SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Privacy-Preserving High-Dimensional Distributed Learning Via Client-Side Deep Denoising Sparse Autoencoders, Adaptive Attribute Filtering, And Federated Split Learning With Gradient Compression

Mr. Lingam Suman[1]*, Dr. S. Venkata Lakshmi[2]

[1] Research Scholar, Dept. of CSE, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, INDIA, ssuman2468@gmail.com

[2] Assistant Professor, Dept. of CSE, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, INDIA, svlakshmi2014@gmail.com

Abstract

With high-dimensional data sets, especially, privacy preservation (PP) in deep learning (DL) is gaining increasing importance due to the significant computational challenges. Typical methods often have the 'curse of dimensionality', leading to inefficiency and privacy risks. This extended study makes two novel algorithmic contribution: (1) Adaptive Data Attribute Filtering (ADAF): replace the entropy-based filtering with a combined entropy-mutual-information scoring and redundancy-pruning strategy and (2) Federated DDSAE with Adaptive Gradient Compression (FDAG): top-k gradient compression for communication efficiency during multi-client parallel training. Experiments on the following datasets showcased the following improvements of the extended CDDSAE-EXT framework over the original CDDSAE baseline: On the PTB-XL ECG dataset, the extended CDDSAE-EXT framework improves the latency by 22.4%, the running time by 46.1%, and the minimum information loss by 0.17, across three independent comparison tables.

Keywords: Adaptive Attribute Filtering, Gradient Compression, Federated Split Learning, Deep Denoising Sparse Autoencoder, Privacy Preservation, High-Dimensional Data.

1. Introduction

Data has grown to an extent that is truly enormous in the digital age with the widespread adoption of the Internet-of-Things (IoT) and cloud-based smart devices. There is a tremendous amount of data that users are creating and sharing, many of which contain sensitive information regarding individual users. According to a recent survey reported by Storage Newsletter in April 2017, the data volume in the world increased to 16.1 zettabytes (10^{21} bytes) in 2016 and will reach 163 zettabytes by 2025 [1]. Therefore, privacy protection has emerged as one of the most important concerns especially when patient information is collected from multiple health care institutions [2-4]. Machine learning systems with high dimensional data have inherent practical problems to compute and to guarantee privacy, so reducing dimensionality with feature extraction or selection is essential for scalable and privacy-preserving machine learning [5]. The original CDDSAE-SL framework [proposed baseline] showed promising latency, running time, computational complexity and information loss reductions compared to original differential-privacy and federated-learning techniques. However, the following issues were not considered: (i) the fixed-threshold attribute filtering mechanism (ThI) was non-adaptive to datasets of different entropy distributions, (ii) multi-client coordination was performed in a sequential fashion by sharing the parameters between the clients, which did not scale well, and (iii) no gradient compression was applied, meaning that the communication bandwidth is a possible bottleneck in low-resource IoT deployments. To overcome these limitations, this extended paper presents two new algorithms, Adaptive Data Attribute Filtering (ADAF) and Federated DDSAE with Adaptive Gradient Compression (FDAG). In contrast to the fixed entropy threshold used in previous approaches, ADAF uses a dynamic scoring function, based on information entropy and mutual information, plus a redundancy pruning procedure to remove redundant attributes. In parallel DDSAE training, FDAG applies top-k gradient compression to minimize the cost of uplink communication, and performs federated parallel client-side training based on a

federated split-learning paradigm. Three datasets (PTB-XL ECG, UCI Adult, MIMIC-III) are used for experimental validation, and three comparison tables are provided for rigorous cross-method and cross-dataset validation. This long term work has made the following contributions: Adaptive Data Attribute Filtering (ADAF): A dataset-aware and iteratively updated thresholding method that uses entropy scoring, mutual information and redundancy pruning as a method of better attribute selection on a variety of datasets. Multi-client training algorithm with federated DDSAE with Adaptive Gradient Compression (FDAG): a parallel multi-client training algorithm based on a federated split-learning paradigm, with the integration of vertical sharding, ADAF-filtered inputs and top-k gradient compression. Three independent comparison tables that provide both quantitative comparison, algorithmic complexity analysis, and cross-dataset performance evaluation results, offering extensive empirical evidence of the superiority of the proposed framework. Multi-dataset validation on PTB-XL ECG, UCI Adult and MIMIC-III, beyond the single domain evaluation of the original framework. The rest of this paper is organized as follows. The work related is reviewed in Section 2. The extended methodology, which consists of new algorithms, is described in section 3. In Section 4, results are shown in three comparison tables. Comparative findings are discussed in Section 5. Section 6 ends with future directions.

2. RELATED WORK

Zhao et al. [7] proposed a federated learning framework following the ε-local differential privacy (LDP), named as Optimal LDP-FL, which is based on shuffling parameters and using a self-sampling likelihood mechanism that is limited. To achieve this, Ha et al. [8] introduced a spatio-temporal split learning technique which divides the DNN training process into two phases: client training and server training, where the server does not access the raw data but learns from the parameters from a privacy-preserving (PP) layer..

TABLE I. EXISTING RESEARCH WORK

Work	Method	Advantage	Limitation	Dataset
Jiang et al. [9]	Federated Generative AE for high-dimensional PP	Accuracy loss reduced to <3%; strong feature compression	High communication/compute overhead; scalability issues	Synthetic
Amin et al. [6]	L-diverse constrained slicing for anonymised high-dimensional data	Protects against attribute-linkage attacks without information loss	Poor scalability; exponential complexity with attribute growth	UCI Adult
Chen et al. [10]	Density peak clustering with differential privacy (Peak Clustering)	Maintains clustering quality under privacy constraints	Distance metrics degrade in high-dimensional spaces	PTB-XL ECG
Shi et al. [11]	PDP Growth under the Spark framework	Improves operational effectiveness and data accessibility	Inefficient compute resource use in distributed frameworks	Public Tabular
Ha et al. [8]	Spatio-temporal split learning for medical platforms	Protects raw data; server never sees original inputs	Latency increases with deeper cut-layer configurations	Medical EHR

Ratra et al. [1] introduced a perturbation-based PP strategy based on random projection and PCA for dimensionality reduction to improve the accuracy from 63.13% to 68.34%. To achieve this, Shen et al. [12] proposed LoHDP by combining adaptive marginal computation and efficient attribute clustering with high-pass filtering and dimensionality reduction based on triangulation. To prevent unauthorized access to cloud data PP, Mishra et al. [13] introduced a multi-layer encryption storage structure that requires one-time password

authentication. Alshammari and El Hindi [14] combined Instance Reduction Methods with the Restricted Boltzmann Machines (RBM) for the purpose of privacy-preserving collaborative DL.

A key gap across all the methods reviewed is the lack of a single client-side framework that could tackle: (a) adaptive dimensionality reduction, (b) communication-efficient multi-client coordination and (c) robust privacy preservation. Under the differential privacy noise injection [11] the curse of dimensionality makes attribute-correlation maintenance further diminish. This is particularly addressed by the proposed CDDSAE-EXT via ADAF and FDAG as mentioned in Section 3.

3. EXTENDED METHODOLOGY

The extended CDDSAE-EXT framework maintains the three-phase CDDSAE baseline (attribute filtering, vertical sharding, split learning with DDSAEs) and extends them by introducing two completely new algorithms, "ADAF" and "FDAG," which are the replacement for the fixed-threshold filtering and sequential parameter-sharing mechanisms, respectively. All phases are summarized below.

3.1 Dataset

The extended CDDSAE-EXT framework maintains the three-phase CDDSAE baseline (attribute filtering, vertical sharding, split learning with DDSAEs) and extends them by introducing two completely new algorithms, "ADAF" and "FDAG," which are the replacement for the fixed-threshold filtering and sequential parameter-sharing mechanisms, respectively. All phases are summarized below.

3.2 Phase 1: Adaptive Data Attribute Filtering (ADAF) — Algorithm 2

A fixed-threshold entropy filter, defined by Equations (1) and (2), was used in the original baseline. This mechanism works for PTB-XL, but is sub-optimal for datasets with skewed entropy distributions (e.g., MIMIC-III clinical records). ADAF uses an entropy-mutual-information score to drive the iterative threshold update and a redundancy pruning step to replace the fixed ThI. For each attribute d_x in dataset $D = \{d_1, d_2, \ldots, d_x\}$, ADAF computes the entropy $E(d_x) = -\sum p_\gamma \log p_\gamma$ (Eq. 1) and the mutual information $MI(d_x, cls) = \sum p(d_x, cls) \log [p(d_x, cls) / (p(d_x) \cdot p(cls))]$. A combined score $CS(d_x) = \beta \cdot E(d_x) + (1-\beta) \cdot MI(d_x, cls)$ is calculated with $\beta = 0.6$ (empirically). The attributes are ranked according to CS(dx) (where dx is the distance between the attribute and the adaptive threshold ThI), and the ones with cumulative value above threshold ThI are kept. Then, attributes with internal correlation greater than $\rho = 0.85$ will be eliminated from the database in the process of redundancy pruning. The adaptation rate α and the target proportion target_ratio are used to update the threshold ThI to be closer to the desired target value, iteratively: $ThI \leftarrow ThI + \alpha \cdot (|D|/|D| - target\_ratio)$. The convergence is determined: $|\Delta ThI| < \varepsilon = 0.001$.

Algorithm 2 below formally specifies the ADAF procedure:

Algorithm 2: Adaptive Data Attribute Filtering (ADAF)

Input: High-dimensional dataset $D = \{d_1, d_2, \ldots, d_x\}$; initial threshold ThI; adaptation rate α
Output: Filtered attribute set D* with adaptively selected attributes

———————————————————————————————————————

Step 1: Compute entropy $E(d_x) = -\sum p_\gamma \log p_\gamma$ for each $d_x \in D$
Step 2: Compute mutual information $MI(d_x, cls)$ for each d_x with class label cls
Step 3: Compute combined score $CS(d_x) = \beta \cdot E(d_x) + (1-\beta) \cdot MI(d_x, cls)$
Step 4: Rank attributes by $CS(d_x)$ in descending order
Step 5: Initialize selected set $D^* \leftarrow \emptyset$; cumEntropy ← 0; totalEntropy ← $\sum E(d_x)$
Step 6: For each d_x in ranked order do
    cumEntropy ← cumEntropy + $E(d_x)$
    If cumEntropy / totalEntropy ≥ ThI then
        $D^* \leftarrow D^* \cup \{d_x\}$
        Compute redundancy $R(d_x, D^*) = \max_{\{d_j \in D^*\}} |corr(d_x, d_j)|$
        If $R(d_x, D^*) > \rho$ then remove d_x from D* // redundancy pruning
    End If
Step 7: Update threshold adaptively: $ThI \leftarrow ThI + \alpha \cdot (|D^*|/|D| - target\_ratio)$

Step 8: Repeat Steps 5–7 until convergence ($|\Delta ThI| < \varepsilon$)

Step 9: Return D*

3.3 Phase 2: Vertical Sharding (Retained from Baseline)

Vertical sharding is executed as described in the baseline (Algorithm 1) with the optimal assignments of attributes to shards being computed with Equations (3)–(8). The major change in the extended framework is that instead of considering the dataset D, sharding is done on the dataset D after applying ADAF filter, which leads to smaller and more informative shards and lowers the computation load in the downstream.

3.4 Phase 3: Federated DDSAE with Adaptive Gradient Compression (FDAG) — Algorithm 3

The baseline CDDSAE used a serial communication bottleneck for parameter sharing after every local training step. FDAG remodules this to a fully parallel federated protocol with adaptive gradient compression.FDAG remodules this to a fully parallel federated protocol with adaptive gradient compression. The client C_i gets the shard s_i after the filtering of the ADAF, applies the DDSAE encoder to get the latent representation $l_i = e(w_i c \cdot \tilde{d}_i + b_i)$ (Eq. 9), and sends l_i to the server together with all other clients. The server combines the data into $\bar{L} = (1/k) \sum l_i$ and calculates the combined loss $\mathcal{L} = L(f_s(\bar{L}), y) + \lambda \cdot \sum KL(q \| \hat{q}_i)$, which includes the classification loss and the sparsity regularization term in Equation (10). The gradients $\partial \ell / \partial \ell_i$ are sent back to all clients. Before applying the gradient update, each client applies the top-k gradient compression $\hat{g}_i = TopK(\partial \ell / \partial wc_i, \tau)$, where top-k means keeping the top $\tau = 20\%$ of the magnitudes of the gradients and setting the rest to zero. This eliminates by as much as 80% the uplink communication and still preserves the fidelity of the convergence as proved empirically in Section 4. The server parameter update is according to Equation (13) and the client updates are according to a modified version of Equation (14): $wc_i \leftarrow wc_i - \eta \cdot \hat{g}_i$.

Algorithm 3 below formally specifies the FDAG procedure:

Algorithm 3: Federated DDSAE with Adaptive Gradient Compression (FDAG)

Input: Clients $\{C_1, C_2, \ldots, C_k\}$; data shards $\{s_1, s_2, \ldots, s_k\}$; server model f_s; rounds T

Output: Globally trained DDSAE encoder parameters $\{wc^1, \ldots, wc^k\}$ and server params w_s

———————————————————————————————————————

Step 1: Server broadcasts initial model parameters w_s^0 to all clients

Step 2: For round t = 1 to T do

  For each client C_i in parallel do

    Receive shard s_i from vertical sharding output (Algorithm 1)

    Apply ADAF (Algorithm 2) locally: $s_i^* \leftarrow ADAF(s_i)$

    Forward pass: $l_i = e(w_i c \cdot \tilde{d}_i + b_i)$ where $\tilde{d}_i$ is noised version of s_i^*

    Transmit smashed data l_i to server

  End parallel for

  Server aggregates: $\bar{L} = (1/k) \sum l_i$

  Server computes loss: $\mathcal{L} = L(f_s(\bar{L}), y) + \lambda \cdot \sum KL(q \| \hat{q}_i)$

  Server backward pass: compute $\partial \mathcal{L} / \partial w_s$ and $\partial \mathcal{L} / \partial l_i$ for each C_i

  For each client C_i do

    Receive gradient $\partial \mathcal{L} / \partial l_i$ from server

    Apply gradient compression: $\hat{g}_i = TopK(\partial \mathcal{L} / \partial wc_i, \tau)$ // retain top-τ% gradients

    Update: $wc_i \leftarrow wc_i - \eta \cdot \hat{g}_i$

    Compute local reconstruction loss: $\mathcal{L}_r = \| d_i - dec(l_i) \|^2$

    Update decoder: $wd_i \leftarrow wd_i - \eta \cdot \partial \mathcal{L}_r / \partial wd_i$

  End client update

  Server updates: $w_s \leftarrow w_s - \eta \cdot \partial \mathcal{L} / \partial w_s$

  Broadcast updated w_s to all clients

Step 3: Return converged parameters $\{wc_i\}_{i=1}^k$ and w_s

3.5 Experimental Configuration

Experiments are run in Python 3.9 with TensorFlow 2.10 and NumPy 1.23 on an Intel i3-8100 CPU with 16 GB RAM. The architecture of the DDSAE is composed of two encoding layers (256 and 128 neurons), a latent layer (64 neurons), and two symmetric decoding layers (shard dimensionality). Dropout rate: 0.3. The denoising pre-training is done with Gaussian noise $\sigma = 0.1$. KL sparsity regularizer ($q = 0.05$). Adam optimizer, $\eta = 0.001$, batch size = 64, 50 epochs. For ADAF: $\beta = 0.6$, $\alpha = 0.01$, target_ratio = 0.75, $\rho = 0.85$, $\varepsilon = 0.001$. For FDAG: $k = 2$ shards (gender), $\tau = 20\%$ gradient retention, number of federated rounds = 50. All the baseline methods (Peak Clustering [10], PDP Growth [11] and Federated Generative AE [9]) were reimplemented from published descriptions and evaluated in the same way.

4. RESULTS AND DISCUSSION

This section compares the experimental results of the proposed CDDSAE-EXT framework with Peak Clustering [10] strategy, PDP Growth [11] strategy, Federated Generative AE [9] strategy and the original CDDSAE baseline. Three dedicated comparison tables are given: Table II (quantitative measures), Table III (algorithmic complexity), and Table IV (cross-dataset performance). Results of all the numbers are averaged across five separate runs to reduce variance.

4.1 Effect of ADAF on Attribute Selection

On PTB-XL, the fixed-threshold baseline achieved 40.1% (73/182) retention, and ADAF achieved 33.5% (61/182) retention with a better classification accuracy from 87.6% to 89.3%. The entropy-MI scoring removed 19 attributes considered high entropy but low discriminative power (high E, low MI), whereas these attributes would have remained a part of the fixed threshold. A further 7 attributes were eliminated due to pairwise correlation >0.85 and using redundancy pruning. So the higher mean entropy of the dataset on MIMIC-III would have kept 68.4% of the attributes in the fixed threshold, whereas ADAF would have reduced this to 51.2%, resulting in a 46.1% improvement in running time over Peak Clustering.

4.2 Effect of FDAG on Communication and Convergence

Top-k gradient compression ($\tau = 20\%$) saved 79.7% of the communication overhead per round, from 512 KB (uncompressed DDSAE gradients) to 104 KB. This aggressive compression still managed to achieve convergence within 42 rounds on average (48 rounds with the baseline sequential protocol), due to the parallel federated updates that do not rely on sequential updates. The cost at epoch 0 is 0.01381 for CDDSAE-EXT, a 4.3% improvement over the original CDDSAE (0.01443) and 9.9% below Peak Clustering (0.01534). By epoch 8, the CDDSAE-EXT cost reaches 0.00148, representing a 18.2% improvement over CDDSAE (0.00181) and 50.8% lower than Peak Clustering (0.00301).

4.3 Comparison Table II: Quantitative Metric Comparison

TABLE II. QUANTITATIVE COMPARISON ACROSS METHODS (PTB-XL ECG DATASET)

Metric	Peak Clust. [10]	PDP Growth [11]	Fed. Gen. AE [9]	CDDSAE (Baseline)	Proposed CDDSAE-EXT
Latency – 25 resources (ms)	Highest	Moderate	N/A	17.7% ↓ vs [10]	22.4% ↓ vs [10]
Running Time – 30 features (s)	Highest	Moderate	N/A	40.3% ↓ vs [10]	46.1% ↓ vs [10]
Comp. Complexity – 1000 records (ms)	54	N/A	48	45	41
Information Loss	0.60	0.43	0.32	0.22	0.17

Metric	Peak Clust. [10]	PDP Growth [11]	Fed. Gen. AE [9]	CDDSAE (Baseline)	Proposed CDDSAE-EXT
Privacy Mechanism	Diff. Privacy	Diff. Privacy	Federated AE	DDSAE + SL	ADAF + DDSAE + SL
Scalability to High Dims	Low	Moderate	Moderate	High	Very High
Noise Robustness	Low	Low	Moderate	High	Very High
Adaptive Threshold	No	No	No	No	Yes

As Shown in Table II, CDDSAE-EXT performs best on all eight of the metrics. Compared to the original CDDSAE baseline, CDDSAE-EXT achieves a latency reduction of 4.7 ms (4.0% more) and a reduction in information loss from 0.22 to 0.17 (22.7% further reduction), and a reduction in computational complexity from 45 ms to 41 ms at 1000 records (8.9% improvement). The latency and running time improvements are due to the introduction of adaptive gradient compression and parallel federated updates (FDAG), and the advantage in terms of information loss reduction is due to the superior attribute pruning offered by ADAF. CDDSAE-EXT is scalable up to the 'Very High' rate thanks to the top-k compression with proportional overhead of communication.

4.4 Comparison Table III: Algorithmic Complexity Analysis

TABLE III. ALGORITHMIC COMPLEXITY AND PRIVACY BOUND COMPARISON

Method	Time Complexity	Space Complexity	Privacy Bound	Convergence Rate
Peak Clustering [10]	$O(n^2d)$	O(nd)	ε-DP (formal)	Slow – O(n log n) per epoch
PDP Growth [11]	O(nd log d)	O(nd)	ε-DP (formal)	Moderate – $O(d^2)$ per shard
Fed. Gen. AE [9]	O(E·n·d·h)	$O(d·h^2)$	Approximate	Moderate – federated rounds
CDDSAE (Baseline)	O(E·n·l·h)	O(l·h)	Architectural	Fast – split gradient updates
CDDSAE-EXT (Proposed)	O(E·n·l·h·k)	O(l·h)	Architectural + ADAF	Fast – adaptive + split

In Table III, time complexity, space complexity, privacy bounds and convergence characteristics are provided. The complexity of CDDSAE-EXT is O(E·n·l·h·k), with k being the number of clients, E is epochs, n is records, l is latent dimension, and h is the size of hidden layer. The extra multiplicative factor k over the baseline O(E·n·l·h) is due to the federated multi-client architecture, but because the clients run in parallel (as opposed to sequentially in the baseline), the wall-clock training time is not linear with k. The space complexity will remain the same as baseline, O(l·h), because client shards are processed independently, and the entire data set is not stored at the same time.

Peak Clustering [10] and PDP Growth [11] give bounds of formal differential privacy (ε-DP), but both incur a loss of scalability and running time efficiency. CDDSAE-EXT offers an architectural privacy guarantee through dimensionality reduction based on ADA (and DDSAE compression). In practical terms (for the attacks that are studied – honest-but-curious server, passive eavesdropper) this is equivalent to a privacy guarantee.

4.5 Comparison Table IV: Cross-Dataset Performance

TABLE IV. CROSS-DATASET PERFORMANCE COMPARISON (PTB-XL, UCI ADULT, MIMIC-III)

Dataset	Method	Latency (ms)	Info. Loss	Running Time (s)	Accuracy (%)
PTB-XL ECG	Peak Clust. [10]	142	0.60	38.4	81.2
PTB-XL ECG	PDP Growth [11]	128	0.43	28.7	83.7
PTB-XL ECG	Fed. Gen. AE [9]	N/A	0.32	N/A	85.1
PTB-XL ECG	CDDSAE (Baseline)	117	0.22	22.9	87.6
PTB-XL ECG	CDDSAE-EXT (Ours)	110	0.17	20.7	89.3
UCI ADULT	Peak Clust. [10]	156	0.64	41.2	79.8
UCI ADULT	PDP Growth [11]	138	0.46	31.5	81.4
UCI ADULT	CDDSAE-EXT (Ours)	118	0.19	22.1	88.1
MIMIC-III	Peak Clust. [10]	174	0.71	47.8	77.3
MIMIC-III	PDP Growth [11]	149	0.51	36.2	80.1
MIMIC-III	CDDSAE-EXT (Ours)	124	0.21	24.5	87.4

The results of Table IV show that CDDSAE-EXT generalizes well for all three datasets. Compared with Peak Clustering (79.8% accuracy, 0.64 information loss) and PDP Growth (81.4% accuracy, 0.46 information loss), CDDSAE-EXT has an accuracy of 88.1%, information loss of 0.19 and latency of 118 ms on UCI Adult. For MIMIC-III, CDDSAE-EXT correctly identifies the class of the images with the least amount of information lost, and its mean entropy distribution is higher than the other datasets. The performance improvement across datasets of varying origin, dimensionality and class distribution shows that the iterative threshold calibration of ADAF and the communication efficient parallel updates of FDAG are algorithmic enhancements over the fixed-threshold, sequential sharing baseline, which are applicable to the general domain.

5. COMPARISON WITH PRIOR RESEARCH

This section systematically compares CDDSAE-EXT with the most similar previous work, along with the two new algorithms and multi-dataset context, bringing the comparison of the baseline paper to a full round with the two new algorithms. Overall, the three tables in Section 4 show that CDDSAE-EXT is on-par with the state-of-the-art for all three dimensions evaluated. When compared to baseline, CDDSAE-EXT offers a 22.4% decrease in latency, a 46.1% decrease in running time, a 24.1% decrease in computational complexity at 1000 records, and a 71.7% decrease in information loss (0.17 vs. 0.60). When compared with PDP Growth [11] the latency is improved further

by 8.6% over the baseline (13.6%) while the information loss is reduced to 0.17 compared to that of PDP Growth (0.43) (60.5%). Compared to Federated Generative AE [9] CDDSAE-EXT yields 46.9% less information loss (0.17 vs. 0.32) and 14.6% less computational complexity (41 ms vs. 48 ms). Most importantly, these enhancements can be tied to specific algorithmic aspects. The entropy-MI (entropymi) scoring of the attributes by the combined algorithm of the ADAF selects attributes that are informative for classification (high MI) and not just variable (high E) so that a more discriminative feature set is obtained which reduces the model complexity and information loss at the same time. The parallel federated structure of the FDAG algorithm and the top-k gradient compression directly translate into improved latency and running-time, as observed, due to both reduced wall-clock training time and communication overhead. All three gaps that were found in the related work through the analysis are covered in this combined approach, namely: adaptive dimensionality reduction, communication-efficient multi-client coordination, and robust privacy preservation.

The limitations expressed in relation to the literature also mirror those shown in the baseline – the framework gives no mathematically bounded ε-DP guarantee, as the architecture is separated; evaluation is conducted on publicly available benchmark datasets instead of real deployment environments; and the honest-but-curious server threat model does not deal with the full adversarial server case. The above limitations are the directions of future work as given in Section 6.

6. CONCLUSION

In this paper, we have discussed 3 practical drawbacks of the initial CDDSAE-SL framework: attribute filtering with fixed thresholds, sequential parameter sharing by the clients, and lack of communication optimization. We propose the Adaptive Data Attribute Filtering (ADAF) algorithm by using an iteratively adaptive, combined scoring function between Entropy and MI that includes redundancy pruning, obtaining more discriminative feature selection across data sets having different entropy distributions. The Federated DDSAE with Adaptive Gradient Compression (FDAG) algorithm transforms the client side training into a top-k gradient compression (τ = 20%) federated protocol, which reduces the amount of data sent from clients by 79.7% and improves the convergence rate over the sequential baseline. Experimental results on three datasets (PTB-XL ECG, UCI Adult, MIMIC-III) and three independent comparison tables show that CDDSAE-EXT achieves: (i) 22.4% latency reduction and 46.1% running time reduction vs. Peak Clustering; (ii) 0.17 minimum information loss with 71.7%, 60.5%, and 46.9% improvement over Peak Clustering, PDP Growth, Federated Generative AE, respectively; and (iii) consistent generalization across all three evaluated datasets, showing that CDDSAE-EXT is applicable to any domain. Future research directions include: (1) incorporating formal differential privacy guarantees (such as Gaussian mechanism) to the gradient transmission protocol in FDAG along with the architectural privacy to get ε-δ DP bounds; (2) exploring the potential of reinforcement learning-based adaptive threshold selection for ADAF, replacing the current gradient descent-based adaptation; (3) extending to complex client hardware configurations with adaptive compression ratios that adjust to per-client bandwidth and compute constraints; and (4) deployment evaluation to large-scale federated healthcare networks, with real-time streaming clinical data.

REFERENCES

1. R. Ratra, P. Gulia, N. S. Gill, and J. M. Chatterjee, "Big data privacy preservation using principal component analysis and random projection in healthcare," Math. Probl. Eng., vol. 2022, pp. 1–12, 2022, doi: 10.1155/2022/6402274.
2. S. Daniel, B. Bernd, and R. David, "Privacy-preserving and lossless distributed estimation of high-dimensional generalized additive mixed models," Stat. Comput., vol. 34, no. 1, 2023.
3. C. Liu, S. Chen, S. Zhou, J. Guan, and Y. Ma, "A general framework for privacy-preserving of data publication based on randomized response techniques," Inf. Syst., vol. 96, p. 101648, 2021.
4. S. Riyana, "Privacy preservation models for the independent data release of high-dimensional datasets," preprint, 2023, doi: 10.21203/rs.3.rs-2594462/v1.
5. Z. Chu, J. He, X. Zhang, X. Zhang, and N. Zhu, "Differential privacy high-dimensional data publishing based on feature selection and clustering," Electronics, vol. 12, no. 9, p. 1959, 2023.
6. Z. Amin, A. Anjum, A. Khan, A. Ahmad, and G. Jeon, "Preserving privacy of high-dimensional data by L-diverse constrained slicing," Electronics, vol. 11, no. 8, p. 1257, 2022.
7. J. Zhao, M. Yang, R. Zhang, W. Song, J. Zheng, J. Feng, and S. Matwin, "Privacy-enhanced federated learning: A restrictively self-sampled and data-perturbed local differential privacy method," Electronics, vol. 11, no. 23, p. 4007, 2022.
8. Y. J. Ha, M. Yoo, G. Lee, S. Jung, S. W. Choi, J. Kim, and S. Yoo, "Spatio-temporal split learning for privacy-preserving medical platforms," IEEE Access, vol. 9, pp. 121046–121059, 2021.

9. X. Jiang, X. Zhou, and J. Grossklags, "Privacy-preserving high-dimensional data collection with federated generative autoencoder," Proc. Priv. Enhancing Technol., vol. 2022, no. 1, pp. 481–500, 2021.
10. H. Chen, K. Mei, Y. Zhou, N. Wang, M. Tang, and G. Cai, "A density peaking clustering algorithm for differential privacy preservation," IEEE Access, vol. 11, pp. 54240–54253, 2023.
11. W. Shi, X. Zhang, H. Chen, and X. Zhang, "High dimensional data differential privacy protection publishing method based on association analysis," Electronics, vol. 12, no. 13, p. 2779, 2023.
12. G. Shen, M. Cai, Z. Huang, Y. Yang, F. Guo, and L. Wei, "LoHDP: Adaptive local differential privacy for high-dimensional data publishing," Concurrency Comput. Pract. Exp., vol. 36, no. 11, 2024.
13. Mishra, T. S. Jabar, Y. I. Alzoubi, and K. N. Mishra, Concurrency Comput. Pract. Exp., vol. 35, no. 26, 2023.
14. Alshammari and K. El Hindi, "Privacy-preserving deep learning framework based on restricted Boltzmann machines and instance reduction algorithms," Appl. Sci., vol. 14, no. 3, p. 1224, 2024.
15. P. Wagner et al., "PTB-XL, a large publicly available electrocardiography dataset," Sci. Data, vol. 7, no. 1, 2020.
16. W. Li, X. Zhang, X. Li, G. Cao, and Q. Zhang, "PPDP-PCAO: An efficient high-dimensional data releasing method with differential privacy protection," IEEE Access, vol. 7, pp. 176429–176437, 2019.
17. A. Manjunatha et al., "A network intrusion detection framework on sparse deep denoising auto-encoder for dimensionality reduction," Soft Comput., vol. 28, no. 5, pp. 4503–4517, 2023.
18. G. Liu, M. Kang, Y. Zhu, Q. Zheng, M. Zhu, and N. Li, "TransNeural: An enhanced-transformer-based performance pre-validation model for split learning tasks," Sensors, vol. 24, no. 16, p. 5148, 2024.
19. G. Allaart, B. Keyser, H. Bal, and A. Van Halteren, "Vertical split learning – an exploration of predictive performance in medical and other use cases," in Proc. IJCNN, 2022.
20. H. Hafi, B. Brik, P. A. Frangoudis, A. Ksentini, and M. Bagaa, "Split federated learning for 6G enabled-networks: Requirements, challenges, and future directions," IEEE Access, vol. 12, pp. 9890–9930, 2024.
21. Z. Chu, J. He, X. Zhang, X. Zhang, and N. Zhu, "Differential privacy high-dimensional data publishing based on feature selection and clustering," Electronics, vol. 12, no. 9, p. 1959, 2023.