敵対的オートエンコーダを用いた垂直連合学習

VERTICAL FEDERATED LEARNING USING ADVERSARIAL AUTOENCODERS

黄　凱
Kai HUANG

指導教員 劉　慶豊

法政大学大学院理工学研究科システム理工学専攻修士課程

With the increasing reliance on machine learning in the financial sector, data privacy has become a critical challenge. Vertical Federated Learning (VFL) enables multiple entities to collaboratively train models without sharing raw data. However, existing privacy-preserving techniques, such as homomorphic encryption and differential privacy, often incur high computational costs or degrade model performance. To address these issues, this study proposes a privacy-preserving VFL framework based on Adversarial Autoencoders (AAE). By encoding features into privacy-enhanced latent representations and aggregating them at a central server, our approach effectively mitigates data leakage risks while maintaining model performance. Experimental results on three financial datasets demonstrate that the proposed method achieves competitive classification performance while significantly enhancing privacy protection. This study provides a practical and efficient solution for privacy-preserving machine learning in financial applications.

Key Words : *Federated Learning, Adversarial Autoencoder, Privacy-preserving, Financial Data Security*

1. INTRODUCTION

Machine learning (ML) has become essential in finance, enabling applications such as credit risk assessment and fraud detection. High-performance ML models require large-scale data from multiple institutions [1]. However, concerns over data privacy, security, and regulatory compliance hinder centralized data collection. Federated Learning (FL) provides a decentralized solution where institutions train local models and share only aggregated parameters [2]. Various privacy-enhancing techniques, including k-anonymity [3], l-diversity [4], t-closeness [5], differential privacy [6], and homomorphic encryption [7, 8], mitigate risks but often degrade model accuracy or impose high computational costs.

FL is classified into Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL) [9]. HFL applies when institutions share the same features but different samples, whereas VFL is suited for cases where institutions hold complementary features for the same entities. VFL is particularly relevant in finance for tasks like credit scoring and fraud detection under stringent privacy regulations. However, VFL faces challenges, including privacy-preserving techniques reducing model performance, data leakage risks through intermediate updates, and high computational and communication costs.

Existing research has leveraged autoencoders to secure feature representations in VFL [10]. However, traditional autoencoders remain vulnerable to reconstruction attacks. Advances in Generative Adversarial Networks (GANs) have led to Adversarial Autoencoders (AAE), which incorporate a discriminator network to enforce a predefined latent space distribution, enhancing privacy without significantly compromising model utility.

This study proposes an AAE-based privacy-preserving framework for VFL in financial applications. Our approach transforms raw data into secure feature representations, mitigating reconstruction attacks while preserving predictive performance. Compared to conventional autoencoders, AAE enhances resilience against adversarial inference and reduces privacy risks. Additionally, our method optimizes the trade-off between privacy protection and computational efficiency.

We evaluate our framework on three benchmark financial datasets : Adult Income [11], Default of Credit Card Clients [12], and Bank Marketing [13], and compare it against overcomplete autoencoders(OAE) [10] and centralized learning. Experimental results demonstrate that our AAE-VFL framework significantly improves privacy protection while maintaining high classification accuracy, advancing privacy-focused machine learning for financial applications. Our findings support the broader adoption of federated learning in privacy-sensitive environments.

2. RELATED WORK

(1) FEDERATED LEARNING

Federated Learning (FL) is a framework that enables a group of clients to collaborate in resolving machine learning issues under a single coordination of a server. In FL, a client holds its training data and, therefore, its privacy is preserved. FL is supported through two principal concepts: model update transmissions and local computation. FL effectively addresses the issue of concerns over privacy and reduces the cost incurred in traditional centralized machine learning approaches through these two processes.

In Federated Learning (FL), individual raw data for each client is kept locally and not exchanged or shared with any party. Instead, a device trains its model with its private information and uploads model parameters to a shared server for aggregation. The server aggregates model parameters and updates a global model and distributes an updated model to the clients, thereby achieving desired learning objectives.

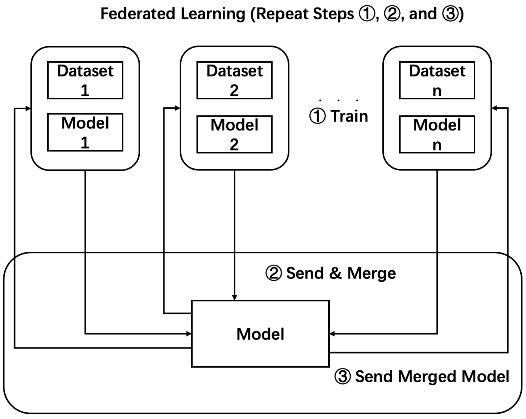

Figure 1. The process of Federated Learning.

According to the classification by Yang et al. [9], federated learning can be divided into three types based on how the datasets owned by participants are distributed. Specifically, it focuses on the sample space, feature space, and ground truth labels of each dataset.

- Sample Space: The sample space refers to the set of all data samples. For example, in a classification problem that determines whether a given image is a picture of a dog, an individual image represents one sample (or instance). Similarly, in tasks using tabular data where each row represents information about a user, each row (i.e., each user) corresponds to one sample.
- Feature Space: The feature space refers to the set of all possible features that a sample can have. For images, the numerical values of each pixel in the sample image represent features. For tabular data, each column (attribute) represents an individual feature.
- Ground Truth Labels: Correct information for an individual instance is understood as ground truth labels. In training, a prediction for a specific sample is produced by the model, and model optimization is performed through minimizing predicted values and ground truth labels' difference.

Based on these elements, federated learning is classified into the following three types:

- Horizontal Federated Learning (HFL): The sample spaces of the datasets differ, but the feature spaces have a common subset. For example, Google uses HFL to enable mobile phone users to collaboratively train a next-word prediction model using their datasets [14].
- Vertical Federated Learning (VFL): The datasets share a common sample space, but their feature spaces differ. For example, banks use VFL to collaborate with billing agencies to build financial risk models for corporate customers [15].
- Federated Transfer Learning (FTL): The datasets have partially overlapping sample spaces and feature spaces. For instance, EEG (electroencephalogram) data from multiple subjects with heterogeneous distributions can be used to collaboratively build a brain–computer interface (BCI) model using FTL [16].

Figure 2, 3, 4 below summarizes the states of datasets in the three types of federated learning.

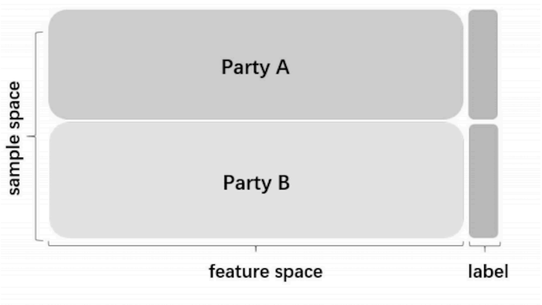

Figure 2. Horizontal Federated Learning

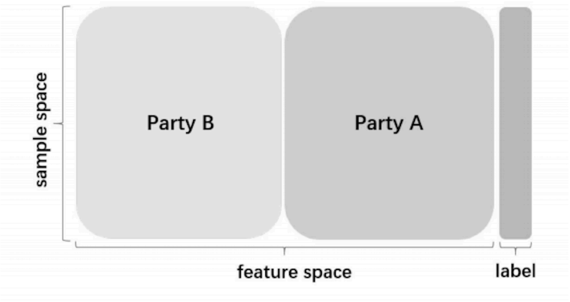

Figure 3. Vertical Federated Learning

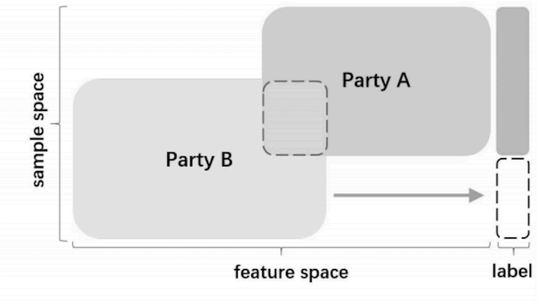

Figure 4. Federated Transfer Learning

(2) VERTICAL FEDERATED LEARNING

Vertical federated learning is most applicable in scenarios when a group of data owners have access to common samples but have datasets with different features. In this case, let participants A and B have datasets represented as

$$X_A \in \mathbb{R}^{n\times d_A}, \quad X_B \in \mathbb{R}^{n\times d_B} \tag{1}$$

where n is the number of samples, and d_A and d_B are the number of features for A and B, respectively. Both participants share the same sample IDs and align their data based on these IDs for each sample.

In vertical federated learning, the goal is to collaboratively train a global model $f(X_A, X_B)$ by exchanging gradients without sharing raw data. A typical optimization problem in this setting can be formulated as follows:

$$\min_{\theta_A,\theta_B} \sum_{i=1}^{n} L\left(f\left(X_A^{(i)};\theta_A\right),\, f\left(X_B^{(i)};\theta_B\right),\, y^{(i)}\right) \tag{2}$$

where θ_A and θ_B are the model parameters for A and B, respectively, and L is the loss function.

(3) AUTOENCODER

Autoencoders (AE) are a class of neural networks used for unsupervised representation learning, mapping high-dimensional input data into a lower-dimensional latent space while preserving essential information [17]. The primary objective of an autoencoder is to learn a function $f : X \to Z$ that transforms input data X into a latent representation Z, from which the decoder attempts to reconstruct X as faithfully as possible. This process can be formulated as follows:

$$Z = f_\theta(X), \tag{3}$$

$$\hat{X} = g_\phi(Z). \tag{4}$$

where f_θ represents the encoder parameterized by θ, g_ϕ represents the decoder parameterized by ϕ, and $\hat{X}$ is the reconstructed output.

Loss Functions The optimization objective of an autoencoder is to minimize the reconstruction loss, commonly expressed as:

$$L_{\text{AE}} = \frac{1}{N}\sum_{i=1}^{N} \|X_i - \hat{X}_i\|^2. \tag{5}$$

where N is the number of samples, and $\|\cdot\|^2$ denotes the squared error (typically mean squared error, MSE). This loss function ensures that the reconstructed data $\hat{X}$ is as close as possible to the original input X, capturing the most significant features while discarding noise.

(4) ADVERSARIAL AUTOENCODER

Following the introduction of Generative Adversarial Networks (GAN) by Goodfellow [18], adversarial training has demonstrated notable success in fields such as image generation, data augmentation, and representation learning. On the other hand, AutoEncoders (AE) have been extensively employed for feature extraction and dimensionality reduction, but traditional AE often encounter difficulties in flexibly modeling data in the latent space.

To address these challenges, Makhzani proposed the Adversarial AutoEncoder (AAE) [19]. By incorporating adversarial training into the latent space of an AutoEncoder, AAE encourages the encoder ' s output distribution to align with a prescribed prior, thereby improving both generative quality and latent representation capacity when compared to standard AE.

Network Architecture An AAE consists of three main components—an encoder, a decoder, and a discriminator—building upon the base structure of a standard AutoEncoder:

- **Encoder:** Maps the input data X to a latent representation Z:

$$Z = f_\theta(X) \tag{6}$$

  where f_θ represents the encoder parameterized by θ.

- **Decoder:** Reconstructs X from the latent representation Z:

$$\hat{X} = g_\phi(Z) \tag{7}$$

  g_ϕ represents the decoder parameterized by ϕ, and $\hat{X}$ is the reconstructed output.

- **Discriminator:** Distinguishes between encoded latent variables Z and samples from a prior distribution $p(Z)$.

Loss Functions The training objective of AAE typically combines two main loss terms: a reconstruction loss and an adversarial loss:

$$L_{\text{AAE}} = L_{\text{reconstruction}} + \lambda L_{\text{adversarial}} \tag{8}$$

- **Reconstruction Loss:** The reconstruction loss ensures that the output of the decoder $\hat{x}$ is close to the original input x. Common choices include Mean Squared Error (MSE) or cross-entropy. An MSE-based reconstruction loss can be written as:

$$L_{\text{reconstruction}} = \frac{1}{N}\sum_{i=1}^{N} \|X_i - \hat{X}_i\|^2 \tag{9}$$

- **Adversarial Loss:** In order to force the latent distribution produced by the encoder to match a chosen prior $p(z)$, an adversarial training mechanism is applied in the latent space. This setup is analogous to a standard GAN, with a discriminator distinguishing between samples from $p(z)$

and Z. The typical formulation for the adversarial loss (using the binary cross-entropy objective) is:

$$L_{\text{adversarial}} = \mathbb{E}_{p(Z)}[\log D_\psi(Z)] + \mathbb{E}_{q_\theta(Z|X)}[\log(1 - D_\psi(Z))] \quad (10)$$

where $D_\psi(\cdot)$ is the discriminator, and λ balances the trade-off between reconstruction loss and adversarial regularization.

- **Training Procedure:** Training an AAE typically proceeds in an alternating fashion:
  - **Discriminator Update:** Hold the encoder and decoder fixed. Update the discriminator D_ψ by maximizing its ability to distinguish real samples $z \sim p(z)$ from encoder outputs Z.
  - **Encoder and Decoder Update:** Fix the discriminator. Update the encoder Z and decoder $\hat{X}$ jointly, minimizing both the reconstruction loss and the adversarial loss to " fool " the discriminator into believing that the latent vectors from Z come from $p(z)$.

A typical combined objective can be written as:

$$\min_{\theta,\phi} \max_{\psi} L_{\text{rec}}(\theta, \phi) + \lambda L_{\text{adv}}(\theta, \psi), \quad (11)$$

where λ is a hyperparameter that balances the importance of the reconstruction loss relative to the adversarial loss.

3. PROPOSED METHOD

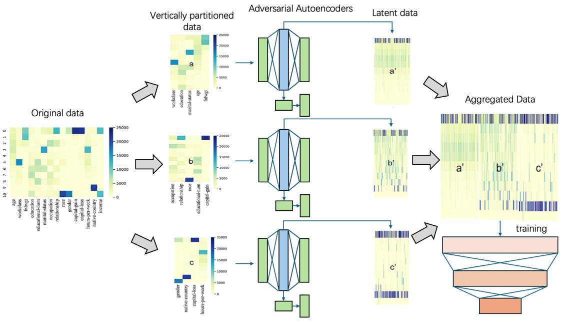

Figure 5. Training Workflow.

```
for each client $i$ do
    Encode local data $X_i$ using AAE to obtain latent
      representation $Z_i$
    Train AAE locally to minimize reconstruction loss
      + adversarial loss
    Send $Z_i$ to the central server
end
for central server do
    Aggregate received $Z_i$ from all clients
    Train a global classifier using the aggregated $Z$
    Distribute model updates back to clients
end
```

Algorithm 1: AAE-VFL Training

(1) OVERVIEW OF VERTICAL FEDERATED LEARNING

Our work targets the unique challenges of vertical federated learning (VFL), where features (rather than samples) are distributed across multiple data holders. Unlike horizontal federated learning, in which participants each have data samples with the same set of features, VFL demands an effective mechanism to align partial records residing at different organizations. This alignment ensures that each data sample ' s features can be combined without revealing private information, thereby facilitating collaborative model training.

(2) DATA ALIGNMENT AND PARTITIONING

To handle the alignment process, we first match the IDs across all participating clients. This step ensures that each data holder can correctly identify the samples they share. For instance, row 3 in Client A's dataset corresponds to row 3 in Clients B and C's datasets. Once the mapping is established, we split the dataset vertically among the clients. Because real-world data can exhibit varying sizes and degrees of overlap, we evaluate the versatility of our approach by dividing the dataset into differing numbers of subsets. This partitioning simulates real-world scenarios where multiple organizations each possess unique feature subsets of the same user base.

(3) LOCAL TRAINING WITH ADVERSARIAL AUTOENCODERS

Following the vertical partitioning, each client independently trains a local model based on an adversarial autoencoder (AAE) framework (Figures 5 a, b, c). By leveraging AAE, each client learns a latent representation of its partitioned features. The core benefit of using adversarial training in the latent space is to encode and preserve the most discriminative information without exposing raw features to other parties.

Concretely, each AAE comprises:

1. **Encoder**: Projects input data into a latent representations space.
2. **Decoder**: Reconstructs the original features from the latent vectors.
3. **Discriminator**: Encourages these latent representations to follow a chosen prior distribution, thereby regularizing the representation.

This adversarial mechanism makes it difficult to recover sensitive, high-dimensional details from the latent representations alone, providing an extra layer of data privacy while still capturing the essential patterns needed for downstream tasks.

(4) LATENT REPRESENTATION AGGREGATION AT THE CENTRAL SERVER

Once trained locally, each client extracts the learned latent representations of its respective features (Figures 5 a', b', c'). These latent representations are then securely transferred to a central server, which aggregates them. Because each client's autoencoder is tuned to produce consistent embeddings aligned with a common prior, the aggregated latent space is effectively unified, despite the original data being distributed.

At the central server, we combine the latent representations from all clients to train a global model for tasks such as classification. This approach significantly reduces communication overhead (since latent representations are typically smaller in dimension than raw data) and addresses privacy concerns by sharing only encoded representations.

(5) MODEL COMPARISON AND BASELINE

To benchmark our approach, we use a centralized model (trained on the full dataset without partitioning) as a baseline. We then compare:

1. A machine learning model trained purely on the vertically partitioned data (Figures 5 a, b, c), where each client ' s features remain separated.
2. A machine learning model trained on the latent representations from each client (Figures 5 a', b', c'), demonstrating how the AAE-based aggregation improves performance.

Finally, by comparing the latent representation-based vertical federated model with the centralized baseline, we assess the practical trade-offs between accuracy and privacy preservation. For performance evaluation, we employ standard classification metrics such as accuracy and AUROC (Figure 5).

Overall, our method bridges the gap between privacy requirements and the need for high-quality modeling in distributed environments. It lays the groundwork for future exploration of adversarial learning techniques within vertical federated settings, aiming to further refine the balance between data security and model performance.

4. EXPERIMENTS

(1) DATASETS

a) ADULT INCOME DATASET

The Adult Income dataset [11] consists of two classification labels indicating whether an individual earns more than $50,000 a year. It includes 8 categorical and 6 continuous variables as input features. Initially, the dataset contained records for 37,155 individuals earning $50,000 or less and 11,687 individuals earning more than $50,000 annually.To balance the dataset, we applied random undersampling, selecting a subset of 11,687 individuals from the lower-income group. This resulted in a final dataset of 23,374 individuals, ensuring an equal class distribution and setting the baseline prediction probability at 50%.As shown in Table 1, we then partitioned this dataset vertically into three segments, simulating a scenario where three different organizations each hold partial information about the same individuals.

Table 1. Dataset composition and training parameters with division to simulate vertically partitioned data.

Dataset	Division	Size	Features	Autoencoder
Adult Income	3 sites	23,374	5, 5, 4	64-128-64
Credit Card	6 sites	15,762	4, 4, 4, 4, 4, 4	64-128-64
Bank Marketing	4 sites	9,280	5, 5, 5, 5	64-128-64

Table 2. Features and labels for the Adult Income Dataset.

ID	Age	Workclass	⋯	Native Country	Income
1	39	State-gov	⋯	United States	$\leq 50K$
2	50	Self-emp-not-inc	⋯	United States	$\leq 50K$
3	38	Private	⋯	United States	$\leq 50K$
⋯	⋯	⋯	⋯	⋯	⋯
48,842	52	Self-emp-inc	⋯	United States	$> 50K$

b) DEFAULT OF CREDIT CARD CLIENTS DATASET

The Default of Credit Card Clients dataset [12] denoted as Credit Card in Table 1, is a publicly available dataset used for predicting credit card payment defaults (binary classification). The original dataset consists of 30,000 records, where 6,636 (22.12%) clients defaulted on their payments, while 23,364 (77.88%) did not.

To mitigate the class imbalance problem, we applied random undersampling, selecting an equal number of non-defaulted clients. Specifically, we randomly sampled 6,636 non-defaulted cases, resulting in a balanced dataset containing 13,272 records. This balanced dataset was then vertically partitioned into 6 sites to simulate a vertical federated learning (VFL) scenario, where each site holds different subsets of the features while preserving the same sample IDs (Table 1).

Table 3. Features and labels for the Credit Card.

ID	LIMIT_BAL	SEX	⋯	Default (Next Month)
1	20,000	2	⋯	1
2	120,000	2	⋯	0
3	90,000	2	⋯	0
⋯	⋯	⋯	⋯	⋯
30,000	50,000	1	⋯	1

c) BANK MARKETING

The Bank Marketing dataset [13] is used to predict whether a customer will subscribe to a term deposit (binary classification). The initial dataset contains 41,188 entries, where the target variable is highly imbalanced: only 4,640 entries (11.28%) are labeled as `"yes"`, while 36,548 entries (88.72%) are labeled as `"no"`.

To address the class imbalance problem and ensure a balanced baseline dataset, we randomly sampled 4,640 entries labeled as `"no"` to match the 4,640 `"yes"` entries. This process resulted in a balanced dataset containing a total of 9,280 entries. We then vertically divided the dataset into 4 sites, with each site containing a subset of the features (Table 1). This setup was used to simulate a vertical federated learning (VFL) scenario while maintaining a balanced target variable distribution for better model performance and fair evaluation.

(2) TRAINING WORKFLOW

In a setup mirroring real-world conditions where data is divided vertically among multiple entities, we split each of the three datasets and utilized a third-party relay server to align corresponding entries across all sites. For instance, this alignment ensured that the third entry on client A matched the same entry

Table 4. Features and labels for the Bank Marketing.

ID	age	job	···	y
1	56	housemaid	···	no
2	57	services	···	no
3	37	services	···	no
···	···	···	···	···
41188	74	retired	···	no

on clients B and C. To examine the robustness of our approach, various data partition strategies were employed (Table 1). Subsequently, each site independently trained its own Adversarial Autoencoder model (Figure 3 a, b, c). Upon completion of these local training processes, the code-layer representations (Figure 3 a ′ , b ′ , c ′) from each site were collected and integrated for further model training. We implemented this workflow using PyTorch [20]. We then assessed classification performance primarily through accuracy and the area under the receiver operating characteristic curve (AUROC).

(3) EVALUATION METRICS

1. **Accuracy**

   Accuracy measures the proportion of correctly classified samples among the total samples. It provides a general overview of the model's performance.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{12}$$

2. **AUROC (Area Under Receiver Operating Characteristic Curve)**

   AUROC represents the area under the ROC curve and is used to evaluate the model's classification performance, especially for imbalanced datasets. This metric helps confirm the effectiveness of the latent representations in downstream tasks.

(4) RESULTS

a) LATENT REPRESENTATION AND PRIVACY ANALYSIS

The adversarial autoencoder (AAE) effectively encoded the original data, which had been vertically partitioned, into latent representations. Notably, this process was achieved without the need for domain-specific expertise.The generated latent representations exhibited substantial differences from the original data in both feature space and distribution (Figure 5a ′ , b ′ , c ′), highlighting the network's effectiveness in concealing raw information. This transformation not only preserved the essential patterns required for downstream tasks but also ensured robust data security by minimizing the risk of sensitive information being reconstructed from the latent space.

b) CLASSIFICATION PERFORMANCE

After aggregating the latent data, we compared the proposed model with both centralized models and independently trained models using the original vertically partitioned data and their corresponding latent representations (Table 3). The results showed that the performance of the adversarial autoencoder (AAE) model improved as the capacity of the latent representation increased. Additionally, employing categorical embeddings for categorical variables ensured that the transformed representations remained continuous, enhancing their compatibility with tabular neural network models.

For the Bank Marketing dataset, which has a relatively large number of rows, the accuracy and AUROC remained stable across different sites, showing minimal performance fluctuation. The Adult Income dataset also demonstrated consistent results, further confirming the robustness of the AAE-based transformation. Despite the vertically partitioned data structure, the Default of Credit Card Clients Dataset, which was divided into six sites, exhibited only a minor reduction in accuracy and AUROC. This suggests that the proposed method effectively preserves data privacy while maintaining acceptable utility for downstream tasks (Table 5).

Table 5. Classification results of the three datasets.

Site	Adult Income		Credit Card		Bank Marketing	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
Central	0.83	0.91	0.80	0.79	0.89	0.95
OAE	0.82	0.90	0.71	0.79	0.88	0.94
AAE	0.82	0.90	0.71	0.79	0.88	0.94

5. CONCLUSION

This study proposes a privacy-preserving framework for Vertical Federated Learning (VFL) using Adversarial Autoencoders (AAE) to balance privacy, computational efficiency, and model performance. By exchanging latent representations instead of raw data, the method enhances privacy protection and reduces the risk of sensitive information leakage. Compared to differential privacy, it avoids noise injection and prevents significant performance degradation, while imposing lower computational and communication overhead than homomorphic encryption. However, challenges remain in terms of explainability and communication costs, as leveraging latent representations can obscure decision-making and high-dimensional data transmission may become a bottleneck. Experimental results on three financial datasets demonstrate that AAE-VFL significantly improves privacy protection while maintaining high classification accuracy. This study highlights AAE ' s potential for advancing privacy-preserving machine learning, particularly in sensitive domains like finance and healthcare, enabling secure cross-institution collaboration. Future work should focus on optimizing the trade-off between privacy and efficiency to enhance real-world applicability.

REFERENCES

[1] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE intelligent systems*, 24(2):8–12, 2009.

[2] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong.

Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[3] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[4] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *Acm transactions on knowledge discovery from data (tkdd)*, 1(1):3–es, 2007.

[5] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd international conference on data engineering*, pages 106–115. IEEE, 2006.

[6] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[7] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[8] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124, 2011.

[9] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[10] Dongchul Cha, MinDong Sung, Yu-Rang Park, et al. Implementing vertical federated learning using autoencoders: Practical application, generalizability, and utility study. *JMIR medical informatics*, 9(6):e26598, 2021.

[11] Becker B. and Kohavi R. Adult [dataset], 1996.

[12] I-Cheng Yeh and Cheh-Chih Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

[13] Rita P. Moro, S. and P. Cortez. Bank Marketing. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5K306.

[14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[15] Yong Cheng, Yang Liu, Tianjian Chen, and Qiang Yang. Federated learning for privacy-preserving ai. *Communications of the ACM*, 63(12):33–36, 2020.

[16] Ce Ju, Dashan Gao, Ravikiran Mane, Ben Tan, Yang Liu, and Cuntai Guan. Federated transfer learning for eeg signal classification. In *2020 42nd annual international conference of the IEEE engineering in medicine & biology society (EMBC)*, pages 3040–3045. IEEE, 2020.

[17] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 27, 2014.

[19] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.