A Utilization Method of Big Data in Blockchain Based on Swarm Learning

Yiran Cao[1], Haiyan Kang[1(✉)], Yanfang Li[2(✉)], and Zhiyong Zhang[3]

[1] Department of Information Security, Beijing Information Science and Technology University, Beijing 100192, China
kanghaiyan@126.com
[2] Beijing College of Finance and Commerce, Beijing 101101, China
lyf2080@126.com
[3] School of Information Science and Technology, Hebei Agricultural University, Baoding 071051, China

Abstract. In the era of artificial intelligence, data security and privacy protection have become a core focus. As a distributed ledger technology, blockchain faces the challenges of data silos and privacy leaks while working with massive amounts of data. To address these issues, this manuscript proposes a local differential privacy based on swarm learning (LDP-SL). The method firstly employs localized differential privacy to ensure that when data is collected and processed on the end node, personal private data is protected by introducing noise to prevent differential and inference attacks. Secondly, leveraging the characteristics of swarm learning, this manuscript achieves data sharing and the aggregation of collective wisdom without directly transmitting raw data, which effectively avoids the probability of data leakage. Furthermore, the manuscript designs a model combining convolutional neural networks (CNN), which can adapt to datasets of different types and dimensions. By introducing Laplace and Gaussian mechanism differential privacy techniques, it ensures privacy protection during the model training process. This method also does a lot of work on no independent and identically distributed datasets. Through experimental validation, the method proposed in this manuscript effectively utilizes massive data in blockchain for data analysis and decision support while protecting data privacy.

Keywords: Differential Privacy · Swarm Learning · Deep Learning · Blockchain

1 Introduction

With the rapid advancement of information technology, human-beings have reached the era of artificial intelligence. In this age, data has become one of the most valuable resources, driving business decisions and serving as a crucial foundation for scientific and technological innovation [1, 2]. However, the exponential growth and widespread application of data also present challenges of data leakage and privacy protection. This

D.-J. Deng and J.-C. Chen (Eds.): SGIoT 2024, LNICST 633, pp. 107–121, 2026.
https://doi.org/10.1007/978-3-031-93825-2_10

issue is particularly pronounced in the realm of blockchain, where the unique characteristic of a distributed ledger offers new possibilities for data storage, transmission, and transactions, but simultaneously gives rise to issues such as data silos and privacy breaches [3, 4].

Data silos emerge when data fails to be shared effectively across different systems and platforms. This not only hampers the full realization of data's value but also constrains cross-disciplinary collaborative innovation [5, 6]. Furthermore, the risk of privacy leakage escalates with the increase in data volume, posing the pressing issue of how to achieve efficient data utilization while protecting personal privacy.

In traditional machine-learning models, data is collected from user endpoints and uploaded to a central server for training [7, 8]. However, considering of privacy concerns, Federated Learning (FL) has been proposed to alleviate users' anxiety regarding the security of their private data. FL has evolved to address the requirements of complicated machine learning scenarios, for example, continuous learning or data diversity, particularly in the context of distributed training involving data from multiple locations [9, 10]. Unlike traditional methods, FL adopts a decentralized collaborative training philosophy, where a central server is still maintained, but user data does not need to be uploaded; instead, learning takes place through iterative gradient updates [11]. However, as technology advances, FL also faces challenges such as insecure servers, gradient leakage, and malicious participants, which may lead to the malicious collection or inference of private data. As a result, swarm learning has emerged [12].

Swarm Learning (SL) represents a decentralized machine learning paradigm that eschews the need for a central server for training purposes. Within this framework, multiple nodes take part in the training process without a central authority [13, 14]. In each iteration of training, a participant node is chosen at random to serve as a provisional aggregator, collecting model refinements from every node involved. This setup eliminates the necessity for nodes to disclose their proprietary data sets, thereby safeguarding the integrity and confidentiality of the model aggregation process. Moreover, SL combines blockchain-driven peer-to-peer architectures with edge computing to facilitate equitable and secure participation among nodes, bypassing the need for centralized coordination.

Swarm learning, as an emerging data processing approach, pools computing power from various sources to tackle complex problems. By applying swarm learning to blockchain, it is possible to break data silos while ensuring the compliant use of data without compromising personal privacy. This offers new avenues for addressing data challenges in blockchain systems.

In the Internet environment, SL faces many new challenges that are distinct from those of mainstream AI paradigms. For example, the development of traditional AI applications relies on large amounts of training data, whereas in more realistic scenarios of SL, data is often scattered among different corporate, institutional, or even massive mobile user groups. Barriers exist between these data silos, making it difficult to integrate them using conventional means [15]. Meanwhile, growing concern about data privacy and security has become a global trend. The General Data Protection Regulation (GDPR) was implemented by the European Union in 2018, the world's toughest data privacy protection regime to date, which introduced a series of measures to safeguard privacy

in the context of big data, mobile internet, and artificial intelligence. China has also enacted laws such as the Cybersecurity Law and the Personal Information Protection Law, which include provisions on data privacy protection. In summary, under the current global wave of data privacy protection, the issue of data silos has become particularly acute, making it increasingly challenging to fulfill the need for joint group data to build models and put SL techniques into practice [16].

This manuscript aims to present a method for the compliant utilization of massive data in blockchain based on swarm learning. Firstly, it examines the main challenges faced in current blockchain data processing, such as data silos and privacy leakage. Then, it provides a detailed account of the proposed data processing framework, which leverages local differential privacy technology and the characteristics of swarm learning. Furthermore, it demonstrates how to integrate convolutional neural networks to design a hybrid model adaptable to various types of data, and incorporate differential privacy technology during the model training process to safeguard data privacy. Finally, the efficacy of the proposed method is validated through experiments, and its potential and challenges in practical applications are discussed.

This manuscript employs swarm learning techniques to train and learn from client users' personal data assets. To prevent attackers from launching parameter inference attacks and safeguard data from unauthorized leakage or loss by untrustworthy third parties, local differential privacy is adopted to inject noise into the swarm learning model parameters, replacing methods such as homomorphic encryption. This approach reduces communication costs, computational costs, and latency, making it more widely applicable on cost-effective and low-performance devices.

The main contributions of this manuscript can be summarized as follows:

1. Development and design of a privacy protection model that ensures high security and low participation cost, safeguarding user data from collection and storage to the security of the entire communication chain.
2. Utilization of swarm learning technology to legally leverage user's personal asset data, which fully exploits the data available on user devices and addresses the issue of data silos.
3. Adoption of local differential privacy for perturbing the swarm learning model parameters, replacing homomorphic encryption and secure multi-party computation, thereby reducing communication costs, computational costs, and latency. This enables more widespread application on cost-effective and low-performance devices.

2 Background Knowledge and Related Work

2.1 Background Knowledge

Differential Privacy. A groundbreaking concept introduced by Dwork in 2006, employs a method of randomized answering to ensure that the influence of any single data record on the released dataset remains within a predetermined threshold. This mechanism effectively thwarts malicious attackers from inferring or extracting information about specific records in the dataset by monitoring minute changes in the output, thereby safeguarding users' private data.

First, let's define differential privacy. Suppose there are two adjacent datasets that differ by only one piece of data, which we call *Data* and *Data′*. For these two adjacent datasets, there are X objects, which either have a certain property A or not, forming sets $M\{a_1, a_2, \cdots\cdots, a_{10}\}$ (containing $a_x = 0$ or 1) and $M'\{b_1, b_2, \cdots\cdots, b_{10}\}$. If the property A in the set changes, or if there is only a unique a_i such that $a_i \neq b_i$, then we say that M and M' are neighboring sets. A randomized algorithm is one that, given a fixed input, produces an output according to some mathematical distribution rather than always producing the same output. Differential privacy refers to a process where, if we run a certain random algorithm S on two adjacent datasets, the outputs are nearly identical. Formally, this can be defined as: $Pr\{A(Data) = O\} \leq e^{\varepsilon} \times Pr\{A(Data') = O\}$. An algorithm satisfies differential privacy if, for any adjacent datasets on which it operates, the probability of obtaining a particular output 0 is roughly the same. The simplest way to implement differential privacy is to add noise into the data, either at the input or the output, thus fuzzing the true data. Many studies and applications opt for the Gaussian mechanism or the Laplacian mechanism to introduce the noise.

Swarm Learning. In the realm of contemporary deep learning, models are often constructed using vast datasets housed on central servers, which poses significant privacy and security risks regarding the raw data. To address these concerns, federated learning has emerged, allowing data to remain on local participant nodes. Despite this, federated learning is still susceptible to sophisticated attacks such as data reconstruction and member inference, leading to potential data breaches. In response, a novel distributed learning technique known as swarm learning has been introduced to facilitate fully decentralized model training.

The architecture of Swarm Learning consists of numerous edge nodes within a swarm. Each node retrieves an initial model from the network and then refines it using its own proprietary data. Subsequently, these nodes disseminate their model parameters across the network. Security is bolstered by the authentication, authorization, and registration of these nodes into smart contracts within a blockchain-based peer-to-peer network. During the training phase, any node has the potential to be designated as a temporary aggregator for model synthesis. Once a local model meets a preset synchronization criterion, such as a specific training batch size, the selected nodes convey their model parameters through the Swarm API. The leading nodes then consolidate these into a global model using a weighted average approach and distribute the updated global model parameters back to the participating nodes. This cycle of model refinement continues until convergence is achieved.

2.2 Related Work

In traditional machine-learning models, data is typically collected by user endpoints and uploaded for centralized model training. However, this approach has sparked public concern over the protection of personal data privacy. To address this issue, federated learning (FL), which does not need a central sever, has emerged. FL enables intelligent data processing while preserving user privacy, as it allows models to be trained on multiple dispersed nodes with no need to collect user's data, thereby reducing the risk of data breaches. Despite this, FL still faces several challenges, including security issues

related to unreliable central server, the risk of gradient leakage, and potential threats from malicious participants, all of which could lead to the exposure or malicious inference of user privacy data.

To address these challenges, Swarm Learning has been proposed as a more advanced distributed learning framework. Swarm learning builds on the strengths of FL while further enhancing the system's security and resistance to attacks. By leveraging blockchain technology and smart contracts, federated learning enables a more decentralized and democratic training process, where each participating node gets a chance to act as a temporary leader responsible for aggregating model parameters. This decentralized design reduces dependence on a single central server, thereby lowering the risk of a targeted attack on the system and enhancing resilience against malicious behavior. As such, federated learning presents a novel approach to safeguarding user privacy and enhancing data security.

Xiang et al. [17] proposed an Industrial Internet of Things (IIoT) architecture based on Digital Twin (DT) and Credibility-Weighted Swarm Learning (CSL) to enhance the security of traditional swarm learning by calculating the credibility value of each party to identify high-quality model training and computational capabilities. However, it is unable to defend against semi-honest parties' inference attacks. Kang et al. [18] proposed a hierarchical Stackelberg game-theoretic incentive mechanism for wireless edge networks, which is applicable to address the issues of uneven distribution of computing resources and high communication costs in edge devices linked to blockchain. Chen et al. [19] presented a defense mechanism against backdoor threats in Swarm Learning (SL) by using a digital signature-based method for node authentication and defining node credibility as the basis for model verification in the model validation phase and trust update in the model aggregation stage, thus realizing dynamic measurement of node credibility. Yang et al. [20] presented an enchanced method of Federated Learning based on YOLOv7, extending the usage of FL to the field of real-time dynamic data processing. However, it is still unable to defend against inference attacks by semi-honest participants. Ling et al. [21] proposed an improved lightweight Paillier homomorphic encryption algorithm, which utilizes the Chinese remainder theorem to optimize the encryption process, reducing the time for modular exponentiation. However, it still incurs computational overhead and time cost for resource-constrained edge devices. Zhang et al. [22] introduced differential privacy techniques into the federated learning process by adding Gaussian noise to gradient values to protect node's private data. However, they only employed relaxed differential privacy techniques and did not specify strict differential privacy scenarios. Fan et al. [23] proposed a edge learning method called CB-DSL, which, inspired by swarm intelligence techniques, employs the particle swarm optimization algorithm to find the global optimum of complex optimization problems, achieving decent accuracy. However, it still suffers from high computational costs.

Therefore, to address the data silos problem in blockchain and make massive data in blockchain comply with usage regulations, this manuscript applies swarm learning to learn users' data assets. Inspired by the idea of 'data is available but invisible', local differential privacy technology is employed to prevent semi-honest participants from leaking or losing data, and the parameters of federated learning are perturbed. Meanwhile, by combining strict differential privacy and loose differential privacy technologies, which

are used as alternatives to homomorphic encryption and secure multi-party computation, the communication cost, computation cost, and time cost are reduced, thus enabling wide application of swarm learning in edge devices with low performance.

3 LDP-SL Method Design

3.1 Description of the Problem

The rapid growth and wide application of data bring challenges to data security and privacy protection. Especially in the realm of blockchain, the unique decentralized ledger characteristics offer new possibilities for data storage, transmission, and exchange. Blockchain's on-chain descriptions provide a mechanism for ensuring data immutability and transparency, as all transactions and data are recorded on a public ledger for verification by all network participants. However, this all-transparent method of data recording is not suitable for all types of data, particularly those that require privacy protection or are related to business secrets. As a result, many applications opt to store this sensitive data off-chain to safeguard their privacy and security. While off-chain storage provides flexibility and privacy, it also gives rise to the issue of data silos. Since off-chain data is not published, data sharing and interoperability between different blockchain networks or applications are restricted. Each off-chain storage system may employ different technology stacks and data formats, further complicating data aggregation. Moreover, the risk of privacy breaches increases with the volume of data, raising the pressing need to balance the protection of personal privacy with the effective utilization of data.

Therefore, the goal of this manuscript is to develop a localized differentially private SL method that enables secure and effective training when clients are honest but curious, thus protecting the privacy of their data and model parameters from being inferred. Specifically, during the SL model training process, where T rounds of iterations are required to complete the global model training, k clients are selected in each iteration t to

Table 1. Relevant Notations and Meanings

Notation	Meaning
N	Number of participants in federated learning
C_i	The i-th participant in federated learning
D	The sum of all participants' training datasets
D_i	The dataset corresponding to the i-th participant
ε	Privacy budget defined in local differential privacy
δ	The related parameter defined in local differential privacy
M	Perturbation mechanism in local differential privacy
σ	Variance of the noise mechanism in local differential privacy
q	Sampling rate in each learning round of federated learning
w	Model parameters in federated learning
T	Total number of communication rounds in federated learning
Δs	Sensitivity in local differential privacy

train the initial model using their local datasets. Each client then sends the updated model to the corresponding data provider. To prevent privacy leakage during the transmission of models trained by clients, it is necessary to design a localized differential privacy mechanism to protect the model parameters in transit. The symbols and parameters related to this study are listed in Table 1.

3.2 Local Differential Privacy Mechanism Based on Swarm Learning

To address the probability of data privacy leakage in swarm learning caused by the presence of semi-honest participants, this manuscript proposes an LDP-SL method, as illustrated in Fig. 1. The method consists of N swarm learning participants, each having a local training dataset and an initially trainable model with other participants.

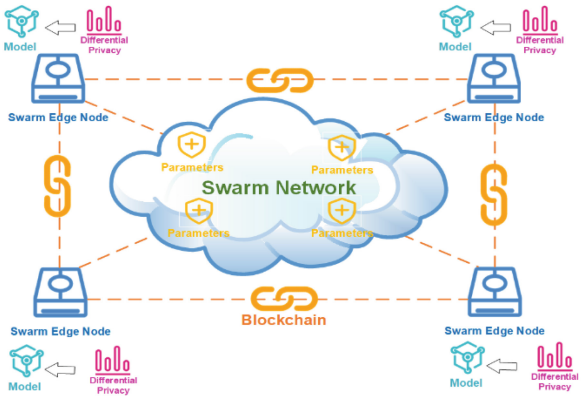


Fig. 1. LDP-SL System architecture diagram

The core idea of LDP-SL is to introduce local differential privacy into swarm learning. Specifically, the participant first selects the model to be trained, finds the corresponding training participants, and acquires the initial model. Then, the participant trains the model with its local dataset. During the local training process at the participant's end, local differential privacy is introduced to disrupts the model parameters. By transmitting the perturbed parameters (not the primary data), the goal of privacy protection is achieved. The participating parties aggregate all the parameters after receiving the perturbation parameters, feed the aggregated parameters back to the parties, and iterate until convergence.

The model consists of two primary modules: a differential privacy mechanism module responsible for adding noise to the locally trained data to enhance data security, and a swarm learning module that distributes models, collects updates, and uses the federated averaging algorithm for model aggregation on the server. In addition to these two modules, there is a console module that, when inputting parameters, determines the type of data set distribution used during system computation, the local model training on the user end, the differential privacy mechanism employed, and the size of the privacy budget in differential privacy calculations, among other parameters. There is also a result

output function that utilizes a visualization data chart library to plot the accuracy curve of the aggregated model. Lastly, there is a variable control function that ensures data consistency across learning processes, assisting the model in training.

The procedure of the algorithm SL_Update is shown in Algorithm 1. First, the list of model parameters to be trained and the accuracy list of the test set are initialized. Secondly, the number of iterations is set. In each iteration, k out of N participants are randomly selected for training with a sampling rate q (where $q = \frac{k}{N}$). For the selected k participants, the global model parameters w obtained in the last iteration are passed to the participant local update algorithm SL_Local (Algorithm 2), which is executed in parallel by the k participants. The parameters of the model in this iteration are obtained separately. Finally, after the parties complete the update operation, the training server aggregates the perturbation parameters, that is, calculates their weighted average to acquire the global parameters, and uses the test set to calculate the corresponding model accuracy.

Algorithm 1: SL_Update
Input: Number of participants N, sampling rate q, number of learning rounds T
Output: Accuracy

Step 1: Initialize w_0
Step 2: for t from 1 to T do
Step 3: Select k participants from N with sampling rate q
Step 4: For each of the k participants selected from N, do
Step 5: $w_{t+1}^{k} \leftarrow SL\_Local(k, w_t)$
Step 6: Aggregation: $w_{t+1} \leftarrow \frac{1}{n}\sum_{i=1}^{k} w_{t+1}^{ii}$
Step 7: Calculate model accuracy
Step 8: end for

From the user's perspective, if a balanced dataset is used, data is randomly sampled uniformly from each class to form a training set, which is then used for differential privacy calculations. The perturbed data is then used to train the model provided by the server, and the updates are sent back to the server. The server uses these updates to calculate the parameters. If an imbalanced dataset is set, the imbalanced dataset is first expanded by a factor equal to the number of users, and then the same number of data points are randomly sampled uniformly from each class to form a training set. This set is used for differential privacy calculations, and the subsequent steps involving the server remain the same.

Algorithm 2: SL_Local
Input: Model parameters learned from the previous round w_t, local model iteration count E, size of the local dataset m, batch size B for stochastic gradient descent, learning rate α for stochastic gradient descent, privacy parameters ε_i and δ_i for the localized differential privacy mechanism.
Output: Perturbed parameter values $\widetilde{w^k}$

Step 1: for $e = 1$ to E do
Step 2: for each data pair b in the training set B do
Step 3: gradient magnitude $g \leftarrow \nabla \mathrm{L}(w, b)$
Step 4: Calculate sensitivity $\Delta s = \frac{2C}{m}$
Step 5: Compute noise scale $\sigma_i = \frac{\Delta s \sqrt{2qTln\left(\frac{1}{\delta_i}\right)}}{\varepsilon_i}$
Step 6: Perturb the parameters $\widetilde{w^k} \leftarrow w^k + N(0, \sigma_i^2)$
Step 7: end for

4 Experiment and Analysis

4.1 Experimental Setting-Ups

Experimental Environment. This section delves into the assessment of the LDP-SL approach introduced in this manuscript, alongside a comparative experimental setup. The testing environment is Windows 10 (64-bit), with Pycharm as the integrated development environment, Python 3.8 as the scripting language, a 13th Gen Intel(R) Core(TM) i9-13900H @2.70 GHz processor, and 64 GB of RAM. For the deep learning model training, PyTorch version 1.11.1 was utilized, employing a Convolutional Neural Network (CNN) to implement the LDP-SL method outlined here. The CNN architecture comprises 2 convolutional layers with 16 and 32 filters, respectively, equipped with a 5 × 5 kernel size and a stride of 2. Additionally, a dense layer is included, featuring an input shape of 7 × 7 × 32 and an output dimension of 10. The training regimen employs a batch size of 64 for gradient descent optimization, with each participant conducting 10 iterations of local training.

Experimental Dataset. Employing the MNIST dataset, which is a classic and frequently utilized dataset, it consists of handwritten images of the numbers 0 to 9 along with their corresponding digit labels (the actual numbers the handwritten images represent). This dataset comprises 60,000 plus 10,000 28 × 28 pixel handwritten digit images. A balanced dataset refers to a collection of data that follows an independent and identically distributed (IID) principle. In probability theory, if *n* numbers are randomly drawn from a dataset, and these n random variables are mutually independent and follow the same distribution, then these n random variables are said to be independently and identically distributed. This dataset is a balanced one. At the outset of machine learning, it is essential to assume that the sample data is balanced so that the model can be trained effectively.

An imbalanced dataset is one in which the data does not conform to the independent and identically distributed (i.i.d.) assumption. Randomly selected n numbers usually have bias and cover a wide range of categories. In real-world applications, data often faces the challenge of variation among different users. Due to individual differences in preferences, interests, and domains, the collected data features can be heavily skewed, with some ranges being extremely abundant and others sparse, resulting in an imbalanced dataset. When dealing with non-i.i.d. data, a common approach is to increase the number of users to broaden the imbalanced dataset, providing more data points for each feature range to draw data independently and identically distributed for model training.

The MNIST dataset used in this project inherently satisfies the i.i.d. condition and is considered a balanced dataset. In this project, the handling of imbalanced datasets involves doubling the number of users in the drawn dataset and then extracting an equal number of data samples from each class to form a small balanced dataset for model training.

Evaluation Metrics. To verify the superiority of this method, the original federated averaging algorithm, FedAvg, is selected as the baseline. The following three metrics are primarily used for evaluation:

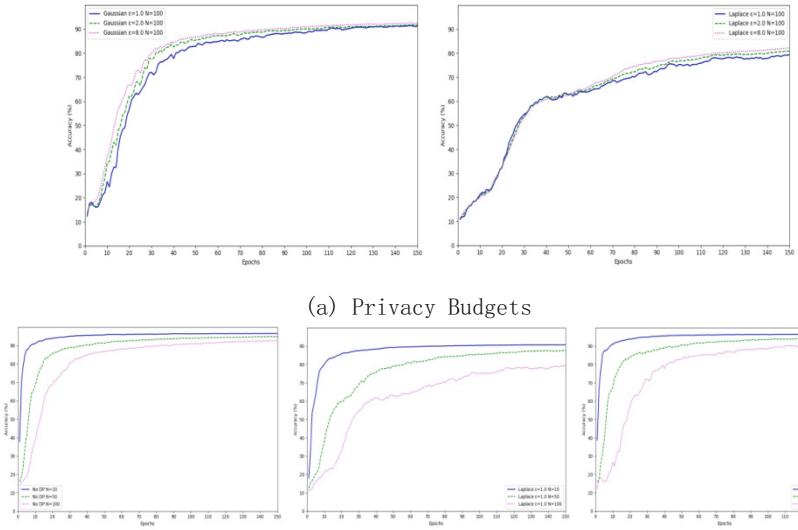
1. Global accuracy: The global accuracy of a model after several iterations is a key metric to evaluate the effectiveness of algorithms. Comparing the global accuracy of various algorithms under the same conditions makes it possible to directly evaluate their performance.
2. Performance loss: Performance loss is a metric that quantifies the performance degradation of the model, calculated through a performance estimation mechanism.
3. Runtime: The runtime of an algorithm is a key metric for measuring communication overhead. The longer the runtime, the higher the communication overhead.

4.2 Model Effectiveness Evaluation Experiment

This section investigates model validity. Utilizing the MNIST dataset, with the number of SL iterations $T = 150$, $\delta = 0.01$, and each party's privacy budget $\varepsilon_i = \varepsilon$, with $\sigma_i = 10^{-5}$. First, the impact of the privacy budget on the global accuracy is explored, with a sampling rate $q = 1$ and 100 parties ($N = 100$), using privacy budgets ε of 1.0, 5.0, and 10.0, as depicted in the Fig. 2. Next, the effect of the number of parties is studied, maintaining a sampling rate $q = 1$ and a privacy budget ε of 1.0, while varying the number of participants: $N = 10$, $N = 50$, and $N = 100$, as shown in the Fig. 2.

By observing the chart, 3 conclusions can be drawn:

1. Under the same number of participants and sampling rate, the higher the privacy budget in the LDP-SL, the more it indicates that the balance between the privacy and usability of the swarm learning model can be achieved by adjusting the privacy budget.
2. Under the same privacy budget and sampling rate, the more participants in the LDP-SL, the lower the global accuracy of the model, suggesting that an increase in the number of swarm learning participants can impact the accuracy.
3. In the above experiments, the global accuracy of the LDP-SL method tends to stabilize after about 100 iterations, indicating good model usability.



(a) Privacy Budgets

(b)Number of Participants

Fig. 2. LDP-SL Model Effectiveness Evaluation Experiment

4.3 Comparison Experiment and Analysis

Global Accuracy Analysis. First, investigate the global accuracy comparison of adding Laplace noise and Gaussian noise in the proposed LDP-SL method with the case without differential privacy noise on the MNIST dataset. Set the number of participants N = 100, sampling rate q = 1, privacy budget $\varepsilon_i = \varepsilon$, $\sigma_i = 10^{-5}$, and $\varepsilon = 1.0$. Figure 3 shows the experimental results of the global accuracy comparison under different noise conditions in the swarm learning scheme.

As shown in Fig. 3 above, the following conclusions can be drawn:

1. The global accuracy of the LDP-SL with differential privacy protection on the MNIST dataset in condition of the same number of participants, is consistently lower than that without differential privacy noise, indicating that the introduction of noise mechanism impacts the accuracy of the swarm learning model compared to the scenario without differential privacy noise.
2. Under the same conditions of both the number of participants and the privacy budget, the learning model with added Gaussian noise converges faster, suggesting that the addition of Laplace noise has a more pronounced impact on the accuracy of swarm learning in the experimental context.
3. Due to the differing complexities of Laplace noise and Gaussian noise, the model that incorporates Gaussian noise exhibits superior accuracy performance in the comparative group where noise is added.

Comparison of Performance Loss Rate. Secondly, this study investigates the performance loss comparison between Laplace noise and Gaussian noise in the LDP-SL

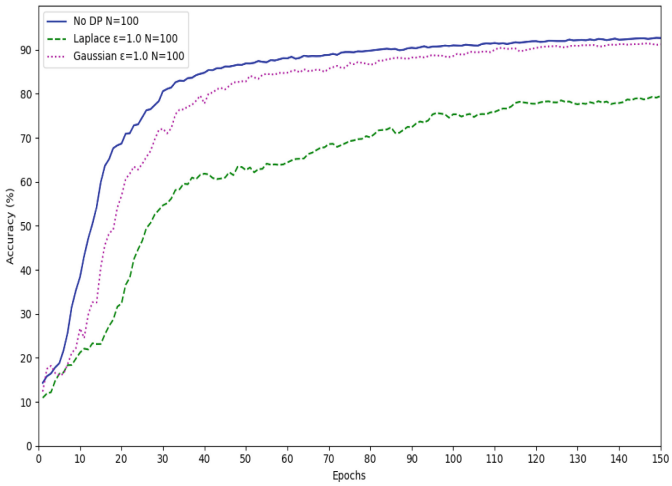


Fig. 3. The Variation of Global Accuracy with the Number of Iterations

method on the MNIST dataset. The parameters are set as follows: the number of participants N = 100, sampling rate q = 1, privacy budget $\varepsilon_i = \varepsilon$ for each participant, $\sigma_i = 10^{-5}$, and $\varepsilon = 4.0$. Figure 4 presents the experimental results of performance loss comparison under different noise conditions for the crowd-sourced learning scheme. As illustrated in Fig. 4, the LDP-SL method demonstrates a performance loss within reasonable limits under different noise conditions, indicating the feasibility of the LDP-SL method's performance.

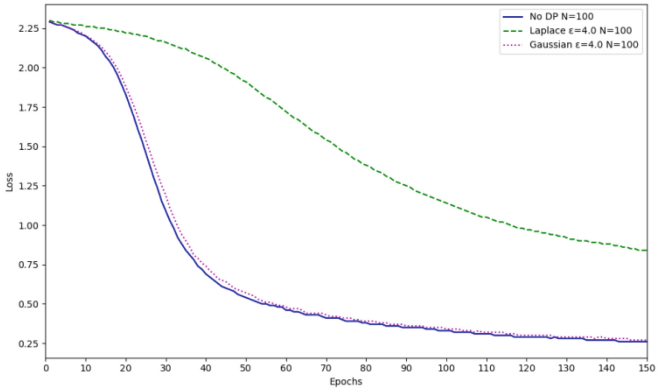


Fig. 4. The Variation of Performance Loss Rate with the Number of Iterations

Algorithm Runtime Comparison. Lastly, investigate the comparative runtime performance of the LDP-SL method proposed in this manuscript on the MNIST dataset when

different types of noise are added. The number of participants is set to N = 10, with a sampling rate q = 1, each participant's privacy budget $\varepsilon_i = \varepsilon$, $\sigma_i = 10^{-5}$, and $\varepsilon = 1.0$. Figure 5 illustrates the runtime variations over the number of computation rounds for two noise addition methods and a no-noise method.

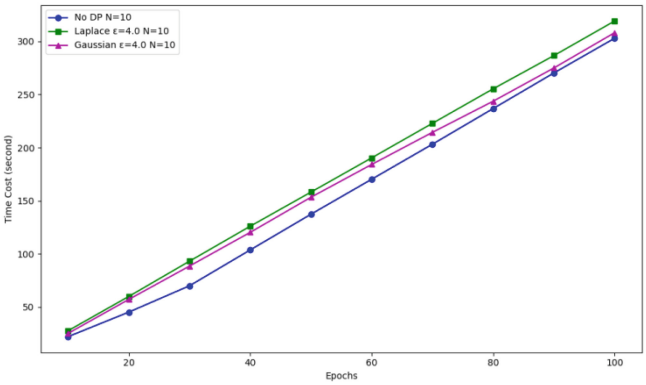


Fig. 5. The Runtime Variation with the Number of Participants

Observing Fig. 5, the following conclusions can be drawn.

1. As the number of computation rounds increases, the runtime of all three methods on the dataset also increases, indicating that an increase in computation rounds leads to longer algorithm runtime.
2. Under the same conditions, the noiseless method has the shortest runtime; among the privacy-preserving schemes that introduce noise mechanisms, the method that introduces Gaussian noise has the shortest runtime, while the method that introduces Laplace noise has the longest runtime.

5 Conclusion

Despite progress has been made in applying differential privacy technology to swarm learning, several pressing issues remain to be addressed. Swarm learning demands an extremely large volume of sample data, often requiring the collection of hundreds of millions of user data samples in practical cases to achieve high accuracy. Some intelligent application companies have found that after deploying localized differential privacy technology on user devices, models trained with private data require a greater amount of data compared to those without added noise, sometimes by as much as two orders of magnitude. Particularly when dealing with high-dimensional data such as autonomous driving, local differential privacy methods face challenges between data usability and privacy. Moreover, in localized swarm learning, the absence of a central server for coordination makes it difficult for users to ascertain information about other users' data, thereby complicating the determination of an appropriate range for random noise addition. This may lead to a decline in the global accuracy of the learning model due

to uneven noise distribution. Therefore, to promote the extensive, efficient, and secure development of the artificial intelligence field, it is necessary to further delve into and research these issues to ensure the secure use of data.

Acknowledgment. The research was financially supported by the National Social Science Foundation of China, with the grant number 21BTQ079, as well as the Beijing Future Blockchain and Privacy Computing Advanced Innovation Center, under the project number GJJ-24.

References

1. Gao, Y., Chen, X., Zhang, Y., et al.: A survey on attacks and defenses in federated learning systems. J. Comput. Sci. Technol. **46**(9), 1781–1805 (2023)
2. Warnat-Herresthal, S., Schultze, H., Shastry, K.L., et al.: Swarm Learning for decentralized and confidential clinical machine learning. Nature **594**(7862), 265–270 (2021)
3. Kang, H., Ji, S.: Hierarchical Stackelberg Game 4. Swarm Learning Incentive Method for Wireless Edge Network. Acta Electronica Sinica. **52**(7), 2382–2392 (2024)
4. Tianlong, G., Long, L., Liang, C., et al.: A survey on fair federated learning and its design. J. Comput. Sci. Technol. **46**(9), 1991–2024 (2023)
5. Lingtao, T., Zoneng, C., Lufei, Z., et al.: Research progress on privacy issues in federated learning. Journal of Software **34**(1), 197–229 (2023)
6. Xiaoshi, W., Haiyan, K.: Research on noise addition and accuracy analysis in differential privacy. J. Lanzhou Univ. Technol. **49**(3), 94–103 (2023)
7. Madni, H.A., Umer, R.M., Foresti, G.L.: Blockchain-based swarm learning for the mitigation of gradient leakage in federated learning. IEEE Access **11**, 16549–16556 (2023)
8. Kang, H., Ji, Y.: Research on time-serial location data publication based on local differential privacy. Acta Electronica Sinica **50**(9), 2222–2232 (2022)
9. Sun, S., Huang, H., Peng, T., et al.: A data privacy protection diagnosis framework for multiple machines vibration signals based on a swarm learning algorithm. IEEE Trans. Instrum. Meas. **72**, 1–9 (2023)
10. Martínez Beltrán, E.T., Pérez, M.Q., Sánchez, P.M.S., et al.: Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges. IEEE Communications Surveys & Tutorials **25**(4), 2983–3013 (2023)
11. Wen, J., Zhang, Z., Lan, Y., et al.: A survey on federated learning: challenges and applications. Int. J. Machine Learn. Cybernet. **14**(2), 513–535 (2023)
12. Liu, Y., Huo, L., Wu, J., et al.: Swarm learning-based dynamic optimal management for traffic congestion in 6G-driven intelligent transportation system. IEEE Trans. Intell. Transp. Syst. **24**(7), 7831–7846 (2023)
13. Guendouzi, B.S., Ouchani, S., el Assaad, H., et al.: A systematic review of federated learning: challenges, aggregation methods, and development tools. J. Netw. Comput. Appl. **220**, 103714 (2023)
14. Wei, L., Yuzhao, L., Congke, T., et al.: Research on federated distillation data sharing model based on blockchain. Computer Science **51**(3), 39–47 (2024)
15. Qi, P., Chiaro, D., Guzzo, A., et al.: Model aggregation techniques in federated learning: a comprehensive survey. Futur. Gener. Comput. Syst. **150**, 272–293 (2024)
16. Kang, H., Wang, J.: Swarm Mutual Learning. Complex & Intelligent System, 1–15 (2024)
17. Xiang, W., Li, J., Zhou, Y., et al.: Digital twin empowered industrial IoT based on credibility-weighted swarm learning. IEEE Trans. Industr. Inf. **20**(1), 775–784 (2024)

18. Kang, H., Ji, S.: Hierarchical stackelberg game swarm learning incentive method for wireless edge network. Acta Electron. Sin. **52**(7), 2382–2392 (2024)
19. Chen, B., Li, G.: Backdoor threat defense in group learning based on multi-level trust measurement. Modern Info. Technol. Sci. **7**(18), 119–124+128 (2023)
20. Yujie, Y., Haiyan, K.: An enhanced detection method of PCB defect based on improved YOLOv7. Electronics **12**(9), 1–18 (2023)
21. Ling, H.: Research on Lightweight Homomorphic Encryption and Consensus Algorithms for Group Learning. Changchun University of Technology (2024)
22. Lei, Z.: Identity Authentication Enhancement Method Based on Differential Privacy Group Learning. Beijing University of Posts and Telecommunications (2024)
23. Fan, X., Wang, Y., Huo, Y., et al.: CB-DSL: communication-efficient and byzantine-robust distributed swarm learning on Non-i.i.d. Data. IEEE Trans. Cognitive Comm. Network. **10**(1), 322–334 (2024)