Differential Privacy Technology of Big Data Information Security based on ACA-DMLP

Yubiao Han*, Lei Wang, Dianhong He
Shandong Information Technology Industry Development
Research Institute, Jinan Shandong 250014, China

***Abstract*—Cloud computing and artificial intelligence have a deeper and closer connection with daily life. To ensure information security, most companies or individuals choose to pay a simple fee to store a large amount of data on cloud servers and hand over a large number of complex calculations of machine learning to cloud servers. To eliminate the security risks of data stored in the cloud and ensure that private data is not leaked, this paper proposes a collusion-resistant distributed machine learning scheme. Through homomorphic encryption algorithm and differential privacy algorithm, the security of data and model in machine learning framework is guaranteed. The distributed machine learning framework is adopted to reduce the data computing time and improve the data training efficiency. The simulation results show that the computational efficiency is improved while the user privacy security is guaranteed. The accuracy of model training is not reduced due to the improvement of privacy data security and computational efficiency. Through this study, we can further propose effective measures for the privacy protection of outsourced data and the data integrity of machine learning, which is of great significance to the security research of cloud intelligent big data.**

Keywords*—Big data; cloud computing; information security; distributed machine learning; differential privacy algorithms

I. INTRODUCTION

The core technology of artificial intelligence is machine learning. Machine learning is mainly through the analysis of a large number of data, statistics, calculations, and other operations, from which to learn experience, build models, and step by step improve the accuracy of model training. In practice, machine learning is widely used for model prediction in medicine, banking, recommendation systems, threat analysis, and authentication technology. Over time, large amounts of data are collected to provide new solutions to old problems [1]. Large-scale Internet companies collect users' online activities and recommend services of interest to users in the future through the analysis of big data. Health data from different hospitals and government agencies can be used to produce new diagnostic models, while financial companies and payment networks can also combine transaction history, merchant data, and account holder information to train more accurate fraud detection engines [2]. Although the progress of technology at this stage makes the processing and computing of big data more efficient, it is still an important challenge to ensure the privacy and security of cloud data. Competitive advantages, privacy concerns, laws and regulations, and issues surrounding data sovereignty and jurisdiction have hindered the development of data training techniques by many outsourcing companies [3]. The algorithm workflow of distributed machine learning can be summarized as follows: the system receives large-scale data and stores them in the cloud, and then communicates data in the distributed network. Each distributed computing node performs the corresponding computing task after receiving the required data, and the system aggregates the sub-models trained by each node [4]. The main bottleneck in the work is the privacy security during data training and the model security after each node trains the sub-model, and the efficiency and accuracy of model training cannot be reduced by improving the security. McMahan et al. employed a differential privacy technique on a distributed parallel architecture to enable a trusted server to add noise to the weighted average of user updates to guarantee the user-level privacy [5]. The aggregation scheme of Adadi et al. is proved to be secure in the semi-honest adversary environment, especially when the secure multi-party computation (MPC) computes the sum of individual local user model updates at the cost of computational cost and communication overhead [6]. Shakeel P. et al. proved that when the server is not trusted, differential privacy cannot rely on the server to complete the task of adding noise, and a small part of the original gradient can be used to explain the local data [7]. Li et al. used the federated learning method to protect the user's privacy, but it increased the cost of computation and storage while protecting the privacy security [8]. Elgabli et al. combined the distributed differential privacy with a three-layer encryption protocol and proposed an unbiased coding algorithm to reduce the mean square error to achieve a better trade-off and combination of security and efficiency [9]. This paper is based on the data analysis and calculation of cloud server ciphertext transmission and machine learning distributed training platform. A distributed machine learning scheme (Distributed Machine Learning Privacy-protection Against Collusion Attacks, ACA-DMLP) against collusion attacks is proposed. This comprises the following steps of: adding the Laplace noise disturbance to a ciphertext of a user by a cloud end, and performing disturbance processing on each piece of ciphertext data distributed to a training platform through a differential privacy algorithm. Through the unsolvability of the system of indeterminate equations in the algorithm, the collusion attack of the adversary is prevented. The security evaluation and efficiency performance analysis of the scheme is carried out through simulation experiments. The main innovations of this paper are:

1) This paper proposes a private data encryption scheme that supports multiple users to encrypt private data with

different public keys at any time and upload it to the cloud server to encrypt user data efficiently.

2) Establish a mechanism for the cloud server to add noise to the ciphertext data to efficiently protect the transmitted data.

3) An efficient distributed machine learning scheme is designed, and an anti-collusion attack algorithm is proposed to protect the privacy of each training node, which ensures the security of user privacy data and the training model of each node.

II. Related Work

A. Distributed Machine Learning

Distributed Machine Learning is mainly used to study how to use multiple computers to train large-scale data models. Big data has a large volume of data, many types of data, and high commercial value. Big data and cloud computing cannot be separated. With the rise of big data, cloud computing is bound to develop. However, big data cannot be processed by a single computer, so users have to adopt a distributed computing architecture. Therefore, distributed machine learning has also been developed rapidly. However, before the theory and technology related to big data were proposed, there had been a lot of related research work in the industry. In order to make the speed of data calculation and model training faster in machine learning, multiple computers or servers are used to run at the same time. Parallel processing is generally called "parallel computing" or "parallel machine learning". Its main purpose is to decompose a large computing task into multiple small computing tasks, and then distribute them to multiple computers or processing nodes in a distributed architecture for processing and computing. Nowadays, under the dual challenges of large-scale data and large-scale models, there are newer and higher standards and requirements for the computing power and storage capacity of servers used in machine learning:

1) The calculation is more difficult and more complex, so that the previous simple parallel calculation may take a lot of time. Therefore, there is an urgent need for a processor or computer cluster with higher parallelism and computing power to complete the data training task.

2) The volume of data is large and the required storage capacity is large, which leads to the fact that a single machine cannot meet the data storage needs at all, so more and more schemes have to adopt the distributed cluster architecture for data storage.

B. Differential Privacy Technology

Because of its strong background assumptions, differential privacy has become a mainstream security algorithm in the privacy protection schemes related to machine learning. It can even be said that in the field of cryptography, any algorithm related to privacy protection can use differential privacy [10]. Generally speaking, the most powerful thing about differential privacy is that as long as every step in the algorithm meets the requirements of differential privacy, it can ensure that the final output of the algorithm still meets the requirements of differential privacy [11].

m_1 and m_2 are two adjacent data sets with different records, which are called adjacent data sets (also known as brother data sets). Differential privacy uses the Laplace mechanism to add measurable disturbance to the ciphertext to ensure the security of data distributed by cloud servers [12].

Definition 1: $\varepsilon - DP$: $\varepsilon - DP$ means that if there is a pair of adjacent data sets m_1 and m_2, and K is within the range of R, then the mechanism R belongs to $\varepsilon - DP$, then the following holds:

$$\Pr\left[R(m_1) = K\right] \le e^{\varepsilon} \Pr\left[R(m_2) = K\right] \tag{1}$$

Where ε is the privacy budget, which refers to the number of bits of information that the data analyst DA can obtain. The smaller ε is, the less bits of information that the data analyst DA can obtain. The stronger the secrecy of $\varepsilon - DP$ is, and the randomness of differential privacy ensures the robustness of differential privacy [13].

Definition 2: Sensitivity: f is a function in the input space of the data set, i.e., $f : m \to R^d$, which is used to describe the mapping function of a data set m to a d-dimensional space [14]. Δf represents the sensitivity of two adjacent data sets, and has the following calculation formula:

$$\Delta f = \max_{m_1, m_2} \left\| f(m_1) - f(m_2) \right\| \tag{2}$$

Where, on R^d with at most one different piece of data, the maximum value is on the pair of m_1 and m_2. $\left\| f(m_1) - f(m_2) \right\|$ represents the Manhattan distance from a point in the data set in the real domain to a point in the data set m_2, which is called the 1-norm. For various different pairs of m_1 and m_2 data sets, finding the maximum distance is the sensitivity [15]. Differential privacy means that for a data set with only one record difference, the probability obtained by query is close. The closer the probability is, the stronger the confidentiality of the algorithm to the private data is. If the results of two data queries are completely consistent, the data set has been completely randomized [16]. In this way, the data will lose its availability again and again to improve security. Privacy protection will lose its original role and significance. Most of the schemes make the query probability close, not exactly the same, hoping to find a balance between the security and availability of private data [17].

III. Big Data Information System Scheme

A. System Model

The system model diagram of the ACA-DMLP scheme is shown in Fig. 1.

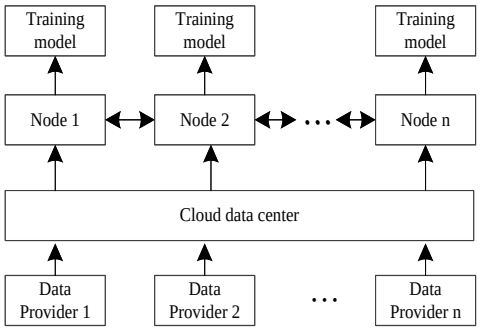

Fig. 1. System Model of the ACA-DMLP Scheme.

A DP is a group of a plurality of different data providers $DP = \{DP_1, DP_2, \cdots, DP_n\}$ in the scheme model, and provides data from a plurality of different sources for a cloud server. The data providers encrypt their own private data and submit the data to a cloud server CS. The cloud server adds a noise mask to ciphertext data and distributes the ciphertext data according to the requirements of a data analyst. Before the private data set is outsourced to the cloud server for storage, each data provider will use its own public key $pk_{DP_i}(i = 1, 2, \cdots, n)$ to encrypt the sensitive data in its data set, and then entrust it to the cloud server for storage and computation [18].

B. Scheme Described

While the efficient training is distributed in the working nodes, the privacy data security and the sub-models trained by each distributed node are protected [19]. Fig. 2 is a system flow chart of that ACA-DMLP scheme.

The data provider DP uploads a large number of ciphertext data sets to the cloud server. The cloud server adds noise disturbance to the ciphertext data sets through the differential privacy scheme based on the Laplace mechanism and trains the logistic regression model through multiple iterations of multiple training nodes on the cloud platform. At the same time, it ensures that the adversary colludes with one or more computers in the distributed cluster and will not leak the encrypted data set distributed by the cloud server to the data analyst and the sub-model that has been trained by the computing nodes [20].

To improve the real-time and dynamic performance of data uploaded by users and ensure the security of data, this paper proposes a homomorphic encryption privacy protection scheme, in which each data provider has a public key. The privacy data is dynamically encrypted in real-time by combining the XOR operator of homomorphic encryption and the Diffie-Hellman theory of separable computation. Then upload it to the cloud server. Even if an adversary steals the cloud data, the plaintext cannot be cracked. A hash function is added to the algorithm to ensure the security of the ciphertext. Finally, through the security analysis and proof of the scheme, the feasibility and data security of the scheme are theoretically explained.

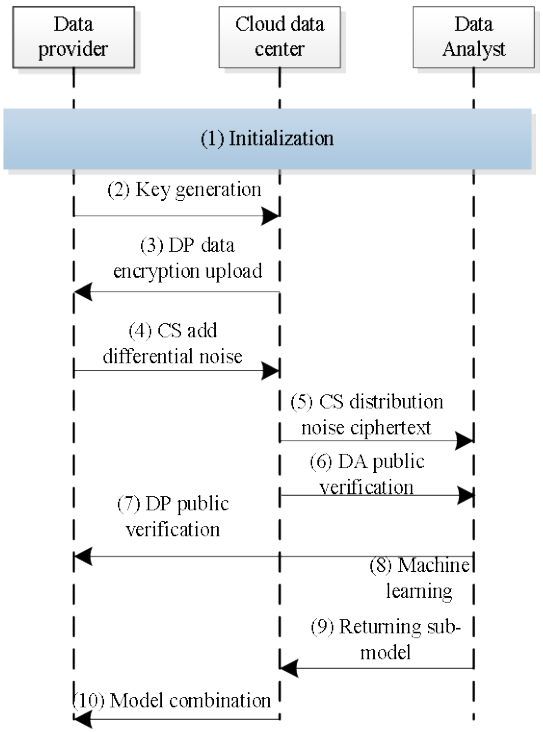

Fig. 2. System Flow Chart of the ACA-DMLP Scheme.

C. Programme Framework Structure

The main content of the ACA-DMLP scheme is that the cloud server adds noise to the ciphertext data and then distributes the disturbing data to each working node of the data analyst. Due to the unsolvability of the indeterminate equations used by differential privacy to add noise, the adversary cannot theoretically steal the disturbed data through a reverse attack. Because of the particularity of the distributed architecture, the adversary cannot steal the trained sub-model of working nodes by conspiring with one or more hosts [21].

On the coordinate plane, the Manhattan distance between point i with coordinate (x_1, y_1) and point y with coordinate (x_2, y_2) is calculated as:

$$d(i, j) = |x_1 - x_2| + |y_1 - y_2| \tag{3}$$

If the scheme satisfies differential privacy, if and only if the following expression holds:

$$C(d) = f(d) + Lap\left(\frac{\Delta f}{\varepsilon}\right) \tag{4}$$

$C(d)$ is the output function encrypted by the differential privacy algorithm, that is, each data set of the differential privacy algorithm outputs a ciphertext. $f(d)$ is the ciphertext data received by the cloud server, that is, the cloud server adds noise to the ciphertext to execute the input function $f(d) = (x_1, x_2, \cdots, x_n)^T$ of the differential algorithm. T

represents the transpose of the vector [22]. Converts the input data set to a vector. $Lap\left(\frac{\Delta f}{\varepsilon}\right)$ is the noise perturbation added by the cloud server to the encrypted data. It is added in the form of a vector in the scheme. The form of adding noise disturbance is the vector addition operation between the vector of the original ciphertext data set and the noise vector, as shown in Formula 5:

$$C(d)=f(d)+\left(Lap_1\left(\frac{\Delta f}{\varepsilon}\right),Lap_2\left(\frac{\Delta f}{\varepsilon}\right),\cdots,Lap_n\left(\frac{\Delta f}{\varepsilon}\right)\right)^T \tag{5}$$

Formula (4) is an algorithm formula for the cloud server to add noise to each piece of ciphertext data (n pieces of data in total) in the scheme. In differential privacy, as usual $\mu=0, b=\frac{\Delta f}{\varepsilon}$, the Laplace function is written as:

$$Lap\left(\frac{\Delta f}{\varepsilon}\right)=\frac{1}{(2\Delta f)/\varepsilon}e^{\frac{-|x|}{(\Delta f)/\varepsilon}} \tag{6}$$

Simplified as follows:

$$Lap\left(\frac{\Delta f}{\varepsilon}\right)=\frac{\varepsilon}{2\Delta f}\exp\left(\frac{-\varepsilon|x|}{\Delta f}\right) \tag{7}$$

The initial ciphertext $f(d)=(x_1,x_2,\cdots,x_n)^T$ is differenced, that is, summed with the Laplace noise vector, to give the following equation:

$$C(d)=(x_1,x_2,\cdots,x_n)^T+\left(Lap_1\left(\frac{\Delta f}{\varepsilon}\right),Lap_2\left(\frac{\Delta f}{\varepsilon}\right),\cdots,Lap_n\left(\frac{\Delta f}{\varepsilon}\right)\right)^T \tag{8}$$

The following formula can be obtained by substituting the above formula (5) into the vector addition formula (6) and then transposing and expanding it:

$$\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix}+\begin{pmatrix}Lap_1\left(\frac{\Delta f}{\varepsilon}\right)\\Lap_2\left(\frac{\Delta f}{\varepsilon}\right)\\\vdots\\Lap_n\left(\frac{\Delta f}{\varepsilon}\right)\end{pmatrix}=\begin{pmatrix}x_1+\frac{\varepsilon}{2\Delta f_1}\exp\left(\frac{-\varepsilon|x_1|}{\Delta f_1}\right)\\x_2+\frac{\varepsilon}{2\Delta f_2}\exp\left(\frac{-\varepsilon|x_2|}{\Delta f_2}\right)\\\vdots\\x_n+\frac{\varepsilon}{2\Delta f_n}\exp\left(\frac{-\varepsilon|x_n|}{\Delta f_n}\right)\end{pmatrix}=C(d) \tag{9}$$

Assume that there are N computing nodes in total in the data analyst, and each node allocates i (i is a random number in $1,2,\cdots,n$) pieces of encrypted data. Then the noisy ciphertext data allocated by each working node is:

$$\begin{cases}C_1(d)=f_1(d)+Lap_1\left(\Delta f_1/\varepsilon\right)\\C_2(d)=f_2(d)+Lap_2\left(\Delta f_2/\varepsilon\right)\\\vdots\\C_i(d)=f_i(d)+Lap_i\left(\Delta f_i/\varepsilon\right)\end{cases} \tag{10}$$

$C_N(d)$ is a noise data set allocated by the cloud server to each work node and added with the Laplace noise through differential privacy. Each work node in the data analyst executes related machine learning algorithms such as query, classification, calculation, statistics and the like on the noise data set, and trains a sub-model of each work node with the allocated data. Then the sub-models are submitted to the data center in the cloud server by the working nodes, and the next sub-model is trained, and finally all the sub-models of all the nodes are summarized by the data center to form a complete machine learning model to complete the machine learning task outsourced by the user.

IV. Safety Analysis

A. Data Integrity Analysis

The adversary A_{tr} attacks the ciphertext and training nodes distributed to the training nodes after the cloud server adds noise. Fig. 3 shows the security model of the ACA-DMLP scheme.

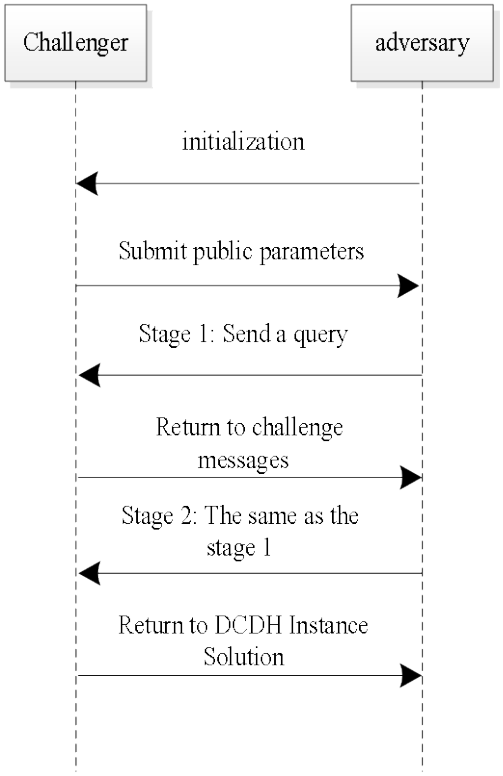

Fig. 3. Security Model of ACA-DMLP Scheme.

Setup: B_{tr} submits a public parameter $(q,G,g,H_1,H_2,H_3,H_4,e_0,e_1)$ to A_{tr}. B_{tr} uses the list $(L_{H_1},L_{H_2},L_{H_3})$ to simulate the random oracle model of H_1,H_2,H_3 respectively, and guarantees their consistency. B_{tr} prepares a table L_k for the public and private keys.

Lemma 2: examine that communication between A_{tr} and the algorithm B_{tr} of the scheme in this paper according to the IND-PRE-CCA game.

Phase 1: the adversary A_{tr} issues a series of queries, and the algorithm B_{tr} responds to these queries according to the scheme algorithm.

Challenge: The adversary A_{tr} challenges B_{tr} to request a ciphertext message $f(d)$ from the cloud server. B_{tr} responds to a series of queries from A_{tr} with a ciphertext $C_N(d)$ that contains x_n and $\frac{\varepsilon}{2\Delta f_n}\exp(\frac{-\varepsilon|x_n|}{\Delta f_n})$. Due to the unsolvability of the indeterminate system of equations, the adversary cannot infer the initial ciphertext $f(d)$ from $C_N(d)$.

Phase 2: A_{tr} continues to issue attack queries as in Phase 1, and algorithm B_{tr} continues to respond to adversary L's queries in the challenging manner described above.

Guess: B_{tr} returns a solution to the DCDH instance. In the random oracle model, the scheme is secure under the IND-PRE-CCA property. If an adversary A_{tr} corrupts CS or DA to obtain the outsourced data, A_{tr} cannot get the plaintext due to the IND-PRE-CCA nature of the scheme. In addition, if A_{tr} gains access to some data, the scheme achieves $\varepsilon - DP$ due to the Laplace mechanism adding noise and the unsolvability of the algorithm equations. Therefore, the scheme is secure under the random $\varepsilon - DP$ model.

B. Collusion-Resistant Analysis

Due to the semi-honesty of the training nodes in the data analyst, suppose that there are $\delta(1 \le \delta \le N)$ training nodes in the data analyst who collude with their training submodel R_i to steal $R_e = (R_1, R_2, \cdots, R_\delta)$. Due to the strong background assumption of differential privacy, at least one training node does not participate in the collusion attack, and the adversary solves the logarithmic equation $C(d) = f(d) + Lap(\Delta f / \varepsilon)$. Because of its difficulty, the adversary cannot solve $1 \le i \le n$ and Δf_i. Assume that the adversary guesses Δf_i after several repetitions with a very low probability. According to equation (7), the adversary conspires to construct a system of equations with δ equations and $\delta + 1$ unknowns:

$$\begin{cases} C_1(d) = x_1 + \frac{\varepsilon}{2\Delta f_1}\exp(\frac{-\varepsilon|x_1|}{\Delta f_1}) \\ C_2(d) = x_2 + \frac{\varepsilon}{2\Delta f_2}\exp(\frac{-\varepsilon|x_2|}{\Delta f_2}) \\ \vdots \\ C_\delta(d) = x_\delta + \frac{\varepsilon}{2\Delta f_\delta}\exp(\frac{-\varepsilon|x_\delta|}{\Delta f_\delta}) \end{cases} \tag{11}$$

Since Equation (9) is an indeterminate equation system with infinitely many solutions, the adversary cannot solve all equation unknowns through the limited unknowns. The adversary cannot conspire to calculate all initial ciphertexts $f(d)$ and all sub-models R_i through Equation (7).

V. Experimental Results and Analysis

A. Experimental Setup

In the scheme, the data set is stored in the cloud server, so the user's local storage cost is small. The main analysis is not done in the scheme. The cost of the scheme depends on the time overhead, including encoding time, communication time, and computation time. To evaluate the time cost of the scheme in this paper, the test platform is shown in Table I.

The experiment examines the running time of the relevant scheme on the MNIST data set of size $(m,d) = (12396,1568)$, where m is the sample size of the training data set and d is the test sample size. The distributed computing is simulated by the platform without considering the network delay. The time cost and accuracy of the proposed scheme are compared with the schemes in [23], [24] and [25].

B. Efficiency Analysis

In the experiment, the total time cost of the scheme in this paper and the schemes in [23], [24] and [25] is simulated and analyzed by setting the number of training nodes $N = (5,10,15,20,25,30,35,40)$. Fig. 4 compares the data calculation efficiency of the four schemes under different numbers of training nodes.

TABLE I. Test Platform Configuration

Type	Settings
CPU	Intel(R) Core(TM) i7-10700 4.8GHZ
RAM	32G DDR4
Hard disk	1T SSD
Operating system	Windows10
Data set	MNIST
Simulation platform	Python

In the experiment, the training time of different schemes is measured while the number of nodes is gradually increased. The following conclusions can be drawn: when $N = 5$, due to the small number of nodes and the small degree of parallelism between hosts, all schemes have almost the same performance; With the increase in the number of nodes, the difference in the time spent by different schemes to process the same amount of data sets is increasing, and the performance comparison between schemes is more obvious. Due to the distributed nature of the proposed scheme, the total amount of data sets are evenly distributed in each node. The more the number of training nodes is, the less time it takes to process the same task, and the shorter the total running time is. Thus, the time to process data sets decreases with the increase of the number of nodes. Compared with the scheme in the reference, the time cost of the scheme in this paper is smaller. Therefore, through the comparison of Fig. 4, it can be seen intuitively that the scheme adopted in this paper has obvious computational advantages. In the MPC scheme of [25], no matter how many hosts there are, each host repeatedly computes the entire data set to meet the needs of processing all tasks, so the computing time tends to increase. Through analysis, when the number of hosts is $N = 40$, it is found that the scheme in this paper has a significant improvement in efficiency compared with the schemes in [23], [24] and [25]. Fig. 5 shows the comparison of communication time (Comm), encoding time (Enco), computation time (Comp) and total time (Total) between the proposed scheme and the reference scheme when $N = 40$.

It can be seen from the images that the running time of each part of this scheme has been significantly improved compared with the reference schemes [23], [24] and [25]. The main reason is that the user encrypts the private data through the homomorphism in the scheme, which simplifies the algorithm and reduces the complexity. In the reference scheme [25], the data set size of each host is the same as the original data set, while the data set of each host in the present scheme is only 1/40 of the original data set. This is because the distributed machine learning provides a large parallelization gain for the scheme, while the reference scheme has a large computational overhead.

C. Accuracy Analysis

MNIST data set is set in the experiment (12396 samples are used in the training set, and 1568 samples are used in the test set). Since [24] does not discuss the problem of training accuracy, the accuracy of this scheme is compared with that of the schemes in [23] and [25]. When the number of hosts is $N = 40$, the accuracy of this scheme is compared with that of the schemes in [23] and [25] under different iterations. Fig. 6 illustrates the experimental comparison results of the three schemes.

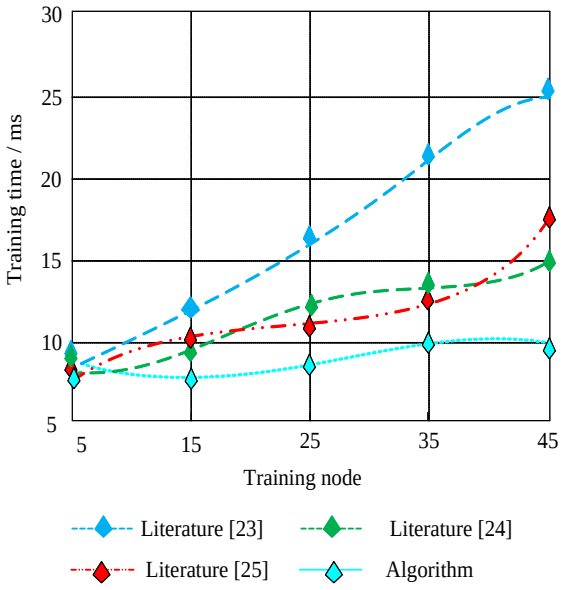

Fig. 4. Comparison of Running Time of Different Schemes.

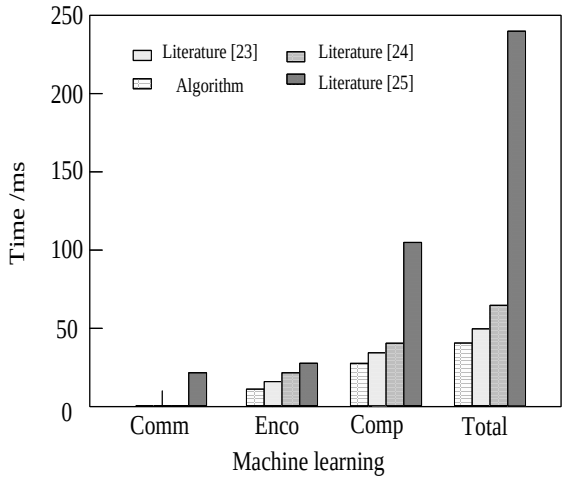

Fig. 5. Time Comparison of each Link of Different Schemes.

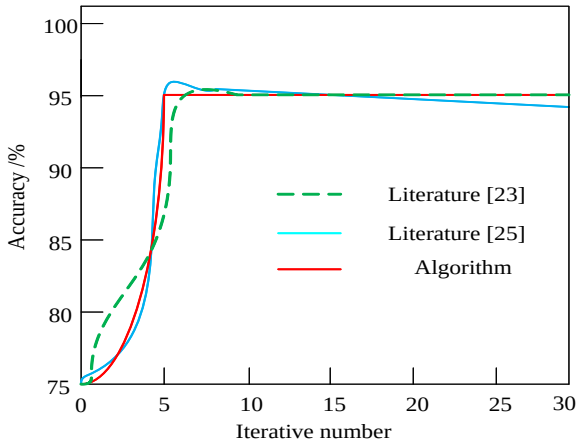

Fig. 6. Comparison of Accuracy of the Three Schemes.

It can be seen from Fig. 6 that during model training, compared with the reference schemes [23] and [25], this scheme has the same number of iterations, and the accuracy of model training of the two schemes is the same. When the number of iterations is five, there is a slight difference in the training accuracy of the scheme in this paper. The training accuracy of the scheme in this paper is slightly different from that of the comparison scheme, but the difference is kept within 2%. With the increase in the number of iterations, the accuracy between the schemes is getting closer and closer. The difference between the training accuracy of the scheme in this paper and that of the scheme in the reference is only 0.2%. This result shows that the scheme used in this paper almost guarantees the same accuracy as the reference scheme when the data set is unchanged and the number of iterations of the three schemes is the same. This scheme does not reduce the accuracy of model training because of the improvement of computational efficiency and data security.

D. Discussion

By comparing the functions of each scheme, it can be seen that [23] uses distributed architecture to improve the efficiency of data training, and uses homomorphic encryption algorithm and differential privacy to ensure the security of user privacy, and supports the public verification of ciphertext by each entity in the model. Appropriate measures are not taken to defend against the collusion between the adversary and the training nodes. The author in [24] uses a distributed model to speed up data analysis and improve the efficiency of training, but does not take security algorithms to protect the security of user data. The scheme in [25] can resist the collusion attack of the adversary and the training nodes in the distributed scheme, but it does not support ciphertext operation and public verification, and does not use differential privacy technology to protect the security of user privacy data. This scheme uses distributed structure to shorten the time of data analysis and improve the efficiency of machine learning, and uses homomorphic encryption algorithm to support the training platform to train on the ciphertext, uses differential privacy to strictly prevent the user's private data from being leaked in the process of transmission and training, and prevents the collusion theft of adversaries and distributed training nodes. At the same time, each role in the model is supported to download and publicly verify the data in the ciphertext domain at any time to ensure the integrity of user privacy data. Through the above analysis and comparison, the scheme in this paper has high feasibility, and strictly guarantees the integrity of user data, and improves the training efficiency of machine learning.

VI. Conclusion

In this paper, a collusion-resistant distributed machine learning privacy- preserving (ACA-DMLP) scheme is proposed.

1) The scheme adopts the architecture of distributed machine learning and improves the efficiency of data training through the cluster parallel systems.

2) A differential privacy encryption algorithm and a Laplace mechanism are used to add noise disturbance to the ciphertext data in the cloud server to ensure data security in the ciphertext domain.

3) The feasibility and high efficiency of the scheme are objectively proved by simulation experiments on relevant platforms. The scheme in this paper improves the security and analysis efficiency of user private data in machine learning and can prevent adversaries from colluding with semi-honest working nodes within data analysts to steal data. The scheme in this paper only considers the safety of the model before training and the sub-model of the training node in machine learning but does not make an effective scheme analysis and demonstration of the data processing, model combination, and the safety of the overall model after machine learning. In future work, the data after training will be processed safely and the outsourcing agencies will be guaranteed to submit the machine learning results to users safely.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability Statement

The data that support the findings of this study are available upon reasonable request.

References

[1] L. Cao, Y. Kang, and Q. Wu, “Searchable encryption cloud storage with dynamic data update to support efficient policy hiding,” China Communications, 2020, 17(6): 153-163.

[2] K. Liu, J. Peng, and J. Wang, “Scalable and adaptive data Replica placement for geo-distributed cloud storages,” IEEE Transactions on Parallel and Distributed Systems, 2020, 31(99): 1575-1587.

[3] Samankumara, Hettige, and Eshani, “Usage of cloud storage facilities by medical students in a low-middle income country, Sri Lanka: a cross sectional study,” BMC Medical Informatics and Decision Making, 2020, 20(1): 1-8.

[4] S. Jing, A. Ebadi, and D. Mavaluru, “A method for virtual machine migration in cloud computing using a collective behavior-based metaheuristics algorithm,” Concurrency and Computation: Practice and Experience, 2020, 32(2): 1-13.

[5] I. Filip, A. Potoac, and R. Stochioiu, “Data capsule: representation of heterogeneous data in cloud-edge computing,” IEEE Access, 2019, 7: 49558-49567.

[6] I. Mavridis, and H. Karatza, “Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing,” Future Generation Computer Systems, 2019, 94(5): 674-696.

[7] P. Shakeel, S. Baskar, and H. Fouad, “Internet of things forensic data analysis using machine learning to identify roots of data scavenging,” Future Generation Computer Systems, 2021, 115: 756-768.

[8] A. Jeavons, “What is artificial intelligence,” Research World, 2017, 2017(65): 75-75.

[9] E. Tapoglou, E. Varouchakis, and I. Trichakis, “Hydraulic head uncertainty estimations of a complex artificial intelligence model using multiple methodologies,” Journal of Hydroinformatics, 2020, 22(1): 205-218.5.

[10] K. Ishii, “Comparative legal study on privacy and personal data protection for robots equipped with artificial intelligence: looking at functional and technological aspects,” AI & society, 2019, 34(3): 509-533.

[11] K. Lin, J. Lu, and C. Chen, "Unsupervised deep Learning of compact binary descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019: 1-1.

[12] W. Li, B. Jiang, and W. Zhao, "Obstetric imaging diagnostic platform based on cloud computing technology under the background of smart medical big data and deep learning," IEEE Access, 2020, (99): 1-1.

[13] W. Liu, J. Guo, and F. Yao, "Adaptive protocol generation for group collaborative in smart medical waste transportation," Future Generation Computer Systems, 2020, (110): 167-180.2.

[14] U. Boryczka, and M. Bachanowski. "Using differential evolution in order to create a personalized list of recommended items," Procedia Computer Science, 2020, (176): 1940-1949.

[15] Helmi, Abrougui, Habib, "Autopilot design for an autonomous sailboat based on sliding mode control," Automatic Control and Computer Sciences, 2019, 53(5): 393-407.

[16] D. Pal, and C. Arpnikanondt, "Analyzing the adoption and diffusion of voice -enabled smart-home systems: empirical evidence from Thailand," Universal Access in the Information Society, 2020: 1-19.

[17] U. Udhayakumar, and G. Murugaboopathi, "To improve user key security and cloud user region-based resource scheduler using rail fence region-based load balancing algorithm," Journal of Ambient Intelligence and Humanized Computing, 2020(6): 1-8.

[18] P. Puteaux, M. Vialle, W. Puech, "Homomorphic encryption-based LSB substitution for high-capacity data hiding in the encrypted domain," IEEE Access, 2020, 8: 108655-108663.

[19] D. Alistarh, D. Grubic, and J. Li, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," Advances in Neural Information Processing Systems 30, Curran Associates, 2017, (2017): 1709–1720.

[20] C. Gao, Q. Cheng, and X. Li, "Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network," Cluster Computing, 2018, (2018): 1655-1663.

[21] P. Li, T. Li, and Y. Heng, "Privacy-preserving machine learning with multiple data providers," Future Generation Computer Systems. 2018, (87): 341-350.

[22] Z. Wei, J. Li, and X. Wang, "A lightweight privacy-preserving protocol for VANETs based on secure outsourcing computing," IEEE Access, 2019, 7 (99): 62785-62793.

[23] H. Alzubair, and H. Rafik, "An efficient outsourced privacy preserving ma-chine learning scheme with public verifiability," IEEE Access. 2019, (7): 146322-146330.

[24] C. Fang, Y. Guo, and N. Wang, "Highly efficient federated learning with strong privacy preservation in cloud computing," Computers & Security, 2020, 96: 101889.

[25] Y. Hana, L. Michael, A. Said, "Systematic review on fully homomorphic encryption scheme and its application," Recent Advances in Intelligent Systems and Smart Applications, 2018, (2020): 537-551.