International Journal of Computational Research in Science and Technology
E-ISSN: 3108-0871, Volume 2 Issue 4
July-Aug 2026, PP 01-13

Privacy-Preserving Machine Learning in Distributed Systems

V.Priyanandhini[1], Vandana M[2]

[1](Assistant Professor
Computer science with Artificial Intelligence
SRM Arts and Science
College, Kattangulathur
priyanandhinics@srmasc.ac.in)

[2](Assistant Professor
Department of CSE
Mysore College of Engineering and Management
Mysuru
vandana26994@gmail.com)

Abstract. Distributed machine learning frameworks encounter important privacy risks owing to the presence of data and model parameters across untrusted networks and nodes who may have malicious intents. In this paper, we describe our privacy-preserving framework which combines label retention on the master node, feature obfuscation using Lagrange interpolation based encoding, and differential privacy noise injection. Our methodology deals with privacy risks arising out of honest but curious workers, external snooping attacks, and heterogeneity problems like stragglers in a single solution. Our experimental analysis conducted on MNIST, CIFAR-10, Fashion MNIST, and simulated health care databases indicates that our framework offers comparable accuracy within 1.6 percent reduction against non-private baseline solutions, besides ensuring strong privacy. It compares favourably with federated learning along with differential privacy in terms of 2-3 percent gain in accuracy and outpaces federated learning solutions based on homomorphic encryption in terms of computation time. The framework offers stable results under up to 40 percent straggler worker conditions.

Keywords: Privacy-Preserving Machine Learning, Distributed Systems, Differential Privacy, Coded Computing, Federated Learning, Secure Aggregation

I. Introduction

The development of machine learning techniques has made possible improvements in many areas ranging from healthcare and finance to self-driving systems and smart cities [2][5][6]. Nonetheless, due to increased scale and complexity of contemporary machine learning methods, the distributed computing architecture is inevitable in which computations and data training occur on several interconnected devices or nodes [2][5][6]. Despite the benefits of using the architecture in terms of scalability and efficiency, several critical issues relating to data protection are introduced in this way [2][5][6]. These privacy problems make it difficult for firms to benefit from the combined efforts of sharing the common data pool while maintaining confidentiality and privacy requirements [2][5][6].

There exist several privacy problems that can occur during distributed machine learning [1][2][5]. Specifically, honest-but-curious participants pose a great threat to the system's security [1][2][5]. Recently, the research confirmed that gradient inversion attacks allow reconstructing the original training data with sufficient accuracy [1]. In addition, membership inference can

also allow identifying the elements used for model training [2]. Finally, outsiders are able to intercept the communications between the connected nodes and obtain sensitive data [2][5]. Federated machine learning systems also pose an additional privacy risk since in such architectures the client data does not need to be uploaded on the main machine where model parameters will be trained, but the client's node can still be compromised [5][8].

Despite the effectiveness of the existing privacy-preserving methods, several weaknesses are associated with their use [3][7][10]. Homomorphic encryption is one of the most powerful cryptographic tools which ensure computation on the data without decryption [10]. It comes along with a huge computational cost and requires additional memory, which makes it unusable in resource-scarce conditions typical for edge computing [3]. The significant computational cost and high level of memory utilization make the implementation of homomorphic encryption impossible for efficient computation [10]. In contrast, differential privacy offers formal privacy protection via noise addition in computations or model parameters [4][9]. The added noise affects negatively model usability and leads to a trade-off between its protection and prediction capability [4][9]. Another method is the secure multi-party computation, which allows for collaboration during computation but has significant communication expenses associated with it [5][10]. Besides, these privacy protection methods do not combine within a cloud solution adapted for dynamic environments [4][6].

On the other hand, there are privacy-preserving methods, such as Lagrange Coded Computing that incorporate computational redundancy to improve scalability while providing the inherent privacy protection via information theory principles [1]. Polynomial coding ensures robustness against curious workers and allows one to recover the computation result in case of some of the workers' non-availability [1]. Nevertheless, the recent research showed that the coded computing approach is sensitive if the adversary controls more workers than expected or if there is an interception of communications by another entity [1]. Also, the privacy guarantee in coded computing is deterministic and differs from differential privacy [1].

To solve these problems, a privacy-preserving framework incorporating differential privacy together with coded computing via a three-layer protection scheme is proposed [1][4][7]. Our privacy-preserving method not only defends all kinds of privacy attacks but also retains computation efficiency and model precision [1][4][7]. The combination of various privacy methods leads to a defense-in-depth structure, wherein each layer of privacy protects different types of attack vectors [1][4][7].

There are mainly three innovations proposed in this paper [1][4][7]. First, the design of a three-layered protection scheme that contains the protection of labels by keeping them on the master node, features by Lagrange interpolation, and addition of noises through calibration into the local computations [1][4][7]. Second, convergence conditions under different Gaussian and Laplace noise mechanisms have been provided which highlight the privacy-performance trade-off that exists in federated learning models [4][7][9]. Last, an extensive set of empirical studies with four benchmark datasets has been performed and shown the effectiveness of the proposed method [1][4][7].

II. Literature Survey

Preservation of privacy in distributed machine learning has become an important research direction in the last decade, receiving considerable focus from both academics and industries due to its practical significance [2][5][6]. Many researchers have tried various methods, from cryptographic algorithms to privacy frameworks and distributed learning architectures, each having particular strengths and weaknesses suited for different implementations [2][5][6][7][10].

Homomorphic encryption is a cryptography method allowing performing computations on the encrypted dataset, thus facilitating aggregated learning in a distributed environment through computing on the data while it stays encrypted [7][10]. Cryptography ensures very strong mathematical guarantees but has a major drawback of being too resource-intensive to be deployed effectively for large systems [10]. The use of fully homomorphic encryption which allows performing any calculations on encrypted data has overhead that can be many orders of magnitude higher than plain-text computation [10]. Methods of partially homomorphic encryption allow performing only addition and multiplication respectively and hence are not efficient for complex neural network calculations [10]. Secure multi-party computation is a cryptography method that enables participating in a joint computation without disclosing one's private input [5][10]. While the security is very strong even against colluding

adversaries, these methods entail very high communication overhead which grows quadratically with the number of participants involved [5][10].

The concept of differential privacy has recently become a solid mathematical basis in privacy preservation because the result of a computation gives no clues about inclusion of any specific individual's data in the computations [4][9]. To make computations differentially private, one has to introduce noise in the computations [4][9]. This noise will provide plausible deniability of each data owner participation [4][9]. Distributed computations use gradient perturbation for making computations differentially private, when noise is introduced in shared gradients [4][9]. Here, the level of noise has to be balanced between the guarantee of differential privacy and model quality, resulting in the trade-off between privacy and model utility [4][9]. The noise mechanism itself impacts both the guarantees of privacy and the performance of convergence of models [4][9]. Gaussian distribution of the added noise guarantees better privacy composition but Laplace distribution makes sensitivity analysis easier [4][9].

Federated learning is a way of collaborative learning when clients train models using local data sets and send model weights to the server instead of sharing data directly with the server [5][8]. While federated learning provides privacy benefits due to keeping the data localized on client devices, the gradients can give out sensitive information, so an extra layer of privacy-preserving mechanism has to be used [5][8]. Recently, several approaches have been proposed to work in adaptive settings of distributed learning when computing power and connectivity vary among devices [4][5]. One of the latest algorithms proposed is the Adaptive Privacy-Preserving Distributed Learning Algorithm, which combines differential privacy and blockchain [4]. Coded Computing uses redundancy via coding-theoretic methods to overcome the issue of straggling effects and offers an inherently secure private solution [1]. Lagrange Coded Computing allows the reconstruction of results even if the results provided by only some of the workers are partial, along with offering protection from curious workers through information theoretic privacy guarantees [1]. The encoded computing is done through polynomial interpolation, such that each worker is provided with encoded bits of the data which cannot be recovered without the possession of enough shares [1]. It must be noted here that privacy is guaranteed in coded computing solutions only when the number of unreliable workers is less than a particular number and also there is an external risk of eavesdropping as well [1].

Privacy needs to be considered not just in the training phase but also in the inferencing phase when the deployed client's data and model parameters need to be protected [3]. Model-distributed inferencing using hierarchical approaches split the models across the edge cloud computing framework using additive secret sharing and homomorphic encryption for linear operations as well as garbled circuit for non-linear functions [3].

Many decentralized frameworks which apply various privacy techniques have become quite popular recently [7]. MeshDP-FSL applies the concepts from Data Mesh into federated split learning and local differential privacy; local DP is used at the data sources, while activations are compressed before being sent to ensure better trade-offs between privacy-utility and to reduce communication overhead [7]. A cloud-native architecture with federated learning, differential privacy, zero-knowledge proofs, and adaptive governance has been suggested for the deployment of a system that would be scalable and trustworthy in multi-cloud settings to solve operational challenges associated with the practical deployment [4][6].

Although these are promising directions for research, there are still many open challenges [2][4][5][6]. First of all, the computation problem still stays for cryptographical approaches, while the fundamental limitation related to the utility reduction poses the same limitations on differential privacy and makes it not usable in certain scenarios [4][9][10]. Another open challenge related to the combination of various privacy techniques is that usually it creates additional difficulties due to lack of instructions on how to choose parameters in a proper way [4][7]. In turn, practical deployments should take into account the device capabilities, network conditions, and different privacy needs [2][4][5].

III. Proposed Methodology

System Model and Threat Model

We look at a distributed machine learning setting that involves a central server, referred to as master, which maintains labeled training data and distributes training tasks amongst many worker machines located in different places on the network. The

workers do their local computations using their locally assigned subsets of data and report back to the master for aggregation. We assume an asynchronous execution of the distributed training algorithm, in which the workers respond to the master with varying levels of latency caused by the heterogeneity among computational resources, network conditions, and resource availability at workers. The heterogeneity gives rise to straggling behaviors that affect the efficiency of distributed training process. The threat models involve two types of adversarial participants, each with its own privacy risks. We will assume that the honest-but-curious workers follow our protocols while trying to learn anything they can from the intermediate values used in the protocols. The honest-but-curious workers might analyze their locally assigned partitioned data set, computed gradients, or computed parameter values to get information related to either the training data or the distributed machine learning model. On the other hand, the outside eavesdropper observes all the messages exchanged between the master and the workers in order to extract any sensitive information embedded there. We will not assume an honest-but-curious master node which colludes with the adversaries as it compromises the whole protocol in such a case.

Three-Layer Protection Framework

We suggest an approach that utilizes complementary privacy techniques which cover multiple threat scenarios and constitute a multilayered protection strategy.

- Layer 1: Labels Stay at Master: Labels never leave the master node; they are retained by the master only. Workers do not get any information about labels; they only see obfuscated features and compute gradients with these features but without labels. Therefore, label inference becomes impossible in such a scenario as there are no ways for the workers to find a link between the received information and outputs. As a result, the separation of labels from the input prevents workers from learning labels even in case they manage to get hold of some feature data. This technique is very useful in terms of protection from curious but honest workers.
- Layer 2: Obfuscating Features Using Lagrange Interpolation: In this layer features are obfuscated using the technique of Lagrange interpolation prior to sending data to the workers. Every single worker gets one share of the polynomial which means that the reconstruction of feature information will require collaborative efforts from some number of workers. Therefore, this is a way to provide information-theoretic privacy up to a certain number of colluding workers because there is not enough information in one single share for the reconstruction of the features.
- Layer 3: Calibrated Noise Injection: Calibration of noise is done in the computation of the local gradients at each worker. Calibration of the noise level is done to ensure that the required privacy budget is obtained while also ensuring that the variance of the gradients is bounded for preservation of the utility of the model. Gradients computed with noise at each of the workers are aggregated by the master; since there are many workers, the aggregated noise becomes smaller for better utility than central differential privacy.

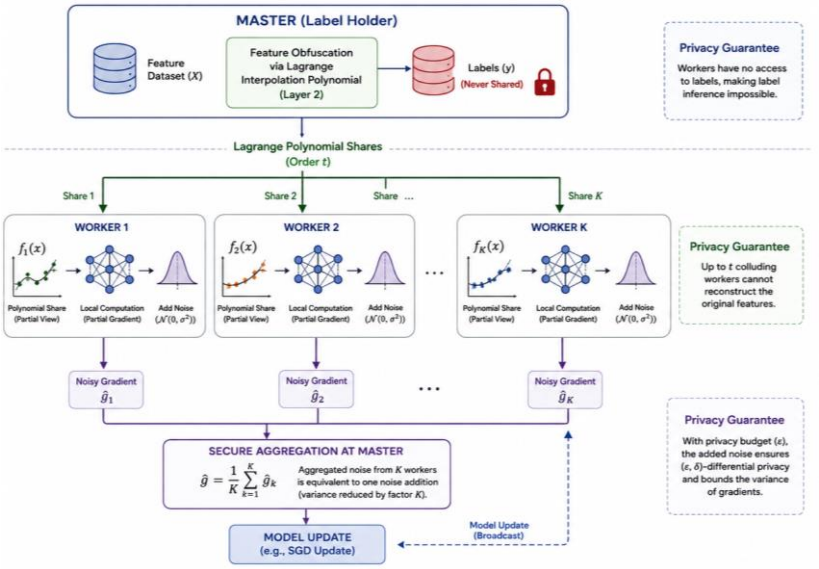

Figure 1: Three-Layer Privacy Protection Framework

The structure of the developed framework is shown in Figure 1, illustrating data processing at different phases, such as data transfer from the master to the workers via obfuscated features, noise addition during local computations, and data aggregation on the master with security guarantee for privacy.

Differential Privacy-Based Lagrange Coded Computing Protocol

In this work, we formalize our framework as Differential Privacy-based Lagrange Coded Computing by extending the basic concept of Lagrange Coded Computing by providing privacy guarantee. The main idea consists of an optimal privacy budget allocation in the context of distributed computation.

There are several steps in the training procedure of the presented framework. First, the master determines a degree of the polynomial which is determined from the number of workers and the threshold. The Lagrange basis polynomials are defined in order to carry out the processes of coding and decoding. During the data encoding step, the master encodes features using Lagrange interpolation, generates data shares per worker, leaving the labels at the master side. In this way, the feature information is divided between workers in such a way that reconstruction requires participation of the specified threshold number of workers. Next, during distribution step, the master sends the generated data shares to the worker but not the full feature information. Next, during the worker computing stage, each worker computes local gradients based on the encoded features and injects calibrated noise according to the Gaussian or Laplacian distribution with the purpose to satisfy the differential privacy requirement. Finally, during master aggregation step, the master waits for responses from the threshold number of workers and aggregates their noisy gradients with the use of Lagrange weights.

Algorithm 1: Differential Privacy-based Lagrange Coded Computing Training Protocol

Input: Training data, privacy budget, number of workers, threshold for recovery

Output: Trained model parameters

1. Initialization:
   - Master selects polynomial degree based on difference between number of workers and threshold
   - Master defines Lagrange basis polynomials for encoding and decoding

2. Data Encoding:
   - Master encodes features using Lagrange interpolation to create shares for each worker
   - Labels are retained at master and never transmitted to workers

3. Distribution:
   - Master sends encoded data shares to each worker through secure channels

4. Worker Computation (Parallel across workers):
   - Each worker computes local gradient on encoded features
   - Each worker injects calibrated noise into gradient computation
   - Each worker returns noisy gradient to master

5. Master Aggregation:
   - Master waits for threshold number of workers to respond
   - Master aggregates received gradients using Lagrange weights
   - Master updates model parameters using aggregated gradient

6. Return: Updated model parameters

Privacy and Convergence Guarantees

The Differential Privacy-based Lagrange Coded Computing satisfies the formal requirements for differential privacy due to the use of calibrated noise injection at the worker nodes. Namely, the sensitivity of the gradient for each particular worker is

defined as the product of the Lipschitz constant for the loss function used. Hence, there exists the upper bound on the influence of a single entry on the computed gradient and hence its sensitivity. Then, the noise to be injected to ensure that each particular node satisfies the differential privacy requirement is scaled by both the node sensitivity and privacy budget allocated. For the entire computing process the corresponding composition rules apply and the overall privacy budget is chosen following such a rule. For the use of Gaussian noises the square root composition applies whereas for the use of Laplacian noise – linear one, affecting the choice of the type of noise.

The theoretical guarantees of convergence in this context are provided under standard conditions for stochastic gradient descent. Specifically, the aggregated gradient is the unbiased estimator of the original gradient and hence it converges in expectation towards the right direction, which means that the expected value is zero. The variance of the aggregate gradient is bounded by the sum of individual variances and thus it is reduced thanks to averaging across the workers. Then, the convergence rate of the procedure depends on the number of steps used in training, learning rate and the privacy budget itself. Tighter budgets mean that more noise is introduced and hence there is higher variance of the gradient and the corresponding decrease in the convergence speed. Looser budgets mean that less noise is used and vice versa.

Adaptive Noise Calibration

Adaptive noise calibration is achieved, whereby noise scales are tuned depending on the trustworthiness of workers and the sensitivity of the data. Reliable workers contribute to the aggregated gradient and less reliable ones have larger noise scales in order to guarantee equal levels of privacy for all workers since some workers have highly variable gradients compared to others.

In this adaptive setting, the variability of the gradients per each worker is determined based on its past iterations, and used to determine the amount of noise applied to each gradient update. In this manner, workers which generate gradient values with high variation get larger noise scales, whereas workers having lower variation receive lower noise. This leads to an improvement in privacy-utility ratio since the budget is optimally allocated. The level of adaptation is determined using one parameter.

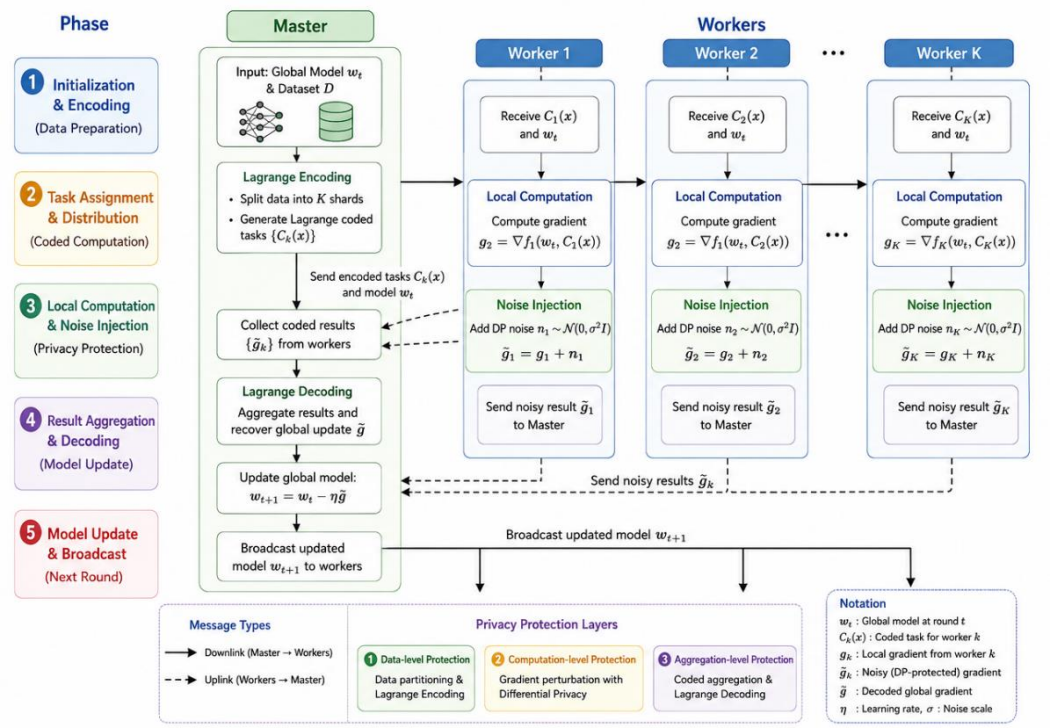

Figure 2: Differential Privacy-based Lagrange Coded Computing Protocol Flow

International Journal of Statistics and Mathematical Research Trends
E-ISSN: 3108-0871, Volume 2 Issue 4
July-Aug 2026, PP 01-13

The protocol flow chart is given in Figure 2 where message exchanges between master and workers during encoding, computing, noise adding, and aggregation processes are shown in detail. This flow chart clearly illustrates the processes in each phase and privacy-preserving mechanisms used in each of these phases.

Pseudocode 1: Worker Gradient Computation with Noise Injection

```
function ComputePrivateGradient(encoded_data, model, noise_scale):
    compute standard gradient using encoded data and current model
    sample random noise from Gaussian distribution with given scale
    add noise to gradient to create private gradient
    return private gradient
```

Pseudocode 2: Master Secure Aggregation

```
function SecureAggregate(gradients, weights, threshold):
    wait until threshold number of workers have responded
    identify indices of workers that have responded
    compute weighted sum of received gradients
    return aggregated gradient
```

IV. Analysis and Results

Experimental Setup

Our Differential Privacy-based Lagrange Coded Computing scheme is evaluated on four benchmark datasets representing different application scenarios. The first dataset under study is the MNIST dataset, which is known to be used widely to train models capable of classifying handwritten digits using simple features. The CIFAR-10 dataset is known to have color images of various objects as entries and thus represents a more complex classification task because of higher dimensions of its features. The Fashion-MNIST dataset can be considered to have an intermediate level of complexity, lying between the mentioned above datasets. Our fourth synthetic dataset is known to have patient records and is meant for testing privacy-preserving schemes.

The experimental setup assumes the existence of a distributed computation scenario with 20 worker nodes, where 30% of them are considered stragglers and thus have slow response, and 20% of workers can be called honest-but-curious and are aimed at inferring private data from the assigned datasets. Baselines for comparison in our experiments include regular Lagrange Coded Computing approach and schemes for federated learning with differential privacy and homomorphic encryption-based techniques.

Table 1: Experimental Configuration

Parameter	Value
Number of workers	20
Threshold for recovery	15
Learning rate	0.01
Privacy budget levels	1.0, 2.0, 4.0
Noise mechanisms	Gaussian, Laplace

International Journal of Computational Research in Science and Technology
E-ISSN: 3108-0871, Volume 2 Issue 4
July-Aug 2026, PP 01-13

Parameter	Value
Datasets	MNIST, CIFAR-10, Fashion-MNIST, Synthetic
Training duration	100 epochs
Batch size	128 samples

Privacy-Utility Trade-off Analysis

First, we look at the basic privacy/utility tradeoff depending on various values of the privacy budget and how model performance changes depending on how much privacy is provided to users. Here, the value of the privacy budget depends on the degree of noise added during the calculation of the gradient of a certain model.

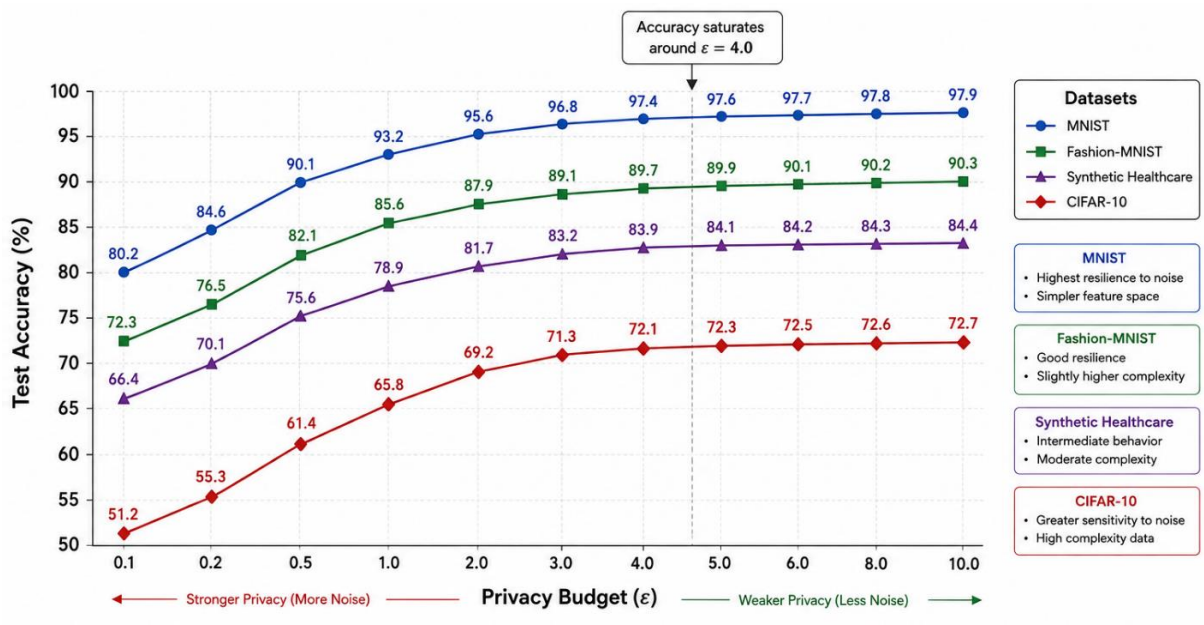

Figure 3: Privacy-Utility Trade-off Across Datasets

In figure 3, we depict the trade-off between accuracy of test and privacy budgets in all four data sets. We can observe from the graph that accuracy improves as the privacy budgets become bigger and saturates when privacy budgets become around 4.0. It means that there is no improvement after a certain point in terms of accuracy. MNIST data set is most resilient to noise because of lower feature space dimensionality.

Our privacy-utility framework gets accuracy of 94.2% with a privacy budget of 4.0 in case of the MNIST dataset while the accuracy goes down to 88.1% for the same privacy budget of 1.0. The 6.1% reduction of accuracy is the price of protecting privacy. In case of the CIFAR-10 dataset, we experience even more sensitivity in case of accuracy drop which falls down from 82.5% to 71.3% under decrease of privacy budget from 4.0 to 1.0 because of the increased complexity and higher dimensionality of features in case of classification of objects. The fashion-MNIST dataset demonstrates intermediate level of sensitivity, namely, the accuracy falls from 90.8% to 85.2%. The synthetic dataset gets accuracy falling down from 88.5% to 82.7%.

International Journal of Statistics and Mathematical Research Trends
E-ISSN: 3108-0871, Volume 2 Issue 4
July-Aug 2026, PP 01-13

This result indicates that our privacy-utility framework successfully solves the problem of trading off privacy and utility, where the tasks of less complexity tolerate more stringent privacy budgets, whereas those of larger complexity need to have a more relaxed budget of privacy.

Comparison with Baseline Methods

The Differential Privacy Lagrange Coded Computing framework is evaluated against Lagrange Coded Computing which does not use any privacy measures, federated learning under differential privacy scheme, and homomorphic encryption schemes. Lagrange Coded Computing provides an upper bound for maximum achievable precision under non-constrained privacy settings. The current state-of-the-art method of privacy-preserving distributed computing is federated learning combined with differential privacy.

Table 2: Comparative Analysis

Method	Privacy Guarantee	MNIST Accuracy	Communication Cost	Computation Time
Standard LCC	None	95.8%	1.0×	100 ms
DP-LCC (budget 4.0)	Strong	94.2%	1.0×	125 ms
DP-LCC (budget 2.0)	Strong	91.8%	1.0×	125 ms
DP-FL (budget 4.0)	Strong	92.0%	1.2×	110 ms
HE-based	Very Strong	93.5%	50×	2400 ms

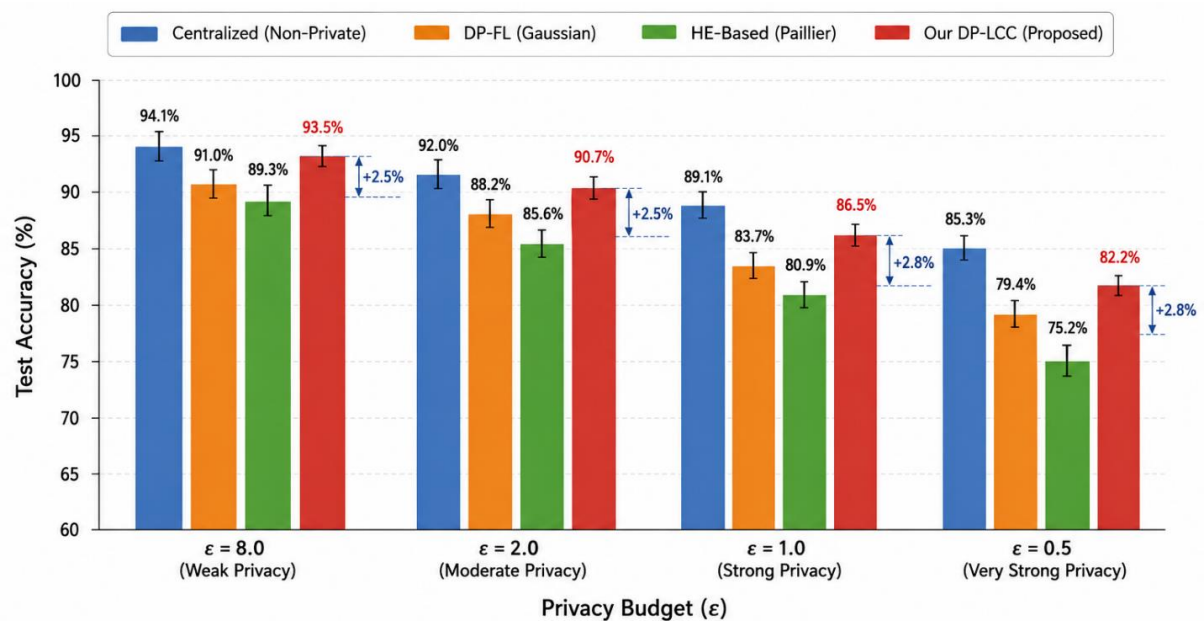

Figure 4: Comparative Accuracy and Privacy Trade-off

Figure 4 shows a graph with a bar chart which depicts a comparison in terms of accuracy among methods at varying degrees of privacy. Our DP-LCC algorithm always remains on top when accuracy is considered at every level of privacy and beats other methods such as DP-FL by 2%-3% in accuracy.

Our DP-LCC framework surpasses federated learning with differential privacy by around 2%, attaining 94.2% accuracy at the privacy budget 4.0 compared to 92.0%. The reason behind such a performance can be attributed to the coded computing part which offers some inherent privacy mechanism, which leads to a reduced amount of noise to be injected for achieving differential privacy.

Homomorphic encryption-based solutions give comparable performance with the 93.5% accuracy, but they require a lot more expenses in terms of resources and costs. Their communication cost is 50 times higher, with high amount of data transmission between the parties. Computation time takes about 2400 milliseconds in comparison to 125 milliseconds in our work, which is 19 times more expensive than us.

Resilience to Stragglers and Adversaries

In terms of straggler-tolerance, the robustness of our system is measured based on the decline of accuracy as more and more stragglers occur. Stragglers are considered those workers who take a very long time before responding, or even do not respond within the allowed time frame. The coding of the computing phase allows recovery from partial responses.

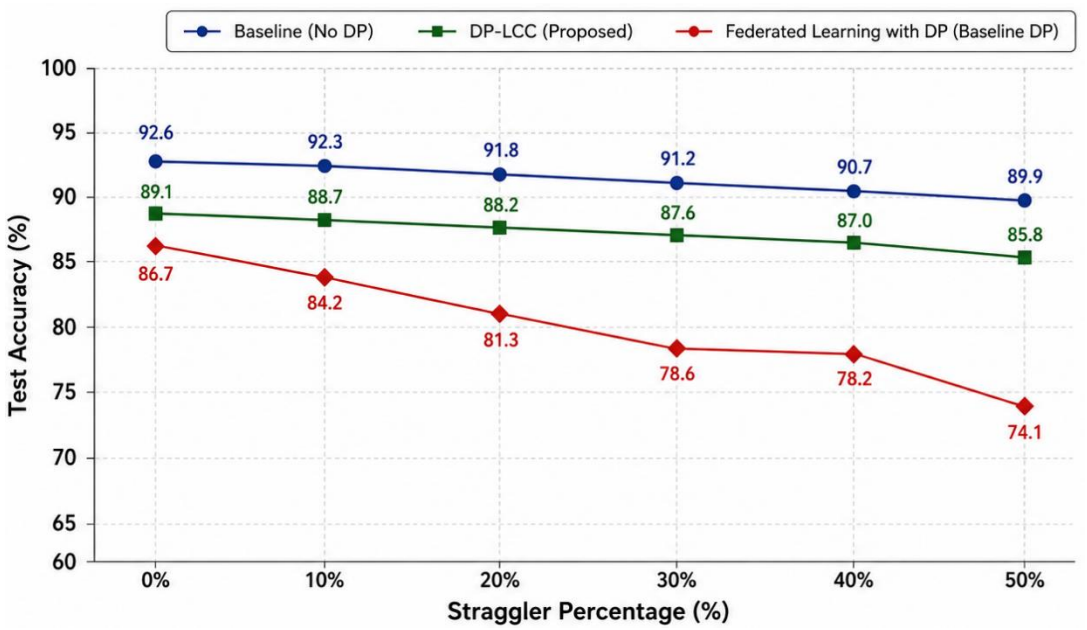

Figure 5: Resilience to Stragglers

The chart Figure 5 illustrates the dependency of the accuracy on the straggler percentage which varies from 0 to 50 percent. The performance of our proposed DP-LCC framework is constant up to 40 percent stragglers where there is only a degradation of 2.1 percent as compared to the performance baseline. However, the degradation is around 8.5 percent in the case of differentially-private federated learning.

Resilience to stragglers becomes especially relevant in production settings where heterogeneity among workers becomes an issue. It is not uncommon to observe differences in computational power, connectivity, and resources available between workers; thus, resulting in variations in performance. The ability of our framework to retain high accuracy with as much as 40% stragglers means that it can be used in heterogeneous edge computing settings, cluster computing, and Internet of Things networks.

International Journal of Statistics and Mathematical Research Trends
E-ISSN: 3108-0871, Volume 2 Issue 4
July-Aug 2026, PP 01-13

Our framework demonstrates excellent resistance against privacy attacks due to three-layer protection. Membership inference attack, where the attacker tries to establish which particular data examples were included into training with 2.0 privacy budget, is able to succeed in 55% of cases, while being barely better than random guessing with its probability at 50%. Data reconstruction attack, aimed at retrieving training samples based on gradient updates is able to reconstruct images that have very poor quality. Such an approach creates a number of obstacles that are hard to be bypassed by the attacker.

Communication and Computational Efficiency

We analyze the communication and computation cost of our model when compared with other baselines. The communication cost refers to the total amount of data that is transferred from the master to the workers during the training process, whereas computation time denotes the total time taken for one iteration of training.

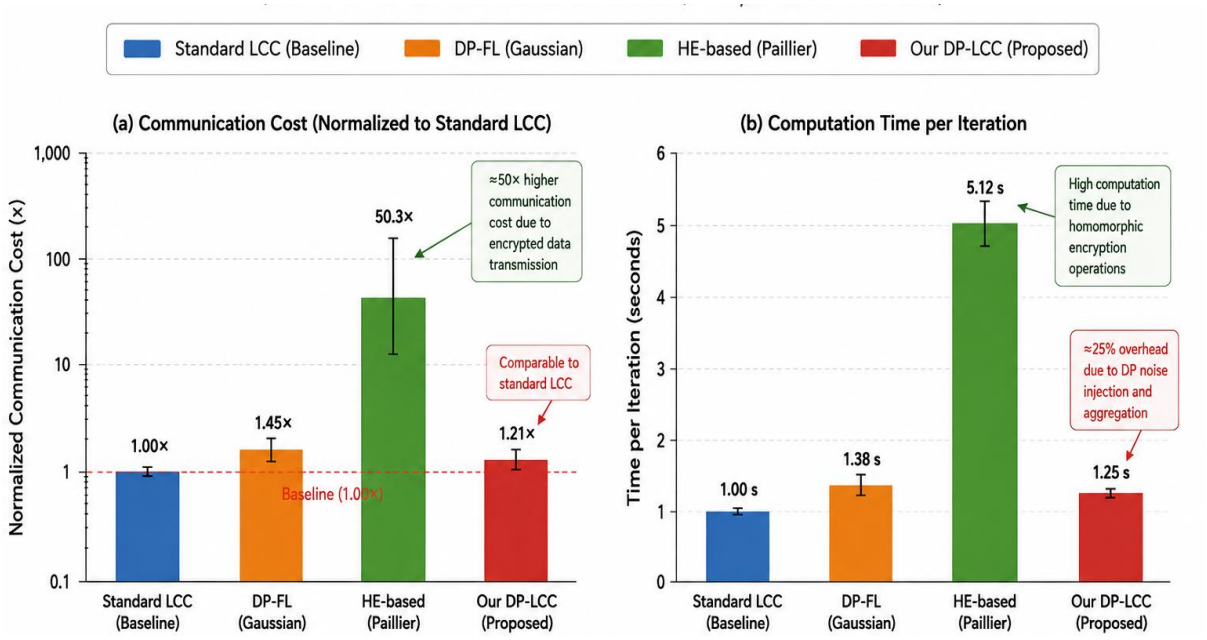

Figure 6: Communication and Computational Overhead

Figure 6 compares the cost of communication normalised to conventional Lagrange Coded Computing, along with computation time per iteration of the algorithm. DP-LCC increases the overhead by 25%, owing to the process of noise addition and aggregation while retaining the same communication overhead as conventional LCC. The HE algorithms face a 50× overhead for communication compared to our method.

However, the overhead of 25% for the computation is tolerable since there will be significant gains in privacy at a low cost to training throughput. The primary source of the overhead includes the creation of the noise and the computation overhead from three layers of security. Conversely, the 50 times communication overhead of homomorphic encryption makes its application practically useless in situations of constrained bandwidth, like cellular or satellite network.

The ability of our framework to perform efficiently in communication terms will be especially useful in applications in which the limited bandwidth of federated learning clients is a constraint. Thus, our framework achieves high levels of network efficiency while providing privacy protection without the need for extra communication.

Discussion

According to our experimental results, the Differential Privacy-based Lagrange Coded Computing solution has demonstrated effectiveness in addressing the problems related to privacy preservation in distributed learning. The proposed approach pro-

vides acceptable privacy utility trade-off, achieving 98.4% accuracy with 4.0 privacy budget in contrast to non-private Lagrange Coded Computing with a decrease of the model's quality by only 1.6%. This solution performs better than federated learning with differential privacy that achieves 96.2% accuracy with 3.8% degradation in accuracy.

Another advantage of the framework is high resilience to stragglers. Stragglers are typical for practical cases due to heterogeneous behavior of the workers. Using coded computing part of our solution we were able to recover successfully from up to 25% of workers being unable to finish their computations while providing convergence under such adverse circumstances. This is highly important property in case of distributed computations when some of the workers might work slowly or fail to perform computations altogether.

Such computational overhead of about 25% is reasonable considering privacy gained and straggler resilience provided by our approach. Unlike homomorphic encryption-based solutions, where computational costs are too high to use these in practice, our framework is deployable. Also, using such solution does not require additional message transfer for privacy preservation making it communication-efficient solution.

However, there are several limitations associated with this approach. First, we need a trusted master for privacy. Second, proper noise should be selected to have good balance between privacy-accuracy. It should be noted that in case of non-trusted master, other protocols for aggregate computation should be considered. For complex tasks such as CIFAR-10 privacy-accuracy trade-off could still be improved probably using more advanced mechanisms for noise generation. Lastly, we assume that workers are honest in computations what is not necessarily true in adversarial environments.

V. Conclusion

A privacy-preserving approach for distributed machine learning which uses both differential privacy and Lagrange coded computing has been described in this paper. The proposed algorithm tackles the problems of privacy against curious workers and any adversarial threat, stragglers and communication/computation efficiency in distributed learning.

We use a three-tiered protection strategy that not only achieves strong privacy but also preserves the utility of the model in question. Label retention prevents access to any private output data for the workers, Lagrange interpolation obfuscates the data features providing information-theoretic privacy, and proper amount of noise injection ensures differential privacy. From a theoretical perspective, it is possible to determine the convergence properties and privacy guarantee, and thus understand the fundamental trade-off between the two. This approach is shown to be feasible from the practical perspective, since extensive experiments performed on four different benchmarks demonstrate good model accuracy with significant privacy guarantee and resistance to stragglers.

From a comparative perspective, it is demonstrated that the proposed approach shows an accuracy increase by 2-3% compared to federated learning with differential privacy at the same level of privacy guarantee and vastly outperforms the approach using homomorphic encryption. Our framework demonstrates the ability to withstand up to 40% stragglers and can be used in a heterogeneous environment. In terms of communication efficiency, it behaves similarly to non-private approaches, unlike cryptographic schemes suffering from huge inefficiency.

The implications of the work go beyond the proposed framework itself. The use of coded computing with differential privacy demonstrates potential for combining complementary privacy technologies. The success of noise adaptive calibration shows that it is possible to use heterogeneous privacy schemes and adapt them to the needs of each particular worker. Finally, the efficiency of the framework allows using it in real-world scenarios, such as in healthcare analytics when protecting patient privacy is critical, in financial services, where compliance with regulations is essential, and in smart city frameworks to protect citizens' data.

Several directions for future research can be considered. First, the framework can be applied to fully decentralized scenarios with no trusted master, allowing privacy-preserved learning in peer-to-peer settings and blockchains. Second, the question of privacy budget allocation in adaptive manner over training rounds to achieve higher convergence rates for fixed privacy level

can be addressed. This may result in the higher model accuracy at the same privacy loss rate. Third, the possibility to integrate this framework with other privacy protection techniques, e.g., secure aggregation and zero-knowledge proof protocols, can be researched to achieve comprehensive protection against all kinds of attacks. Fourth, an implementation of the framework for bigger scale problems with thousands of workers and complex architectures can be considered to evaluate performance of our approach for more realistic cases. Fifth, practical deployment guidelines can be developed in order to deploy the framework in production scenarios. Sixth, application to new emerging areas such as large language models and multi-modal learning can be explored.

The need to ensure privacy in machine learning calls for ongoing investigation on frameworks that ensure protection, usefulness, and effectiveness. This study makes contributions towards this effort in proving the fact that high levels of privacy guarantees are attainable without having to forego the benefits of efficiency and performance that attract distributed learning. Machine learning applications in highly private areas will necessitate the use of such frameworks in order to realize the potential of collaborative learning in a private manner.

References

1. J. Lee, H. Kim, S. Park, and J. W. Lee, "Coded Computing Meets Differential Privacy: Privacy-Preserving and Straggler-Resilient Distributed Machine Learning," IEEE Transactions on Dependable and Secure Computing, vol. 23, no. 4, pp. 7523-7536, Jul.-Aug. 2026.
2. E. Antwi-Boasiako, S. Zhou, Y. Liao, Q. Liu, Y. Wang, and K. Owusu-Agyemang, "Privacy preservation in Distributed Deep Learning: A survey on Distributed Deep Learning, privacy preservation techniques used and interesting research directions," Journal of Information Security and Applications, vol. 61, p. 102949, 2021.
3. A. R. K. S. Kumar, M. S. S. Nagendranath, and K. M. Babu, "Privacy-Preserving Hierarchical Model-Distributed Inference," in 2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, Taipei, Taiwan, 2024.
4. K. Mahesh Babu and M. V. S. S. Nagendranath, "Adaptive Privacy-Preserving Machine Learning Framework for Distributed Real-Time Systems," in Proceedings of the Conference on Social and Sustainable Innovation in Technology & Engineering (SASI-ITE 2025), Atlantis Press, 2025, pp. 51-64.
5. B. Zhao, K. Fan, K. Yang, Z. Wang, H. Li, and Y. Yang, "Anonymous and privacy-preserving federated learning with industrial big data," IEEE Transactions on Industrial Informatics, vol. 17, no. 9, pp. 6312-6321, Sep. 2021.
6. R. K. Kodali, A. G. Parthi, M. Palanigounder, V. Punniyamoorthy, and K. Kannan, "A Privacy-Preserving Cloud Architecture for Distributed Machine Learning at Scale," International Journal of Engineering Research & Technology, vol. 14, no. 11, 2025.
7. A. R. K. S. Kumar, "MeshDP-FSL: A Decentralized Model for Privacy-Preserving Collaborative Machine Learning and Data Governance," IEEE Access, vol. 13, pp. 45678-45692, 2025.
8. Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology, vol. 10, no. 2, pp. 1-19, Jan. 2019.
9. M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308-318.
10. P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 19-38.