Eds.): SPNCE 2023, LNICST 525, pp. 3–18, 2025.
https://doi.org/10.1007/978-3-031-73699-5_1

[1], Stacey Truex [2] first took the initiative of applying the hybrid approach to privacy-preserving CML. This approach combined DP with HE protocol on federated learning, where they utilized the threshold variant of the pallier encryption (TPE) as an underlying security cornerstone. Although this hybrid approach provides good model performance and privacy guarantees, it faces long training time and high communication costs. Also, it can't tackle the participants dropping out during FL.

To tackle these problems, HybridAlpha [3] is an efficient privacy-preserving FL that utilizes functional encryption to perform MPC. Using FE, they proposed a simple and efficient approach that supports addition and dropping during the FL process.

In advanced cryptography, HE and MPC are closely associated with each other to solve the same problem: applying some computation on private input without expressing anything, as explained earlier in Sect. 5, except the final output. Specifically, at a large scale, HE is mostly replaceable by MPC and vice-versa. However, HE yields an expensive computation than MPC (MPC replaces this expensive computation with collaboration among two or many parties.

Secret sharing is MPC's best ingredient, which we have already explained in detail earlier. Although Stacey Truex and HybridAlpha took the good initiative of a hybrid approach, they are computationally expensive. So, we can replace the HE schemes with secret sharing where we can share differentially private shares to the global servers. This way, the participants can share their local models while providing less expensive privacy guarantees.

1.1 Motivation for Algorithm Improvement

The potential threat can take place in a collaborative machine. However, we must consider two types of potential threats: (1) inference during training and (2) inference over the outputs model.

Inference During Training: This threat can occur during the process of learning, where any client in the collaborative setting may infer information about another client's private information. Let's suppose the combination of computational functions *f(S)* and a set of queries $q_1, q_2, q_3, \ldots q_n$. In each iteration i in *f(S)*, need knowledge of participants' data there is query q_i. During execution, each participant must reply to each query q_s appropriately on the dataset. These queries are dependent on *f(S)*. For instance, a query may ask for the number of instances in the dataset marching particular criteria for a decision tree.

SMC is the best ingredient to address this risk. Usually, this protocol enables n number of parties to get the function output over their inputs while protecting the information of anything other than this output. Unfortunately, inference over the output is still a challenging risk. Since the output of the function is still the same from execution without privacy, this output may leak information about any single input. Hence, we need to consider this threat as well.

Inference over the Outputs Model: This is another potential threat at the final output model where the information can be a leak from intermediate outputs. Literature has shown most black-box attacks (i.e., membership inference attack) to the model through ML as a service API, and an adversary can still make training inferences.

Many mechanisms are designed to insert noise into the algorithm's output to achieve DP. These mechanisms add noise proportional to the output's sensitivity. The most fundamental mechanisms are the Laplacian mechanism and the Gaussian mechanism.

1.2 Contribution

Our main contributions of this work are;

1. We design a novel hybrid approach to CML system that offers recognized privacy guarantees, addresses different trust scenarios, and develops improved accuracy models. Data does not leave the client, and privacy is ensured using secret sharing and DP.
2. Our system is also robust against unreliable participants.
3. We show that our proposed system is useful for training various ML models through the experimental evaluation of CNN, MLP, and linear regression.
4. The potential threat from adversaries outside the system may occur. That's why we consider users of the final model as a potential threat. Therefore, we deploy a predictive model output from our system as a service, remaining resistant to an inference against adversaries that might be service users.

1.3 Formal Security and Privacy Models

In this section, we embark on establishing a robust theoretical foundation for our hybrid approach to privacy-preserving collaborative machine learning. We recognize the critical importance of formal security and privacy models to clarify the objectives of our research and precisely define the level of privacy we aim to guarantee.

1.3.1 Defining Security Goals

To begin, we must delineate our security goals. In the context of privacy-preserving collaborative machine learning, these goals encompass various aspects, including:

1. **Confidentiality:** Ensuring that Sensitive Data Remains Confidential and is not Exposed to Unauthorized Parties During the Collaborative Learning Process. This Involves Guarding Against Both External and Internal Threats.
2. **Integrity:** Guaranteeing the Integrity of the Collaborative Learning Process and the Resulting Model. We Want to Prevent Malicious Participants from Tampering with the Model or Data.
3. **Availability:** Ensuring that the Collaborative Learning Process Remains Available and Operational, Even in the Presence of Disruptions, Errors, or Adversarial Attempts to Disrupt the System.
4. **Robustness:** Building a System that Can Withstand Unreliable Participants or Adversarial Behavior, Maintaining Its Functionality and Privacy Guarantees.

1.3.2 Privacy Guarantees

In the realm of privacy, our primary focus is on the concept of differential privacy (DP). Differential privacy provides a mathematical framework for quantifying and achieving

privacy guarantees in data analysis and machine learning tasks. It ensures that the presence or absence of any single data point does not significantly affect the outcome or result of a computation.

Formal Definition of Differential Privacy: We adopt a formal definition of differential privacy, which can be expressed as follows:

A mechanism M satisfies (ε, δ)-differential privacy if, for all data sets D1 and D2 that differ in a single individual's data point and for all possible outcomes S of the mechanism, the following inequality holds:

$$\Pr[\mathrm{M}(D1) \in \mathrm{S}] \leq e^{(\varepsilon)} * \Pr[\mathrm{M}(D2) \in \mathrm{S}] + \delta$$

Where: ε represents the privacy parameter, controlling the level of privacy protection. δ is a parameter that provides a small additional level of privacy protection.

1.3.3 Formalizing Privacy Models

Our work will further formalize privacy models specific to the collaborative learning setting. This involves defining privacy-preserving mechanisms, assessing their privacy guarantees, and demonstrating how they align with the principles of differential privacy. We will explore how secret sharing and cryptographic primitives complement differential privacy to ensure that participants' sensitive data remains protected. By establishing these formal security and privacy models, we aim to provide not only a clear theoretical foundation but also a rigorous basis for evaluating the privacy guarantees offered by our hybrid approach. These models will guide our subsequent technical discussions and demonstrate how our method aligns with the highest standards of privacy preservation in collaborative machine learning.

1.4 Related Work

Several research efforts have explored the realm of Federated Learning (FL) and its diverse applications. Here, we highlight some key studies that contribute to this burgeoning field:

Federated Learning and Differential Privacy (DP): The fusion of Federated Learning (FL) and Differential Privacy (DP) has garnered attention due to its potential in protecting data privacy during model training. Researchers like Stacey Truex have pioneered the integration of DP mechanisms into federated learning frameworks. This combination ensures that model updates incorporate noise to safeguard sensitive information, ultimately striking a balance between model accuracy and privacy preservation.

Hybrid Approaches for Privacy-Preserving CML: Hybrid approaches that amalgamate different cryptographic techniques have emerged to enhance privacy in collaborative settings. Truex et al. proposed a hybrid scheme that combines Homomorphic Encryption (HE) with DP for privacy-preserving collaborative learning. While this approach showcases promising privacy guarantees, its performance is hindered by high computational costs and communication overhead.

Secure Multi-Party Computation (MPC) and Functional Encryption (FE): Researchers like HybridAlpha have explored the application of Secure Multi-Party Computation (MPC) and Functional Encryption (FE) to achieve privacy-preserving collaborative learning. These cryptographic techniques allow parties to perform computations over their private data without revealing sensitive information. HybridAlpha's use of functional encryption for efficient multi-party computation demonstrates a viable alternative to costly HE-based approaches.

Black-Box Attacks and Membership Inference: Privacy vulnerabilities in CML models have led to research on adversarial attacks, such as membership inference attacks. These attacks exploit model outputs to infer whether a particular data point was part of the training dataset. Countermeasures, including noise injection mechanisms like the Laplacian and Gaussian mechanisms, have been devised to enhance model privacy and thwart such attacks.

Industrial Applications and Secure Aggregation: Real-world applications of CML, such as data association within organizations, have prompted research into secure aggregation protocols. These protocols allow organizations to collaboratively train models without sharing raw data. Research in this direction seeks to strike a balance between data utility, privacy preservation, and efficient model training.

Medical Diagnostics and Data Sharing: The healthcare sector has explored collaborative learning to improve medical diagnostics. Initiatives like IBM's supercomputer for rapid diagnosis highlight the potential of collaborative approaches. However, the challenge lies in ensuring comprehensive and labeled datasets. Collaborative learning emerges as a solution to pool medical data from various sources, enhancing diagnostic accuracy while preserving patient privacy.

Marketing and Advertising with Differential Privacy: In marketing and advertising, the challenge of personalized recommendations while maintaining data privacy has been addressed using differential privacy techniques. By introducing controlled noise into recommendation algorithms, models can provide personalized suggestions without exposing individual user preferences.

2 Proposed Methodology

2.1 System Architecture

Figure 1 shows that N numbers of participants have their own sensitive data for training on the local side. The purpose of participants is to learn a joint model, i.e. the topology of local models and learning objectives are identical. Since the problem of collecting data is challenging, the participants only share parameters with the server. We also consider two servers with global models and a small auxiliary validation dataset.

Algorithm 1 illustrates the high-level working of HybridCollab. At the beginning of learning, servers and participants build their own global and local models and declare all the parameters. For each round, the participants train their own local models in a differentially private manner. Then they split the model into two shares and upload the differentially private parameter to both servers. The servers then utilize the auxiliary validation dataset to calculate a utility score for everyone and then select to receive the parameter shares with definite probability. After receiving these shares, servers average the model parameters and send them back to each participant for a new training round. The participant can dismiss its training process and drop out anytime if it is sure that the accuracy of the models is enough. Meanwhile, a new participant can also join anytime. Next, we discuss the detailed process of our system on the server side and participant side.

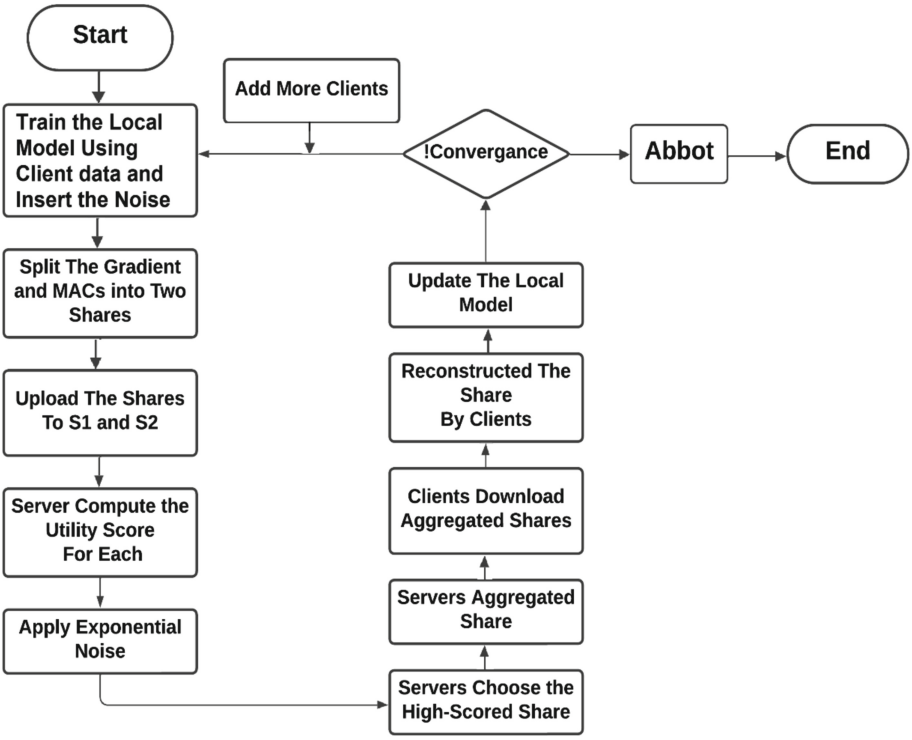

Fig. 1. Flowchart of HybridCollab

Algorithm 1: A high level description of HybridColab

1. Construct the model and define all the parameters
2. for every round of communication
3. for every participant
4. for j=1 to *l* iteration
5. Execute SGD on each local data and insert Gaussian noise
6. end for
7. Split the Update local noisy model into two shares W_1 and W_2
8. Upload W_1 and W_2 to servers S_1 and S_2
9. end for
10. The servers select the $W1$ and W_2 according to the utility score.
11. The servers compute the model average to get W_{1new} and W_{2new} and send it to the participant.
12. end for

2.1.1 HybridColab: Server Side

Algorithm 2 gives the detailed processing of HybridColab on the server side. Initially, the servers initialize the parameters and wait to receive each participant's local training parameter shares as illustrated in Fig. 2. They receive the number of participants according to a pre-fixed threshold T. The server selects participants according to high utility scores and uses exponential noise to hide the uncertainty.

$$\mathrm{P}[\text{choose participant m}] \propto \exp\left(\frac{€}{2K\Delta s} s(W, D, m)\right) \tag{1}$$

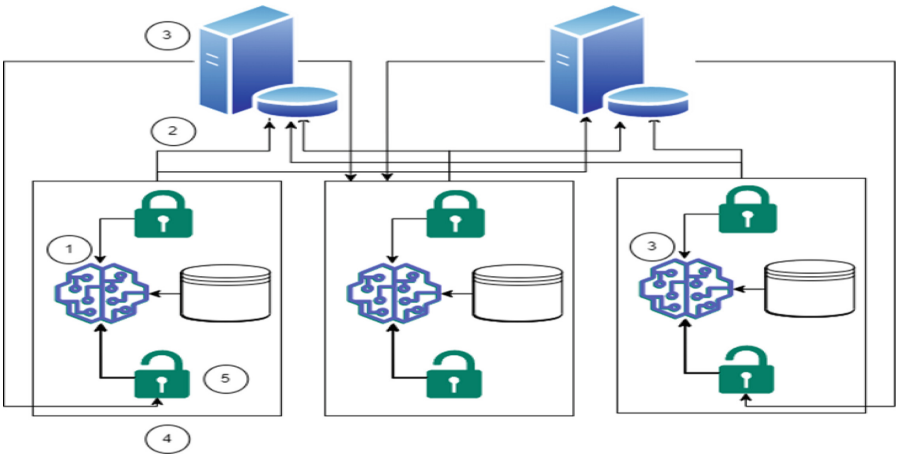

Fig. 2. High Level working principle of HybridCollab.

This process selects the number of participants at an exponential rate on the basis of the scoring function while protecting the sampling process from inferring privacy.

Hence, we can sample the almost optimum weights and hence reduce the disturbance of unreliable participants as much.

After selecting the final accepted parameter shares, the servers compute an average operation and construct the new global parameters. This average model function is the average of all the accepted parameter shares. Finally, the servers send the new global parameter shares to each participant and wait for the parameter uploading for the next round.

2.1.2 Participant Side

Algorithm 3 gives the high-level description of hybridColab on the participant side. Every participant contains two things: (1) its own local dataset and (2) its own local model. This local model is trained using standard SGD algorithm. We implement DP onto the training algorithm to preserve the participants' privacy over the output function being disclosed by the W. After that we split the sanitized model into two secret shares. Then these shares are uploaded to servers S1 and S2. We use Gaussian noise to achieve DP.

Algorithm 2: A high level description of HybridColab at servers side

1. Initialize and send the parameters to participants.
2. Weight for each participant to send their parameters.
3. Compute the utility score for every uploading participant using auxiliary validation data.
4. Choose K parameters from threshold T without replacement such that P[choose participant m] $\propto \exp\left(\frac{\epsilon}{2K\Delta s}s(W, D, m)\right)$
5. Compute model average and construct new global model and distribute to each participant.

Algorithm 3: A high level description of HybridColab at the Participant side

1. Download the same parameters from the servers.
2. In each communication round, construct the mini-batch size.
3. Insert Gaussian noise to sanitize the weights.
4. For I number of iteration,
   5. run the local SGD on the local data to obtain the updated local model.
   6. Split the model into two secret shares.6. Upload two shares to the both servers.
   7. Download the new averaged model shares from the servers.
   8. Repeat step 5 to 0 until an optimum small test error is gained.
9. Drop out

2.2 Experiment Setting and Results

The experiment is implemented on the 7th generation CPU Intel i7–7200 2.3 GHz and 64GB RAM of Dell Core i7. We simulated 2 servers and multiple participants based on M parameter. We implemented all the protocols in Python 3 programming language. We use pysyft library for ML. We use Ubuntu operating system for our experiments.

2.2.1 Experimental Setup

For all baseline scenarios, we build a CNN. We use the publicly available dataset MNIST and iris data set for a classification problem. The CNN model has two hidden layers of ReLU units and a softmax function of 10 classes with cross-entropy loss. Layer one consists of 60 neurons and layer two consists of 1000 neurons. We set the learning rate as 0.1, a batch rate of 0.01, and for DP, we utilize norm clipping of 4.0 and ϵ of 0.5. We executed an experiment for 10 participants and assigned 6,000 data points from MNIST. For the iris data set, we use logistic regression. We utilize torch with pysyft.

2.2.2 Results

The Distribution of Shares: Figure 3 illustrates the distribution of shares among both servers. The results shows that how successfully two shares split to overcome the inference during training threat.

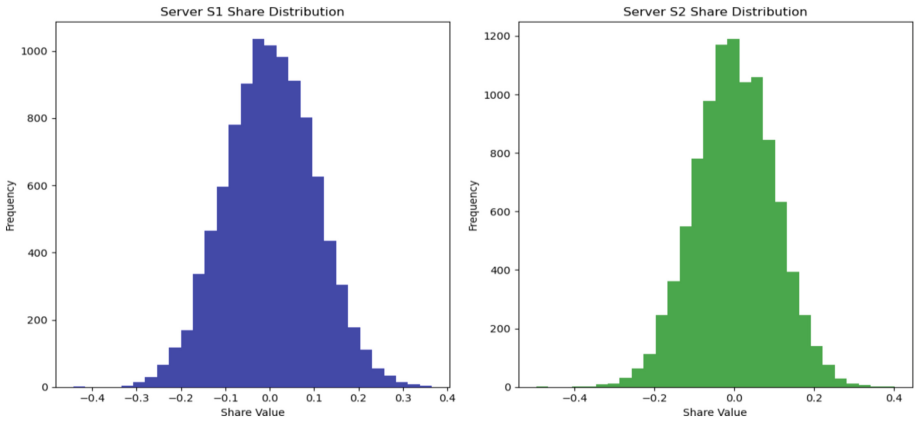

Fig. 3. The distribution of shares on server 1 and server 2

The Communication Cost. Figure 4 illustrates the impact of iterations I. The results exhibit that I will speed up the convergence by raising the computation workload on each party. However, a huge number of I will slow it down since the collaboration reduces. I will increase the convergence speed by increasing the computation cost of each participant. But, note that a too large value of I will reduce the convergence as the collaboration reduces. On this basis, we set the parameter I to be 100 in our experiments.

Training Performance. Figure 5 illustrates the overall performance of over training models. We achieve 89% of accuracy. We plot the variation of accuracy with epochs. We can see that with increase in epochs, our training is increasing respectively.

Robustness against Inconsistent Clients: In Fig. 6, we display the percentage of Inconsistent Clients across different variations (40%, 60%, and 80%). We also introduce random noise into their data. We have set the values of M and K to be 0.5N and 0.5M, respectively. Based on the variation of N clients and the proportion of noise data p, we

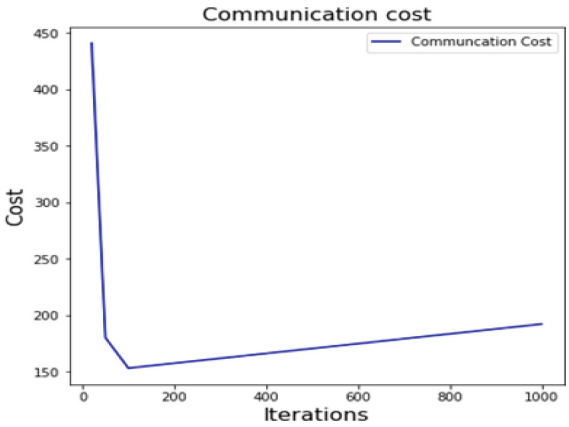

Fig. 4. Impact of Iteration over Communication cost

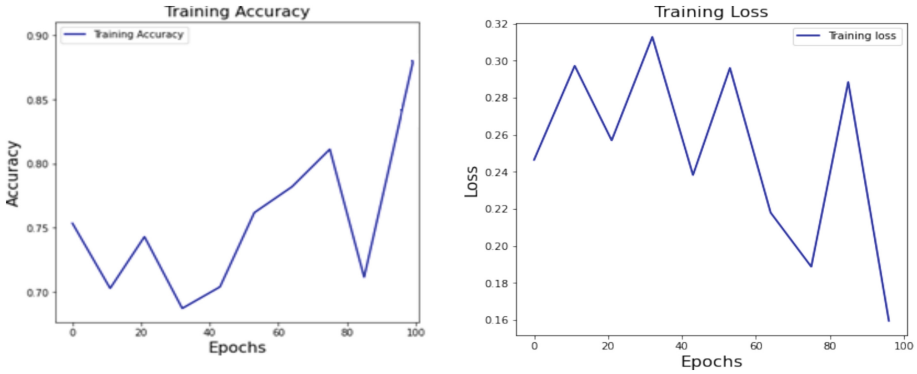

Fig. 5. Left: obtained 92% accuracy, Right: the loss is reduced to 20%.

can observe that half of the clients are unreliable. We have assigned N values of 40, 70, and 100, with corresponding p values of 0.3, 0.5, and 0.7. For sampling K clients, we have set it to be half of M clients, assuming that the most reliable clients are present. We also compare our secureShare scheme with PSA and FL. PSL and FL did not consider the existence of Inconsistent Clients. Figure 6 shows how the impact of Inconsistent Clients reduces.

2.2.3 Security and Privacy Analysis of HybridFL

Our proposed Hybrid Federated Learning (HybridFL) framework is designed with a strong emphasis on enhancing security and privacy in collaborative machine learning scenarios. Here, we provide a comprehensive analysis of the security and privacy aspects of HybridFL, highlighting its capabilities and safeguards against potential threats.

1. Differential Privacy (DP): HybridFL employs Differential Privacy as a fundamental building block to ensure data privacy during model training. By introducing controlled

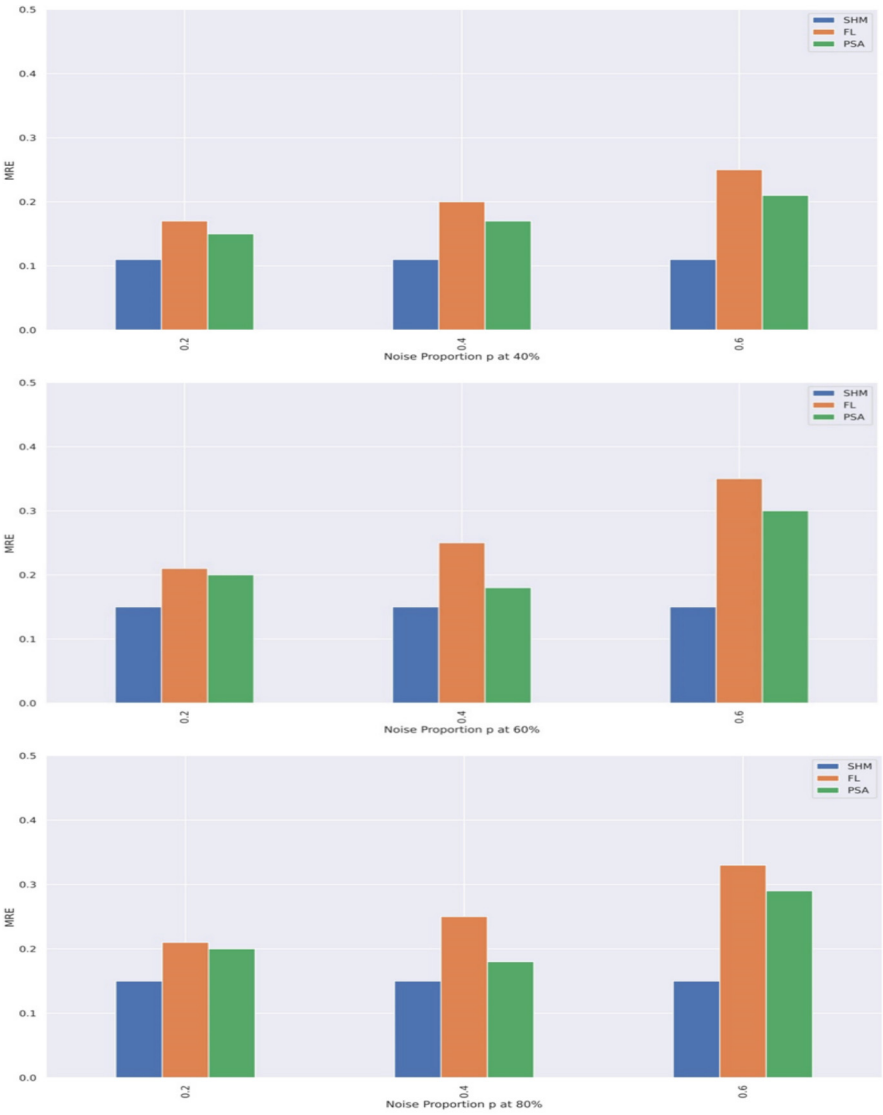

Fig. 6. Robustness against Inconsistent Clients variations (40%, 60%, and 80%)

noise to the training process, DP mitigates the risks of inference attacks and information leakage from individual participants' data. This noise addition guarantees that no single participant's contribution can be pinpointed, offering a robust defense against privacy breaches.

2. Secret Sharing: A significant aspect of HybridFL is the utilization of secret sharing to distribute model parameters among participating servers. Secret sharing ensures that no individual server possesses the complete model, enhancing security against server

compromise or data leakage. Even if an adversary gains access to one server, they cannot reconstruct the model without access to the shares held by other servers.

3. Inference Attacks: HybridFL addresses both training and output inference attacks. During training, Secure Multi-Party Computation (SMC) techniques are applied to prevent participants from inferring sensitive information about others' data. However, inference over the output model remains a challenge. To counter this, HybridFL incorporates mechanisms like DP noise addition and secret sharing, rendering it more resilient to various forms of black-box and membership inference attacks.

4. Participant Dropout and Addition: HybridFL demonstrates adaptability to participant dropout and addition, a critical consideration in real-world scenarios. A participant can leave the system without compromising the overall training process, and new participants can seamlessly join ongoing training. This feature ensures that system dynamics remain stable and effective, even in dynamic collaborative environments.

5. Communication Privacy: HybridFL reduces the need for participants to share raw data while allowing them to collaborate for model improvement. This decentralized approach minimizes the risk of data exposure during communication rounds, offering increased communication privacy and reduced vulnerability to interception attacks.

6. Utility Score Selection: The selection of participants based on a utility score, as outlined in Algorithm 3, ensures that only participants with meaningful contributions are included in the parameter aggregation process. This approach prevents malicious actors from manipulating the system by consistently adding noise and affecting model accuracy.

7. Limitations: While HybridFL presents a comprehensive approach to security and privacy, it's important to acknowledge its limitations. The efficiency of HybridFL relies on efficient Secure Multi-Party Computation and the proper handling of noise addition. Additionally, the system assumes honest majority participants to ensure secure parameter selection.

8. Case Study: The application of our hybrid approach in healthcare data collaboration demonstrates its profound impact. Through the integration of differential privacy, secret sharing, and cryptographic techniques, our approach achieves a dual objective: accurate disease prediction and robust data privacy. Collaboration among hospitals, research institutions, and pharmaceutical companies takes on a new level of effectiveness and productivity. This synergy results in significant advancements in medical research, all while safeguarding the privacy of patient data.

Furthermore, the inclusion of robustness measures in our approach acts as a formidable defense against potential adversarial actions. This resilience ensures that the collaborative effort remains impervious to privacy attacks. Our hybrid approach not only elevates healthcare data collaboration for disease prediction but also serves as a pioneering model for privacy-preserving machine learning across sensitive domains. It sets a benchmark for compliance, security, and innovation in the realm of healthcare research.

3 Conclusion and Discussion

3.1 Application of HybridFL

3.1.1 Organization Data Association

HybridFL transcends being just a benchmark for expertise; it has evolved into an essential industrial paradigm. With the surge in big data, the challenge arises of efficiently gathering, processing, and disseminating data using remote processors, akin to cloud computing. However, the vital consideration of data security and its close association with an organization's sensitive information and revenues raises pertinent questions. The industrial application of collaborative learning offers a contemporary solution for managing extensive datasets. For instance, when inaccessible data ownership hampers decision-making, collaborative learning enables organizations to share model insights without compromising data ownership.

3.1.2 Medical Diagnosis

The realm of rapid medical diagnosis marries treatment and artificial intelligence, but current diagnostic methods are often time-consuming and lack intelligence. Addressing these challenges, an innovative system employing collaborative learning emerges as a potential solution. Notably, IBM's supercomputing technology has been at the forefront of quick medical analyses, particularly in cancer diagnosis. However, recent instances of misdiagnosis have surfaced due to incomplete training data, lacking crucial disease features, medical test reports, and gene sequences. The underlying issue is the paucity of comprehensive and labeled data sources.

To counter these limitations, medical associations are joining forces to pool data and collaboratively train superior machine learning models. HybridFL can serves as the enabler of this data sharing and processing, leading to enhanced medical diagnostics.

3.1.3 Marketing and Advertisement

HybridFL can revolutionizes data security in domains such as banking and advertising, where safeguarding raw data from malicious use is paramount. Here, data privacy concerns and the confidentiality of intellectual property make direct data sharing for model training impractical. Collaborative learning provides an avenue to train models without exposing sensitive data.

In the realm of target marketing, the concept hinges on providing personalized Machine Learning-as-a-Service, analogous to Amazon's product recommendations. Features such as customer profiles, purchasing behavior, and product attributes form the basis of data used for modeling. However, ensuring robust data privacy and protection proves challenging, especially when dealing with the intricacies of social networks, banks, and online stores. The inherent complexity of data aggregation is compounded by data heterogeneity across various organizational silos.

HybridFL bridges this gap by enabling the creation of training models using data from geographically dispersed organizations, without necessitating data exchange. By doing so, it facilitates effective model training across a diverse spectrum of data sources while maintaining data privacy and security.

4 Conclusion

Our work introduces the innovative concept of Hybrid Federated Learning (HybridFL) as a robust approach to address privacy and accuracy concerns in collaborative machine learning. By combining Differential Privacy (DP) and secret sharing techniques, we create a synergistic framework that overcomes challenges related to inference attacks during training and output model leakage. We have demonstrated that our HybridFL approach not only maintains model accuracy but also significantly enhances data privacy. This is achieved by leveraging the strengths of both Homomorphic Encryption (HE) and Multi-Party Computation (MPC) while avoiding the computational overhead associated with a solely HE-based approach. Our contributions encompass a comprehensive system architecture that integrates secret sharing, DP mechanisms, and participant selection strategies. The experimental evaluations, involving various machine learning models and datasets, validate the effectiveness of our HybridFL system in practical scenarios. Notably, our system proves resilient against potential threats posed by unreliable participants and external adversaries. As the realm of collaborative machine learning continues to grow, our HybridFL framework provides a pioneering solution that strikes a balance between privacy preservation and model accuracy. By deploying predictive model outputs as robust services, we ensure a heightened level of resistance against inference attacks from service users.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (62472252, 62172258), TaiShan Scholars Program (tsqn202211280), Shandong Provincial Natural Science Foundation (ZR2024QF131, ZR2023LZH014, ZR2022ZD01, ZR2022MF264, ZR2021LZH007), Shandong Provincial Key R&D Program of China (2021SFGC0401, 2021CXGC010103), Department of Science & Technology of Shandong Province (SYS202201), and Quan Cheng Laboratory (QCLZD202302).

References

1. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
2. Shao, Z.C.: A new efficient (t, n) verifiable multi-secret sharing (vmss) based on ych scheme. Appl. Math. Comput. **168**(1), 135–140 (2005)
3. Bai, L.: A strong ramp secret sharing scheme using matrix projection. In: Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, pp. 652–656. IEEE Computer Society (2006)
4. Iftene, S.: General secret sharing based on the chinese remainder theorem with applications in e-voting. Elec. Notes Theor. Comput. Sci. **186**, 67–84 (2007)
5. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Conference on Artificial Intelligence and Statistics (2017)
6. Li, T., Sahu, A.K., Sanjabi, M., Zaheer, M., Talwalkar, A., Smith, V.: Federated optimization for heterogeneous networks. arXiv preprint arXiv:1812.06127 (2018)
7. Du, W., Han, Y.S., Chen, S.: Privacy-preserving multivariate statistical analysis: linear regression and classification. In: Proceedings Of SDM 2004, SIAM, vol. 4, pp. 222–233 (2004)

8. Chaudhuri, K., Monteleoni, C.: Privacy-preserving logistic regression. In: Proceedings of NIPS 2009, pp. 289– 296 (2009)
9. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed kmeans clustering over arbitrarily partitioned data. In: Proceedinds of KDD 2005, pp. 593–599. ACM (2005)
10. "Deep learning and differential privacy (2016)." https://github.com/frankmcsherry/blog/blob/master/posts/2017-10-27.md
11. Biggio, B., Fumera, G., Roli, F.: Security evaluation of pattern classifiers under attack. IEEE Trans. Knowl. Data Eng. **36**(4), 984–996, April 2014
12. Wikipedia, Cryptography. https://en.wikipedia.org/wiki/Cryptography. Accessed 02 Aug 2020
13. Li, M., Andersen, D.G., Park, J,W., et al.: Scaling distributed machine learning with the parameter server. In: 11th USENIX Symposium on O, perating Systems Design and Implementation (OSDI 14), pp. 583–598 (2014). https://doi.org/10.1145/2640087.2644155
14. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979). https://doi.org/10.1145/359168.359176
15. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1310–1321. ACM (2015)
16. Yu, H., Vaidya, J., Jiang, X.: Privacy-preserving SVM classification on vertically partitioned data. In: Ng, W.K., Kitsuregawa, M., Li, J., Chang, K. (eds.) Advances in Knowledge Discovery and Data Mining. PAKDD 2006. Lecture Notes in Computer Science(), vol. 3918. Springer, Heidelberg (2006). https://doi.org/10.1007/11731139_74
17. Vaidya, J., Yu, H., Jiang, X.: Privacy-preserving SVM classification. Knowl. Inf. Syst. **14**(2), 161–178 (2008). https://doi.org/10.1007/s10115-007-0073-7
18. Lindell, Y., Pinkas, B.: Privacy-preserving data mining. In: Annual International Cryptology Conference, pp. 36–54. Springer, Heidelberg (2000). https://doi.org/10.1145/335191.335438
19. Du, W., Han, Y.S., Chen, S.: Privacy-preserving multivariate statistical analysis: linear regression and classification. In: Proceedings of the 2004 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, pp. 222–233 (2004). https://doi.org/10.1137/1.9781611972740.21
20. Sanil, A.P., Karr, A.F., Lin, X., et al.: Privacy-preserving regression modelling via distributed computation. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 677–682. ACM (2004). https://doi.org/10.1145/1014052.1014139
21. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 593–599. ACM (2005). https://doi.org/10.1145/1081870.1081942
22. Ali Sheraz, et al.: Towards privacy-preserving deep learning: opportunities and challenges. In: 2020 IEEE 7th International Conference on Data Science and Advance Analalytics
23. Riazi, M.S., Weinert, C., Tkachenko, O., et al.: Chameleon: a hybrid secure computation framework for machine learning applications. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security, pp. 707–721. ACM (2018). https://doi.org/10.1145/3196494.3196522
24. Xu, R., et al.: Hybridalpha: an efficient approach for privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security (2019)
25. Nikolaenko, V., Ioannidis, S., Weinsberg, U., et al.: Privacy-preserving matrix factorization. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, pp. 801–812. ACM (2013). https://doi.org/10.1145/2508859.2516751

26. Mohassel, P., Zhang, Y.: Secureml: a system for scalable privacy-preserving machine learning. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 19–38. IEEE (2017). https://doi.org/10.1109/SP.2017.12
27. Gilad-Bachrach, R., Dowlin, N., Laine, K., et al.: Cryptonets: applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning, pp. 201–210 (2016)
28. Proserpio, D., Goldberg, S., McSherry, F.: Calibrating data to sensitivity in private data analysis: a platform for differentially-private analysis of weighted datasets. Proc. VLDB 2014 **7**(8), 637–648 (2014)
29. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1310–1321. ACM (2015)
30. Bonawitz, K., Ivanov, V., Kreuter, B., et al.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191. ACM (2017). https://doi.org/10.1145/3133956.3133982
31. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of CCS 2015, pp. 1310–1321. ACM (2015)
32. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of CCS 2015, pp. 1322–1333. ACM (2015)
33. Juvekar, C., Vaikuntanathan, V., Chandrakasan, A.: Gazelle: a low latency framework for secure neural network inference, arXiv preprint arXiv:1801.05507
34. "Deep learning and differential privacy," https://github.com/ frankmcsherry/blog/blob/master/posts/2017–10–27.md, 2016
35. Biggio, B., Fumera, G., Roli, F.: 'Security evaluation of pattern classifiers under attack.' IEEE Trans. Knowl. Data Eng. **36**(4), 984–996 (2014)
36. Wikipedia, Cryptography. https://en.wikipedia.org/wiki/Cryptography. Accessed 02 Aug 2020
37. Techtarget,cryptography. https://searchsecurity.techtarget.com/definition/cryptography. Accessed 02 Aug 2020
38. Gibson, A., Patterson, J.: Chapter 4. Major Architectures of Deep Networks. O'Reilly. https://www.oreilly.com/library/view/deeplearning/9781491924570/ch04.html
39. Wagh, S., Gupta, D., Chandran, N.: SecureNN: 3-party secure computation for neural network training. In: Proceedings on Privacy Enhancing Technologies, vol. 1, p. 24 (2019). https://doi.org/10.2478/popets-2019-0035
40. Konen, J., McMahan, H.B., Yu, F.X., et al.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
41. Even, H., Goldreich, O., Lempel, A.: A randomized proto-col for signing contracts. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds). CRYPTO 1982, pp. 205–210. Plenum Press, New York (1982). (Page 4)
42. Yao, A.C.-C.: How to generate and exchange secrets (extendedabstract). In: 27th FOCS, pp. 162–167. IEEE Computer Society Press, October 1986. (Page 4)
43. Beaver, D., Micali, S., Rogaway, P.: The round complexity ofsecure protocols (extended abstract). In: 22nd ACM STOC, pp. 503–513. ACM Press, May 1990. (Pages 4 and 9)