RVE-PFL: Robust Variational Encoder-Based Personalized Federated Learning Against Model Inversion Attacks

Wael Issa, Nour Moustafa, *Senior Member, IEEE*, Benjamin Turnbull, *Senior Member, IEEE*, and Kim-Kwang Raymond Choo, *Senior Member, IEEE*

***Abstract*— Federated learning (FL) enables distributed joint training of machine learning (ML) models without the need to share local data. FL is, however, not immune to privacy threats such as model inversion (MI) attacks. The conventional FL paradigm often uses privacy-preserving techniques, and this could lead to a considerable loss in the model's utility and consequently compromised by MI attackers. Seeking to address this limitation, this paper proposes a robust variational encoder-based personalised FL (RVE-PFL) approach that mitigates MI attacks, preserves model utility, and ensures data privacy. RVE-PFL comprises an innovative personalised variational encoder architecture and a trustworthy threat model-integrated FL method to autonomously preserve data privacy, and mitigate MI attacks. The proposed architecture seamlessly trains heterogeneous data at every client, while the proposed approach aggregates data at the server side and effectively discriminates against adversarial settings (i.e., MI); thus, achieving robustness and trustworthiness in real-time. RVE-PFL is evaluated on three benchmark datasets, namely: MNIST, Fashion-MNIST, and Cifar-10. The experimental results revealed that RVE-PFL achieves high accuracy level while preserving data and tuning adversarial settings. It outperforms Noising before Model Aggregation FL (NbAFL) with significant accuracy improvements of 8%, 20%, and 59% on MNIST, Fashion-MNIST, and Cifar-10, respectively. These findings reinforce the effectiveness of RVE-PFL in protecting against MI attacks while maintaining the model's utility. The source code for RVE-PFL can be found on GitHub: https://github.com/UNSW-Canberra-2023/RVE-PFL.**

***Index Terms*— Federated learning (FL), variational autoencoder (VAE), model inversion (MI) attack, differential privacy (DP).**

I. Introduction

FEDERATED learning (FL) has gained widespread adoption in various applications, ranging from telecommunications to healthcare to different Internet of Things (IoT) settings (e.g., Internet of Vehicles – IoV). This is not surprising since FL supports data privacy by allowing clients to keep their data on their local machines (e.g., devices) and train the models locally. According to the report by Research [1], for example, the FL market is expected to grow at a compound annual growth rate (CAGR) of 10.7% from 2022 to 2030, with a projection of reaching USD 266.77 million by 2030. This growth is partly driven by the increasingly privacy-aware society. FL presents several compelling advantages. Notably, FL excels in safeguarding the privacy of data stored on IoT devices, a critical aspect in sensitive domains like finance and healthcare. Additionally, FL offers scalability by facilitating the distributed training of large-scale ML models. This decentralized approach often leads to accelerated training times and reduced communication costs compared to traditional centralized methods. Ultimately, FL emerges as a promising solution for upholding privacy in ML for IoT, contributing to enhanced efficiency and security in IoT systems [2].

Susceptibility of FL to model inversion (MI) Attacks: It is, however, known that conventional FL does not entirely prevent information leakage during the sharing of trained models with the server(s). In other words, these models could be vulnerable to privacy attacks, such as membership inference, generative adversarial network (GAN) reconstruction attacks, and MI attacks [3], [4], [5]. In the context of MI attacks, an attacker can exploit the parameter exchanges between clients and the server to infer the training data, even if the data is not directly shared. Thus, the leakage of sensitive training data has serious consequences. For instance, adversaries could (ab)use such information to facilitate nefarious activities (e.g., identity theft or financial fraud). Therefore, clients may be reluctant to participate in the FL process due to privacy leakage concerns. In the literature, there are three popular mitigation approaches, namely: cryptographic and differential privacy (DP) approaches [3], [6], [7], [8], and encoding-based techniques [9], [10], [11].

Limitations of cryptographic and DP Solutions: A number of FL-based approaches are known to be susceptible to MI attacks [5], [12]. In addition, commonly used cryptographic approaches (e.g., homomorphic encryption and secure multi-party computation – SMC) often involve extra encryption and decryption processes, which result in a substantial computation overhead. While DP ensures statistical privacy protection for individual records and guards against model inference attacks, the addition of noise during the training

Manuscript received 1 August 2023; revised 10 December 2023; accepted 13 February 2024. Date of publication 22 February 2024; date of current version 2 May 2024. This work was supported in part by University of New South Wales (UNSW) Canberra and in part by Australian Research Council's Discovery Early Career Researcher Award (DECRA) under Project DE230100116. The work of Kim-Kwang Raymond Choo was supported in part by the National Science Foundation (NSF) CREST under Grant HRD-1736209. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaojing Liao. *(Corresponding author: Kim-Kwang Raymond Choo.)*

Wael Issa and Nour Moustafa are with ADFA, University of New South Wales, Canberra, ACT 2610, Australia.

Benjamin Turnbull is with Australian Centre for Cyber Security, University of New South Wales, Campbell, ACT 2612, Australia.

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78258 USA (e-mail: raymond.choo@fulbrightmail.org).

Digital Object Identifier 10.1109/TIFS.2024.3368879

process often leads to the generation of less precise models (i.e., models with low utility) [6], [13], [14], [15].

Limitations of encoding-based approaches: Several existing encoding-based approaches and frameworks rely on training autoencoder models on the clients for feature encoding, followed by transmitting the encoded features to the server for centralized model training. However, transmitting the encoded features from clients to the server can lead to privacy risks as the encoded features may contain sensitive or private information, which could be exposed during transmission [9], [10]. Alternatively, other approaches focus on centrally training the autoencoder, overlooking the data heterogeneity and decentralized nature of IoT devices [16], [17]. Another set of approaches involves local training of the autoencoder and classifier, with the parameters of all three components (encoder, decoder, and classifier) being sent to the server for aggregation. However, such an approach significantly increases the communication cost [17], [18].

Our proposed approach: In light of these limitations, it is imperative to implement effective privacy-preserving FL approaches, along with adversarial settings. Thus, the proposed RVE-PFL introduces an effort to safeguard against MI attacks in FL by utilising a variational encoder and personalised federated learning (PFL). In contrast to other encoding-based approaches that rely on central training of the encoder on a server-side dataset or send subsequent feature extraction by clients, our RVE-PFL involves jointly training and fine-tuning both the encoder and classifier on each client's available data. To adhere to the principles of FL, we strictly exchange only the learned classifier parameters for model aggregation, rather than sharing labels or extracted features with the server. RVE-PFL incorporates personalised encoder locally on each client while sharing only the classifier parameters with the server. This ensures that the private data representation learned by the encoder remains confined to the client device, providing enhanced privacy protection and robustness against MI adversarial settings.

Key Contributions: We propose RVE-PFL as a robust variational encoder-based personalised FL approach that aims to address the risk of model inversion attacks while maintaining the global model's utility. A summary of this work is as follows:

- We propose RVE-PFL, which is a novel and simple approach that combines variational encoding with personalised FL. RVE-PFL consists of two primary components, namely: a personalised variational encoder and a classifier. The personalised variational encoder converts the client-private data into a probabilistic latent space, while the classifier is locally trained on the transformed data and then aggregated globally by the server. Therefore, the encoder and local classifier are simultaneously trained and fine-tuned in each global round.
- We use the MIFace [19], DLG [20], and iDLG [21] attacks to evaluate the resilience of RVE-PFL against MI attacks. The success of such attacks depends on whether the recovered data reveals sensitive information about a specific label. Therefore, we conducted a quantitative and qualitative analysis to investigate the attack's performance.
- We investigate the potential vulnerabilities of FL to model inversion attacks, which are launched by semi-honest or honest but curious adversaries.

Paper Structure: The remainder of this work is organized as follows. Section II offers background information on personalised FL and MI attacks, while Section III reviews the extant literature. Section IV outlines the proposed threat model, before introducing the proposed approach in Section V. The experimental setup and results are discussed in Section VI. Section VII outlines the conclusion and future directions for this work.

II. Preliminaries

This section aims to provide comprehensive insights into personalised FL and the potentially detrimental impact of model MI on FL.

A. Personalized Federated Learning (PFL)

PFL is a variant of FL that aims to train a model collaboratively while handling non-IID (Independent and Identically Distributed) data and improving privacy preservation. Multiple methodologies can be employed for implementing PFL, and one of them involves two distinct stages. In the first stage, a global model is learned collaboratively, whereas in the second stage, each device fine-tunes the global model to its local data to create a personalised model. In an alternative methodology, the model parameters are split into two sections: local parameters and global parameters. The first few layers constitute the local parameters, whereas the last few layers form the global parameters [22], [23], [24]. For example, the authors of [25] proposed a personalised FL approach, Local Global Federated Averaging (LG-FedAvg), that combines local representation learning and global federated training to address data heterogeneity and communication efficiency. The model parameters are split into local and global parameters, and devices train and update the whole model locally, but only the global parameters are communicated with the server for aggregation.

As per [22], the mechanism which divides model parameters between private and global is a deliberate choice made during architectural design. There are typically two approaches used in decoupling parameters for deep neural networks. The first is a "base layers + personalised layers" model. Here, personalised deep layers are kept private by clients for local training, enabling them to develop customized representations for specific tasks, while the base layers are shared with the FL server to learn generic, low-level features. The second approach involves creating personalised feature representations for each client. For instance, in [26], a bidirectional LSTM architecture document classification model is trained utilising FL by considering user embeddings as personal model parameters and character embeddings (i.e., LSTM and MLP layers) as global model parameters.

It is noteworthy that our approach adheres to the architecture-based PFL approaches, which are focused on

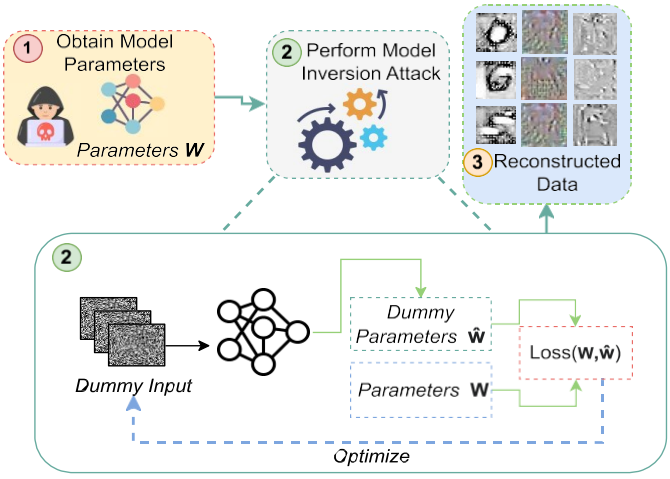


Fig. 1. Procedure of model inversion attack.

achieving personalisation via the second model decoupling approach mentioned above [22], [24]. To achieve this, our approach utilises the second decoupling approach, which involves the division of the model into two distinct components. The first component, referred to as the encoding part, consists of the personalised layers that are kept private by each client. This component is responsible for extracting and transforming features into a latent space. On the other hand, the second component is the FL classifier, which can be locally trained by the clients using the latent space and then shared with the server for global aggregation. By segregating these two components, we can achieve the desired level of personalisation while preserving the security and privacy of client data.

B. Model Inversion (MI) Attack

MI is widely acknowledged as one of the most powerful privacy attacks against the confidentiality of ML models [27], [28]. This attack aims to reconstruct the data that was utilised to train the model. Hence, the concept behind model inversion is that a learned model captures a mapping between the input and output spaces (a relationship between the input and output domains). This mapping can be used to make predictions in one direction (from an input sample to an output), but it can also be turned around to find an optimal input (reconstructed data) that minimizes the difference between the predicted value and the target response, such as a specific class label [29].

In the realm of FL, the inversion attack can be succinctly described as a three-step procedure, as depicted in Figure 1. In the initial step, the attacker procures either a local or global-trained model. Subsequently, the attacker employs a model inversion method (such as MIFAce [19]) that leverages gradient descent methods to optimize the local value of a loss function, iteratively adjusting the input until a more accurate solution is attained. Specifically, model parameter inversion attacks often require solving an optimization problem. Initially, the attacker retrieves the model parameters, denoted as W. Next, the attacker generates dummy samples, denoted as $(\hat{X}, \hat{y})$, and seeks to minimize the difference between the received parameters W and the dummy parameters $\hat{W}$.

The dummy parameters are computed by running the dummy samples through the target model using one forward-backward pass. During the optimization process, the values of the dummy samples are adjusted to better approximate the original training data. Particularly, the values of the dummy samples are tuned to approximate the training samples by the end of the attack [30]. Lastly, the attacker is capable of reconstructing data that is nearly indistinguishable from the original training data. This reconstructed data can then be utilised by the attacker to extract sensitive information about the participants involved in the FL. Instances of MI attacks that adhere to this methodology include Deep Leakage from Gradients (DLG) [20] and its enhanced version, improved Deep Leakage from Gradients (iDLG) [21].

DLG [20] constitutes an MI attack specifically tailored for revealing sensitive information within collaborative learning frameworks, exemplified by FL. DLG introduces an optimization algorithm that can infer both training inputs and labels through a limited number of iterations. The attack protocol initiates by randomly generating a set of "dummy" inputs and labels, followed by the execution of standard forward and backward passes. Diverging from conventional training procedures that optimize model weights, DLG uniquely directs its optimization towards dummy inputs and labels. This optimization process seeks to minimize the dissimilarity between gradients derived from the dummy data and those emanating from authentic training data. In the DLG, the authors generate synthetic data and associated labels by leveraging shared gradients. Despite this, DLG encounters challenges related to convergence and the consistent identification of ground-truth labels. Conversely, the improved DLG (iDLG) [21] method has observed that the sharing of gradients inadvertently discloses the actual labels. As a result, iDLG demonstrates the capability to reliably extract the ground-truth labels, distinguishing itself from DLG in this regard.

Specifically, Model inversion requires minimal knowledge from an adversary to be successful [19]. The most essential knowledge is understanding the model's output and knowledge of the model itself. The output knowledge enables the attacker to comprehend the expected outcome, and the knowledge of the model architecture and parameters is required to execute gradient descent for input optimization. Furthermore, knowledge of the data that was used to train the model can also be beneficial as it provides insight into the features and characteristics that the model is looking for in the input data, which can aid in the optimization process and enhance the accuracy of the reconstructed input. It is worth noting that the level of knowledge required may vary depending on the specific task and model being targeted.

III. Privacy Mechanisms in Federated Learning: Related Studies

In this section, we will explore relevant studies that have aimed to enhance privacy in FL.

A. Encryption and Differential Privacy-Based Approaches

A plethora of research has been proposed to address the issue of data leakage through MI attacks. Thus, ensuring the privacy of individual participants' data is of paramount importance in FL, where multiple parties work together to train a global model. Not all participants may be fully trusted, and thus it is essential to implement countermeasures to prevent malicious actors from accessing and stealing sensitive information. Therefore, it is important to use advanced techniques to protect the privacy of participants [31]. Thus, Truex et al. [32] presented a hybrid approach to privacy-preserving FL that combines DP and SMC to protect against inference threats.

Huang et al. introduced InstaHide [33], a simple encryption technique designed specifically for training images in distributed deep learning frameworks. InstaHide seamlessly integrates into existing systems and utilises a "one-time secret key" to encrypt each training image. The encryption process involves a combination of the target image with randomly selected images and the application of a random pixel-wise mask. To mitigate the risk of MI attacks, Madi et al. [34] employed a method incorporating Homomorphic Encryption (HE) and Verifiable Computing (VC) techniques. This involves conducting the federated averaging operation directly within the encrypted domain using HE while ensuring the correctness of the operation through formal proofs enabled by VC. Similarly, Triastcyn and Faltings [8] proposed an approach utilising a combination of Bayesian differential privacy and encryption techniques to achieve privacy preservation in FL.

In [35], Xu et al. introduced HybridAlpha, a method for privacy-preserving FL. To prevent model inversion attacks, the method employs an SMC protocol that utilises functional encryption. While homomorphic encryption and SMC are commonly used cryptographic techniques, they require additional encryption and decryption operations, leading to a notable increase in computational workload. Zhang et al. [36] presented a Privacy-Enhanced Momentum FL (PEMFL) method to protect sensitive data in industrial cyber-physical systems using DP and chaos-based encryption. However, the PEMFL method has some shortcomings, including reduced accuracy due to the addition of noise and increased computational complexity from the use of a chaotic system.

In a recent study, Wei et al. [6] presented a novel framework that leverages the principles of DP to enhance privacy in FL. This framework, named Noising before Model Aggregation FL (NbAFL), involves introducing artificial noise to the parameters at the clients' side before aggregation. The authors highlight that this approach introduces a tradeoff between the level of privacy protection and the convergence performance of the FL process. Similarly, Wei et al. [37] introduced a novel approach called Fed-CDP, which focuses on preserving privacy in FL by incorporating per-training example-based client DP. They also conducted a rigorous analysis of Fed-CDP, establishing its $(\epsilon, \delta)-$DP guarantee. A formal comparison was made between Fed-CDP and server-coordinated DP approach codenamed as Fed-SDP regarding privacy accounting. According to the authors, Fed-CDP incorporates a dynamic decay noise-injection policy, which contributes to enhancing the accuracy and resilience of the approach.

In their research [38], Wei et al. proposed the Fed-αCDP approach, which enhances traditional per-client methods by incorporating per-example gradient perturbation and adaptive parameter optimizations. Firstly, Fed-αCDP introduces DP-controlled noise to per-example gradients during local training at the client level. Secondly, Fed-αCDP determines the sensitivity of a differentially private FL algorithm using the l_2 max of gradients, resulting in reduced Gaussian noise variance during local Stochastic Gradient Descent (SGD) at the clients' training stage. Lastly, Fed-αCDP implements a dynamic decaying noise scale, σ , to align Gaussian noise variance and noise injection with the trend of gradient updates during local SGD.

Despite the effectiveness of DP in protecting against MI attacks, its use can significantly impact the model's learning ability. This is because achieving privacy often requires high levels of noise, which can impede the model's capacity to learn from the data. These issues raise important questions about the trade-offs between privacy and model utility and whether there exist alternative approaches that can provide similar protection without compromising the model's learning ability. Moreover, further research is necessary to fully understand the utility of DP as a defense mechanism beyond privacy preservation and explore other potential solutions that can protect the model against privacy leakage attacks while preserving its learning ability [3], [39].

B. Encoding-Based Approaches

Numerous investigations have been undertaken to tackle privacy-related issues within the realm of deep and federated learning by employing autoencoders (AE). Zhang et al. [9] proposed a framework to safeguard the privacy of data owners through the distribution of the machine learning process between clients and a central server. Within this framework, the clients undertake the responsibility of encoding the original data using their respective autoencoder. Subsequently, the encoded data, along with corresponding labels, is transmitted to the server for centralized model training. During the inference stage, clients extract features using their individual encoders and transmit them to the central server for classification.

Keshk et al. [10] introduced a novel framework that utilises the combined potential of blockchain technology and deep learning to enhance the privacy and security of smart power networks. The framework leverages a VAE to transform data into an encoded format, thereby mitigating the risks associated with inference attacks. However, it is worth noting that this approach has not been tested against model inversion attacks and does not adequately account for the data heterogeneity exhibited by IoT devices. Jiang et al. [16] presented DP-Fed-WAE, a privacy-preserving framework specifically designed for the collection of high-dimensional categorical data. The proposed framework involves training a local autoencoder to learn the representation of the data, followed by the application of DP to perturb the autoencoder's parameters.

In the context of privacy-preserving analysis of big data, Alguliyev et al. [17] introduced a deep learning approach. The primary objective of this approach is to convert the sensitive

portion of personal information into non-sensitive data. Ma et al. [18] proposed an innovative approach to enhance privacy protection in collaborative learning, specifically addressing the challenges posed by gradient-based reconstruction attacks. The proposed scheme incorporates an initial permutation step that applies a scalable block size transformation to the sensitive training images. He et al. [40] introduced a defense mechanism utilizing Dropout, wherein random neurons are deactivated during inference to hinder adversaries from accurately reconstructing original images from intermediate values. While this defense mitigates the impact of model inversion attacks during the inference stage, it does not protect during FL training, where model parameters are exchanged between participants and the server.

In another related study, Li et al. [41] introduced Resistance Split FL (ResSFL), a two-step framework comprising a pre-training step to establish a feature extractor with MI resistance and a subsequent resistance transfer step, where the resilient feature extractor initializes the client-side model in the SFL scheme. While ResSFL mitigates the impact of MI attacks, it deviates from FL principles as it relies on a centralized dataset, posing privacy concerns. Additionally, the pretraining step introduces additional computational overhead. In contrast to the previous approach that relies on central training of the encoder on a server-side dataset and subsequent feature extraction by clients, our approach adopts a distinct architecture. We adopt a client-centric training paradigm where both the encoder and classifier are jointly trained and fine-tuned on each client's available data. Additionally, instead of directly sharing labels or extracted features with the server, we strictly adhere to the principles of FL by only exchanging the learned classifier parameters for model aggregation. Our approach incorporates the concept of PFL by retaining the encoder locally on each client while exclusively sharing the classifier component with the server. This means that the private data representation learned by the encoder remains within the confines of the client device, enhancing privacy protection.

IV. Threat Model

The vulnerability of clients to inversion attacks constitutes a fundamental issue in FL systems. This susceptibility emanates from requiring clients to transmit their locally trained parameters to the server as a precondition for the global model aggregation. Such transmission makes clients' data vulnerable to privacy violations and security breaches, underscoring the importance of implementing robust security measures to protect the integrity of FL and safeguard client data [12].

In this research, we investigate the potential vulnerabilities of FL systems to MI attacks, which are launched by semi-honest or honest-but-curious adversaries. These adversaries, who may be clients or servers within the FL system or external passive attackers, attempt to extract sensitive information about the training data that is not intended to be shared. We specifically focus on the threat posed by these adversaries, who, while adhering to the FL protocol, seek to gather unauthorized information through the use of MI techniques. Furthermore, we consider scenarios where the attacker has some knowledge of the FL system, referred to as gray-box [29] settings, which can make the attack more successful and challenging to defend against.

In the gray-box attack setting, we assume that the adversary has a significant level of knowledge about the classifier being used. This includes knowledge of the structure of the classifier and access to the model parameters, which can be obtained through the coordinating server or from the FL participants. It is noteworthy that the structure of the model can be infringed by the model-stealing attack [42]; however, we assume that the attacker has this knowledge without performing such attacks. Furthermore, the attackers are limited in their knowledge in that they do not have access to the training data used to train the classifier, except for the shape of the input.

This type of attack is considered more challenging for the defender as the attacker has more information about the classifier than in a black-box attack but less than in a white-box attack. Additionally, it is important to mention that gray-box attacks can be used to evaluate the robustness and the vulnerability of the classifier in the case of an insider attacker or a side-channel attacker. Regardless of whether the adversary performing the model inversion attack is an honest and curious aggregation server, participant, or external adversary, the goal remains the same: to infer the corresponding training data from the local or global model, leading to privacy leaks in FL.

Specifically, MI attacks can occur and obtain the trained model parameters through four entry points. The first involves the adversary compromising the FL participants and obtaining a copy of their locally trained classifier. Additionally, the MI attack can also be performed by the FL participant itself if it is malicious. The second and third involve the adversary compromising and intercepting the communication between the edge server and the FL participating devices to obtain a copy of the local or global classifier. Finally, the fourth entry point involves the MI attack being performed by a malicious edge server or by an external adversary who has compromised and gained access to the edge server.

V. Proposed Approach (RVE-PFL)

This section outlines the details of RVE-PFL, a variational encoder-based personalised FL approach, designed to defend against MI attacks. The description includes the RVE-PFL architecture, and the FL algorithm illustrating how clients communicate with the edge server. The RVE-PFL approach is reliant on personalised federated learning (PFL) and the variational encoder. We highlight the pivotal roles of these elements as essential milestones in the development of the RVE-PF approach. Additionally, we present an overview of the PFL technique that we utilised in designing the RVE-PF approach.

Our emphasis is on a personalised approach to federated learning, wherein a group of intelligent IoT devices, also known as clients or participants, continuously monitor the physical environment and store the gathered data in their respective databases. These devices have inherent computational capabilities that allow them to train local models. An edge server also takes charge of coordinating the collaboration among the devices, improving their models by utilising data

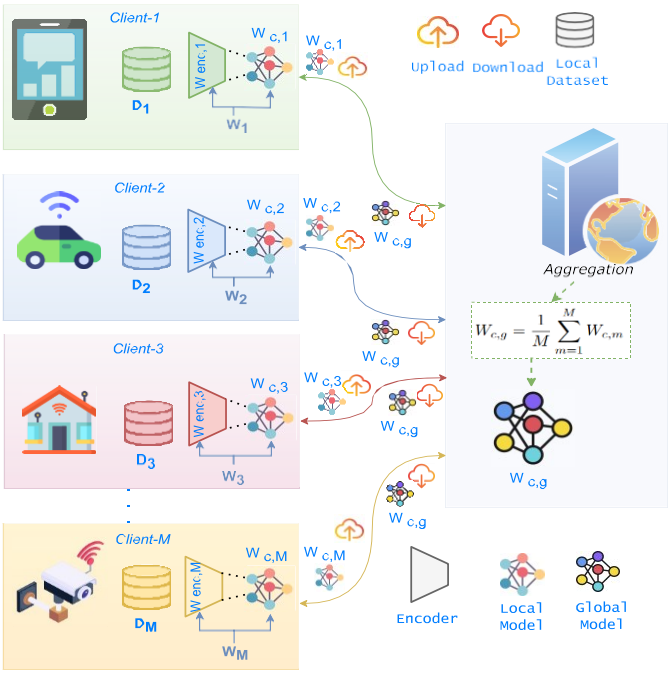


Fig. 2. Overview of proposed FL approach (RVE-PFL).

from other devices through the global model aggregation method, all while maintaining a reasonable level of privacy preservation and preserving the model's utility. RVE-PFL ensures that it becomes challenging for adversaries to execute model inversion attacks against the exchanged local models between the clients and server, or even against the global model, as we will explain in the next paragraphs.

RVE-PFL also ensures the confidentiality of the data and preserves the privacy of the participants against adversarial settings (i.e., MI attacks). To do so, we consider M devices participating in FL, where each device possesses its local training dataset $D_m = (X_m, y_m)$, with X_m representing a feature space and y_m denoting the corresponding label vector. RVE-PFL divides the model with parameters W_m into two dependent components. The first component represents the personalised variational encoder with parameters $W_{enc,m}$ that is trained locally and is kept by the clients. The second component of the model is the classifier with parameters $W_{c,m}$ and it sends by the clients to the edger server for aggregation, where:

$$W_m = W_{enc,m} \cup W_{c,m} \tag{1}$$

The architecture of RVE-PFL is illustrated in Figure 2. The model is divided into a personalised variational encoder and a shared classifier for global aggregation and experience exchange among participating clients. This design aims to balance privacy preservation and model utility, ensuring that the model achieves reasonable performance while preserving the privacy of the client's data. The edge server initialises the model parameters and sends them to the clients to start the local training. Every client receives the model parameter W and starts to train its local model with parameters W_m on its own data using their objective function f_m. This was achieved by minimising the following optimisation problem:

$$W_m^* = \arg\min_{W_m} f_m(X_m, y_m, W_m), \tag{2}$$

where $f_m(X_m, y_m, W_m)$ is the local objective function of client m that should be minimised to find the best model parameters W_m^* given dataset D_m with (X_m, y_m). The superscript $*$ denotes that we are looking for the optimal value of W_m that minimises the objective function f_m.

The local parameters W_m of client m are divided into two components; the first one is for the personalised variational encoder parameter and the second is for the classifier. The encoder network produces two sets of output parameters: the mean μ and the standard deviation σ . These parameters are used to define a Gaussian distribution over the latent variables z [43], [44], as follows:

$$z \sim N(\mu, \sigma^2) \tag{3}$$

To generate a sample from this distribution, we use the reparameterization trick:

$$z = \mu + \sigma * \epsilon, \tag{4}$$

where ϵ is a random sample from a standard Gaussian distribution, it is worth noting that the variational encoder is employed in our approach to preserving privacy. It achieves this by mapping input data to a probabilistic latent space, which effectively conceals any sensitive information. Rather than creating a deterministic mapping from input data to latent space, probabilistic mapping is used, which allows the encoder to provide a natural means of privacy protection.

In RVE-PFL, a different approach is considered to use the variational encoder for privacy preservation. Instead of mapping the input data into a low-dimensional probabilistic latent space, the proposed encoder maintains the same dimensions as the input. This means that the latent space can be used directly as input for the classifier without any information loss resulting from compression. By maintaining the same dimensions in the latent space as the input data, we can preserve all of the original information while still achieving a more compact representation of the data. This allows us to more effectively preserve privacy while maintaining the accuracy of the subsequent classifier. Thus, the variational encoder acts as a feature transformation technique that allows us to create a more compact representation of the input data while still preserving privacy.

During local training, both the variational encoder and the classifier are trained as a single model. Once the training is complete, the clients keep their respective trained variational encoders $W_{enc,m}$, which serve as personalised feature transformers. Meanwhile, the trained parameters of the classifier $W_{c,m}$ are sent to the edge server for global aggregation. This approach ensures that the shared classifier is trained on the probabilistic latent space rather than the original data, making it challenging for adversaries to perform an MI attack. This is due to the probabilistic latent space that is non-deterministic and thus not easy to reconstruct. By using this approach, we can preserve the privacy of the individual clients' data while still being able to train a robust classifier.

After the clients complete the local training, they upload the classifier parameters $W_{c,m}$ to the edge server. Then, the local classifiers $W_{c,m}$ are combined into the global classifier $W_{c,g}$ using Federated Averaging (FedAvg) [45]. This can be done using the following equation:

$$W_{c,g} = \frac{\sum_{m=1}^{M} n_m W_{c,m}}{\sum_{m=1}^{M} n_m}, \tag{5}$$

where M is the total number of clients, n_m is the number of samples used for training by client m, and $W_{c,m}$ is the local classifier of client m. FedAvg [45] is a popular FL algorithm that allows multiple clients to collaboratively train a machine learning model without sharing their data with a central server. In this algorithm, each client trains a local model on its own data and sends the model updates to the server. The server aggregates the model updates using FedAvg to generate a global model that is sent back to the clients for further training. By using FedAvg, the clients can collectively learn from each other's data without compromising their privacy to some extent. In the proposed approach, since all clients have the same number of data samples in this scenario, the weighting term $\frac{n_m}{\sum_{m=1}^{M} n_m}$ is the same for all clients, and the FedAvg equation reduces to a simple average as follows:

$$W_{c,g} = \frac{1}{M} \sum_{m=1}^{M} W_{c,m} \tag{6}$$

The outcome is that the global model becomes the arithmetic mean of the local models of all clients. This approach ensures that each client has an equal contribution to the global model, regardless of the number of data samples it has.

Algorithm 1 RVE-PFL - Server Side

Input: M: the number of clients, Wc, m: the classifier models from the clients $m \in 1, 2, \ldots, M$, R: the number of communication rounds
Output: $W_{c,g}$: federated trained global classifier model
Edge server do:
  Initialize: the full model parameters W_0
  Send: the initial full model parameters W_0 to the clients $m \in 1, 2, \ldots, M$
  for r *in* $1, \ldots, R$ **do**
    Receive: the local classifier models updates Wc, m from the clients $m \in 1, 2, \ldots, M$
    Aggregate: $W_{c,g} = \frac{1}{M} \sum_{m=1}^{m} W_{c,m}$.
    Send: the aggregated global classifier model (federated trained global classifier model) $W_{c,g}$ to the clients $1, 2, \ldots, M$.
  end
end

More specifically, the proposed approach is comprised of two distinct sides, which are elucidated in Algorithms 1 and 2. The server-side is presented in Algorithm 1, while the client-side is elaborated upon in Algorithm 2. From the perspective of the server side, the edge server initializes the full model parameters W_0, and broadcasts them to all clients. In each communication round, the server receives the local classifier model updates $W_{c,m}$ from each client. These updates are aggregated by computing the average of the local models from all clients, resulting in the federated trained global classifier model $W_{c,g}$ as in Equation 6. The server then sends the aggregated model $W_{c,g}$ back to all clients to start a new global training round. This process is repeated until the number of communication rounds reaches R.

Algorithm 2 RVE-PFL - Client Side

Input: η learning rate, L: batch size, E: local epochs, W_0: the initial full model parameters, and D_m: local dataset, r: current global round index
Output: $W_{c,m}$: the local classifier model updates and W_m: the full model including the personalised encoder $W_{enc,m}$ and the classifier model $W_{c,m}$
if $r == 0$ **then**
  Download: the initial full model parameters W_0
  Set: $W_m \leftarrow W_0$
end
else
  Download: the global classifier model parameters $W_{c,g}$
  Split: W_m into $W_{enc,m}$ and $W_{c,m}$
  Set: $W_{c,m} \leftarrow W_{c,g}$
  Set: $W_m \leftarrow W_{enc,m} \cup W_{c,g}$
end
Calculate: number of batches $B \leftarrow |D_m|/L$.
for e *in* $1, \ldots, E$ **do**
  Set: $W_{m,0} \leftarrow W_m$
  Compute number of batches: $B \leftarrow \lceil |D_m|/L \rceil$
  for b *in* $1, \ldots, B$ **do**
    Compute local loss $L_{m,b}(X_m, y_m, W_{m,b})$
    Compute local gradient $g_{m,b} \leftarrow \nabla_{W_{m,b}} L_{m,b}(X_m, y_m, W_{m,b})$
    Update local model weights: $W_{m,b} \leftarrow W_{m,b-1} - \eta \frac{1}{B} g_{m,b}$
    Set: $W_m \leftarrow W_{m,b}$
  end
end
Split: W_m into $W_{enc,m}$ and $W_{c,m}$
Keep $W_{enc,m}$ as a personalised feature transformation.
Send $W_{c,m}$ to the edge server for aggregation.

On the client side, each client begins the collaborative training by checking if the global round index is zero. If it is, each client downloads the initial full model parameters and sets the full model W_m to be equal to these parameters. Otherwise, the client downloads the global classifier model parameters $W_{c,g}$ and splits the full model W_m into the personalised encoder $W_{enc,m}$ and the classifier model $W_{c,m}$. The local classifier model parameters $W_{c,m}$ is then updated with the global classifier model $W_{c,g}$, while the personalised encoder parameters are kept on the client side. Then, the full model W_m is set to be equal to the union of personalised encoder

$W_{enc,m}$ and the global classifier model $W_{c,g}$ to start a new collaborative training round (Lines 3-12 Algorithm 2).

Next, each client calculates the number of batches B in the local dataset D_m based on the given batch size L. Accordingly, each client proceeds to update the local model weights for each epoch and batch using SGD with a fixed learning rate η. For each epoch, the clients set the initial model weights $W_{m,0}$ to be equal to the current full model weights W_m and compute the number of batches B. For each batch, the algorithm calculates the local loss and gradient and then updates the local model weights accordingly (Lines 18-21, Algorithm 2). Then, the current full model parameters W_m are then set to be equal to the updated local model parameters $W_{m,b}$.

Finally, the client splits the full model W_m into the personalised encoder $W_{enc,m}$ and the classifier model $W_{c,m}$. The personalised encoder $W_{enc,m}$ is kept as a personalised feature transformation on the client side, while the classifier model $W_{c,m}$ is sent to the edge server for aggregation with the updates from other clients in the FL system.

VI. Experimental Settings and Evaluations

This section presents a comprehensive overview of the experimental setup, discussing how the experimental results evaluate the privacy preservation and utility of RVE-PFL.

A. Experimental Settings

1) Datasets: Three well-known datasets are used in the experiments (i.e., MNIST, Fashion-MNIST, and CIFAR-10), which have become standard benchmarks in the field of machine learning research. The MNIST dataset is composed of 60,000 training images and 10,000 test images of handwritten digits (0-(9), each of which is a grayscale image of size 28 × 28 pixels. On the other hand, the Fashion-MNIST dataset was developed to provide a more challenging alternative to MNIST, containing 60,000 training images and 10,000 test images of 10 different clothing items such as T-shirts, dresses, and shoes, with each image being a grayscale image of size 28×28 pixels. Lastly, the CIFAR-10 dataset contains 60,000 32 × 32 color images of 10 different classes, including airplanes, cars, and cats, with 50,000 training images and 10,000 test images. This dataset poses a more significant challenge than the former two datasets, as it necessitates models to recognize and classify images with more intricate features and colors.

2) Data Partitioning: We randomly combine the training and test data and then randomly split it equally among the ten clients participating in FL. In this manner, we ensure that each client has a representative sample of the data that does not overlap with any of the other clients. We then split each client's local data into a training set and a test set. The training set is used to train the local model, while the test set is used to evaluate the performance of the global model. This split is performed by allocating a certain percentage of the data (typically 90%) to the training set and the remaining data to the test set (typically 10% from the client's local data. Hence, the test set of each client is employed to assess the performance of the global model, and subsequently calculate the average across all clients in FL.

3) Computational Settings: In this research, the experiments were carried out on a computing system consisting of an AMD® Ryzen Threadripper 1950 × 16-core processor, along with an NVIDIA Titan (X, Xp), and GeForce GTX 1080 Ti GPU. The system is also equipped with 128 GB of RAM, providing significant computational resources. The experiments were conducted on the Ubuntu 22.04.1 LTS operating system, ensuring a stable and reliable environment for testing and analysis. In addition, the implementation of the models and data preprocessing was carried out using popular deep learning frameworks including TensorFlow, Keras, and PyTorch.

4) Variational Encoder: The Convolutional Variational Encoder (CVE) model is made up of seven layers. The first two layers are convolutional layers with 32 and 64 filters, respectively. This is followed by a fully connected layer with 512 nodes and a dropout layer with a 25% dropout rate. The next two layers are also fully connected with the same flattened input size. These layers are used to create a latent space with the same dimensions as the input data using a reparameterization trick layer. The purpose of generating a latent space with the same dimensions as the input is to prevent information loss resulting from a low latent space size. Thus, the CVE model is personalised and is kept by each client to transform their local data into the randomized latent space. The randomized latent space is then reshaped to create an input for the classifier. This classifier is trained collaboratively as a global model by all participants.

5) Classifier: The classifier model consists of seven layers. The first layer is a convolutional layer with 32 filters, followed by a max-pooling layer with a stride of 2. The third layer is also a convolutional layer with 64 filters, followed by another max-pooling layer with a stride of 2. The fifth layer is a fully connected layer with 256 nodes, which is followed by a dropout layer with a dropout percentage of 25%. The last layer is a softmax layer with nodes equal to the number of classes, which is responsible for producing the final predictions.

6) Baselines: RVE-PFL approach was evaluated and compared against four baselines to demonstrate its effectiveness, privacy preservation, and generalization. The first baseline was the traditional FL method (FedAVG), where the server distributes the global model to clients for local training and then aggregates the locally updated models. This process is repeated until a certain number of global rounds are completed. The second baseline involved the use of NbAFL, a popular technique for privacy preservation in FL. The third and fourth baselines, namely Fed-CDP and Fed-αCDP, address privacy preservation in FL by integrating client DP based on per-training examples. Our experiments with NbAFL, Fed-CDP, and Fed-αCDP aimed to highlight the robustness of the RVE-PFL approach compared to DP, which is a well-known privacy-preserving technique in the FL literature.

7) Hyperparameters: In our experiments, we set the number of FL clients, M , to 10. The total number of global rounds, R, is set to 100, and the number of local epochs, E, is set to 10. We use a batch size of L of 256 and a learning rate η of 0.001. For the optimization algorithms, we use the Adam optimizer with $1e^{-3}$ weight decay and epsilon $\epsilon = 1e^{-7}$ to

train the RVE-PFL approach. For FedAvg, we also use the Adam optimizer but without epsilon $\epsilon = 1e^{-7}$. For NbAFL, we use the DPSGD optimizer supported by the TensorFlow Privacy framework. We set the L_2 norm clipping to be 1, the number of micro-batches to be 1, and the noise multiplier to be 1.1. For Fed-CDP and Fed-αCDP, we use the DPSGD optimizer as well and we set the L_2 norm clipping to be 1, the number of micro-batches to be 1, the noise multiplier to be 0.5, the batch size 1, the gradient accumulation steps to be 64, and the learning rate to be 0.05.

B. Experimental Evaluation and Analysis

1) Evaluation Metrics for Privacy Leakage: The performance of RVE-PFL for preserving data privacy in FL is evaluated using the following metrics:

a) Mean squared error (MSE): MSE measures the pixel-wise squared error between the original image x and the reconstructed image x'. When both x and x' are of size $M \times N$, the MSE can be denoted as:

$$MSE(x, x') = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (x_{ij} - x'_{ij})^2 \quad (7)$$

A lower value of MSE between the original x and the reconstructed image x' indicates a higher degree of similarity between the two. This implies that the reconstructed image has a smaller difference from the original image.

b) PSNR (peak signal-to-noise ratio): PSNR is a widely used image quality metric that measures the quality of a reconstructed or compressed image compared to the original image.

$$PSNR(x, x') = 10 \log_{10} \left\{ \frac{max(x)^2}{MSE(x, x')} \right\}, \quad (8)$$

where x is the original image, x' is the reconstructed or compressed image, $max(x)$ is the maximum possible pixel value, and $MSE(x, x')$ is the mean squared error between the original and reconstructed or compressed images. The $PSNR$ formula essentially calculates the ratio between the maximum possible pixel value and the mean squared error of the two images. A higher $PSNR$ value indicates that the reconstructed image is closer to the original image in terms of quality, while a lower $PSNR$ value indicates a larger difference between the two images.

c) Structural similarity (SSIM): $SSIM$ is a perception-based metric that measures the similarity between two images by comparing their luminance, contrast, and structure. It ranges from 0 to 1, with a value of 1 indicating perfect similarity. The $SSIM$ index can be calculated as:

$$SSIM(x, x') = \frac{(2\mu_x \mu_{x'} + c_1)(2\sigma_{xx'} + c_2)}{(\mu_x^2 + \mu_{x'}^2 + c_1)(\sigma_x^2 + \sigma_{x'}^2 + c_2)}, \quad (9)$$

where μ_x and $\mu_{x'}$ are the means of x and x', respectively, σ_x^2 and $\sigma_{x'}^2$ are their variances, $\sigma_{xx'}$ is their covariance, and c_1 and c_2 are constants that stabilize the division by weak denominator.

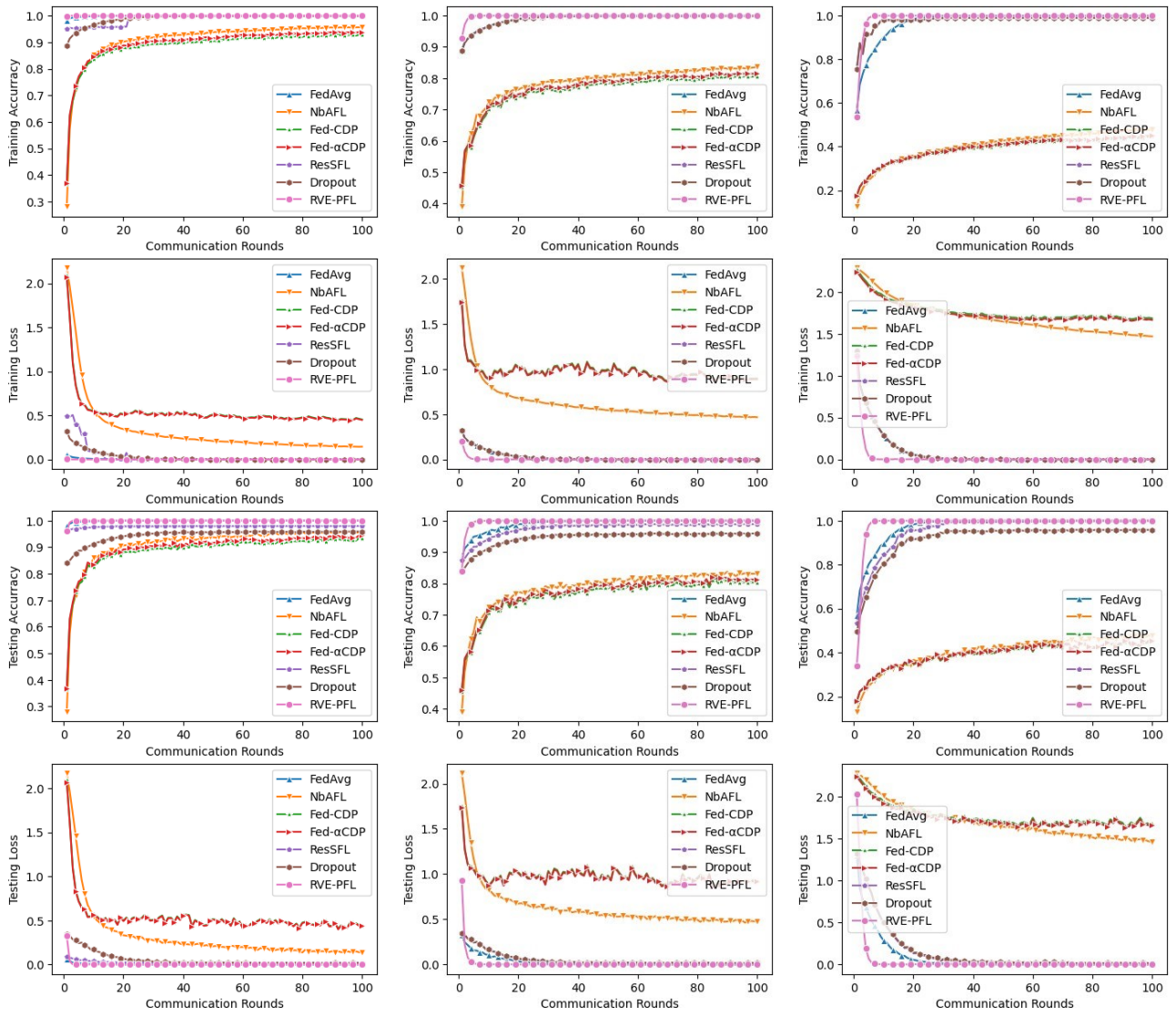
d) FID (fréchet inception distance): FID is a popular image quality metric that measures the similarity between the distribution of real and reconstructed images. The FID metric is calculated based on the statistics of the features extracted by the Inception$v3$ neural network trained on the ImageNet dataset. Given two sets of images, the real images x and the reconstructed images x', the FID distance is defined as:

$$FID(x, x') = ||\mu_x - \mu_{x'}||^2 + Tr(\Sigma_x + \Sigma_{x'} - 2(\Sigma_x \Sigma_{x'})^{1/2}), \quad (10)$$

where μ_x and $\mu_{x'}$ are the mean feature vectors of the real and reconstructed images, respectively, and Σ_x and $\Sigma_{x'}$ are their covariance matrices. Tr denotes the trace operator, and $||.||$ denotes the Frobenius norm. A lower FID value indicates a higher degree of similarity between the two distributions, implying that the reconstructed images are closer to the real visual quality.

2) Experimental Results and Discussion: Initially, we conducted a performance evaluation of traditional FL (FedAvg) and (NbAFL, Fed-CDP, Fed-αCDP, ResSFL, and Dropout), that utilised as defense mechanisms against model inference attacks, including MI attacks. Subsequently, we conducted an experimental verification of the effectiveness of the RVE-PFL approach in defending against model inversion attacks in FL, as compared to the state-of-the-art approaches in the literature. Specifically, we assessed the effectiveness of RVE-PFL in two aspects: privacy preservation, as determined by the degree of similarity between the reconstructed image and the original image, and model utility, as determined by the impact of utilising DP, feature transfer (ResSFL), Dropout, and our approach (as a defense against model inversion attacks) on the model's performance. Our experimental results demonstrate our approach's effectiveness in improving privacy preservation and model utility in FL scenarios.

a) Performance and convergence: Tables I, II and Figure 3 show the performance evaluation results of RVE-PFL, FedAvg, NbAFL, Fed-CDP, Fed-αCDP, ResSFL, and Dropout. Notably, our approach achieves performance that is comparable to FedAvg, while also surpassing NbAFL, Fed-CDP, Fed-αCDP, ResSFL, and Dropout in terms of accuracy, precision, recall, and F1-score on both training and test data. To report our results, we conducted experiments on the three datasets for 100 communication rounds. At the end of each communication round, we evaluated the global model on the local test data of each participating client and calculated the average performance across all clients. We then reported the final results by computing the overall average and standard deviation for the 100 rounds. This was necessary as performance results may vary from one round to another, and we needed to express the model's overall performance across all training rounds. We applied this measurement criterion to both the training and test datasets and reported the performance evaluation results in Tables I, II. For example, as indicated in Table I, our approach achieves an accuracy of 0.99 ± 0.053 on the Cifar-10 data set. This means that the model's estimated accuracy is 99% with a margin of error of 0.053.



(a) MNIST (b) Fashion-MNIST (c) Cifar-10

Fig. 3. Performance comparison on training and test data.

According to the observations from Figure 3, RVE-PFL exhibits similar convergence behavior to that of the standard FL (FedAvg), as the curves of FedAvg and RVE-PFL match after a small number of global rounds, achieving a reasonable performance compared to NbAFL, Fed-CDP, and Fed-αCDP. This is indicated by the fact that RVE-PFL converges to a global model that attains high accuracy on the test data during the early rounds of communication. Moreover, we observed that RVE-PFL (represented by the purple colour in Figure 3) exhibits an earlier convergence to a reasonable level of accuracy compared to FedAvg. This experimental evidence supports the effectiveness of RVE-PFL on both the training and test datasets. By contrast, NbAFL, Fed-CDP, and Fed-αCDP demonstrate poor convergence behavior and fail to achieve satisfactory performance results, which leads to a degradation in the utility of the global model. However, we add a moderate amount of noise with a noise multiplier of 1.1 and 0.5, which means that if we add more noise, the NbAFL, Fed-CDP, and Fed-αCDP will converge more slowly and be less effective. This conclusion is consistent with the findings of other studies [15], [31], [46] that indicate that the addition of DP during model training results in a trade-off between utility and privacy, which consequently negatively impacts the global model's performance. Thus, we suggest RVE-PFL as a solution to enable the development of FL with reasonable utility and privacy levels.

The ResSFL and Dropout exhibit comparable performance to RVE-PFL during training. However, in the testing phase, RVE-PFL outperforms ResFL and Dropout by 2% and 5% on the MNIST and Fashion-MNIST datasets, and by 3% and 7% on the CIFAR-10 dataset, respectively. Despite Dropout incurring no additional computational cost compared to ResSFL and RVE-PFL, it provides a lower level of privacy than our approach. Moreover, Dropout may only mitigate privacy leakage during the inference stage.

TABLE I
PERFORMANCE EVALUATION ON TRAINING DATA

Dataset	Approach	Metrics			
		Accuracy	Precision	Recall	F1-score
MNIST	FedAvg [45]	0.99 ± 0.002	0.99 ± 0.002	0.99 ± 0.002	0.99 ± 0.002
	NbAFL [6]	0.91 ± 0.087	0.91 ± 0.078	0.91 ± 0.088	0.91 ± 0.092
	Fed-CDP [37]	0.89 ± 0.072	0.89 ± 0.061	0.88 ± 0.074	0.88 ± 0.081
	Fed-αCDP [38]	0.90 ± 0.071	0.90 ± 0.060	0.89 ± 0.073	0.89 ± 0.080
	ResSFL [41]	0.99 ± 0.003	0.99 ± 0.003	0.99 ± 0.003	0.99 ± 0.003
	Dropout [40]	0.99 ± 0.001	0.99 ± 0.001	0.99 ± 0.001	0.99 ± 0.001
	RVE-PFL	**0.99 ± 0.001**	**0.99 ± 0.001**	**0.99 ± 0.001**	**0.99 ± 0.001**
FMNIST	FedAvg [45]	0.99 ± 0.020	0.99 ± 0.02	0.99 ± 0.020	0.99 ± 0.020
	NbAFL [6]	0.79 ± 0.066	0.78 ± 0.063	0.78 ± 0.066	0.78 ± 0.078
	Fed-CDP [37]	0.76 ± 0.060	0.76 ± 0.050	0.76 ± 0.060	0.75 ± 0.073
	Fed-αCDP [38]	0.77 ± 0.059	0.77 ± 0.049	0.77 ± 0.059	0.75 ± 0.072
	ResSFL [41]	0.99 ± 0.010	0.99 ± 0.01	0.99 ± 0.012	0.99 ± 0.013
	Dropout [40]	0.99 ± 0.031	0.99 ± 0.030	0.99 ± 0.031	0.99 ± 0.030
	RVE-PFL	**0.99 ± 0.008**	**0.99 ± 0.007**	**0.99 ±0.008**	**0.99 ± 0.008**
Cifar-10	FedAvg [45]	0.97 ± 0.071	0.97 ± 0.069	0.97 ± 0.071	0.97 ± 0.071
	NbAFL [6]	0.41 ± 0.070	0.40 ± 0.064	0.41 ± 0.070	0.39 ± 0.077
	Fed-CDP [37]	0.39 ± 0.056	0.39 ± 0.053	0.39 ± 0.056	0.38 ± 0.059
	Fed-αCDP [38]	0.39 ± 0.055	0.39 ± 0.052	0.39 ± 0.055	0.38 ± 0.058
	ResSFL [41]	0.98 ± 0.06	0.98 ± 0.059	0.98 ± 0.057	0.98 ± 0.052
	Dropout [40]	0.98 ± 0.062	0.98 ± 0.063	0.98 ± 0.062	0.98 ± 0.062
	RVE-PFL	**0.99 ± 0.053**	**0.99 ± 0.052**	**0.99 ± 0.053**	**0.99 ± 0.055**

TABLE II
PERFORMANCE EVALUATION ON TEST DATA

Dataset	Approach	Performance Metrics			
		Accuracy	Precision	Recall	F1-score
MNIST	FedAvg [45]	0.99 ± 0.002	0.99 ± 0.003	0.99 ± 0.002	0.99 ± 0.003
	NbAFL [6]	0.91 ± 0.087	0.91 ± 0.077	0.91 ± 0.088	0.91 ± 0.093
	Fed-CDP [37]	0.89 ± 0.073	0.89 ± 0.063	0.89 ± 0.075	0.88 ± 0.082
	Fed-αCDP [38]	0.90 ± 0.072	0.90 ± 0.062	0.90 ± 0.074	0.89 ± 0.081
	ResSFL [41]	0.97 ± 0.076	0.97 ± 0.074	0.97 ± 0.074	0.97 ± 0.075
	Dropout [40]	0.94 ± 0.052	0.94 ± 0.051	0.94 ± 0.052	0.94 ± 0.053
	RVE-PFL	**0.99 ± 0.004**	**0.99 ± 0.003**	**0.99 ± 0.004**	**0.99 ± 0.004**
Fashion-MNIST	FedAvg [45]	0.99 ± 0.020	0.99 ± 0.020	0.99 ± 0.020	0.99 ± 0.020
	NbAFL [6]	0.79 ± 0.066	0.78 ± 0.066	0.79 ± 0.066	0.78 ± 0.078
	Fed-CDP [37]	0.76 ± 0.060	0.76 ± 0.054	0.76 ± 0.060	0.74 ± 0.073
	Fed-αCDP [38]	0.77 ± 0.060	0.77 ± 0.053	0.77 ± 0.060	0.75 ± 0.072
	ResSFL [41]	0.97 ± 0.082	0.97 ± 0.082	0.97 ± 0.084	0.97 ± 0.082
	Dropout [40]	0.94 ± 0.062	0.94 ± 0.063	0.94 ± 0.062	0.94 ± 0.064
	RVE-PFL	**0.99 ± 0.017**	**0.99 ± 0.014**	**0.99 ± 0.017**	**0.99 ±0.018**
Cifar-10	FedAvg [45]	0.97 ± 0.071	0.97 ± 0.069	0.97 ± 0.071	0.97 ± 0.072
	NbAFL [6]	0.40 ± 0.070	0.40 ± 0.065	0.40 ± 0.070	0.39 ± 0.077
	Fed-CDP [37]	0.22 ± 0.18	0.22 ± 0.17	0.22 ± 0.18	0.20 ± 0.18
	Fed-αCDP [38]	0.22 ± 0.18	0.22 ± 0.17	0.22 ± 0.17	0.20 ± 0.18
	ResSFL [41]	0.96 ± 0.032	0.96 ± 0.034	0.96 ± 0.034	0.96 ± 0.032
	Dropout [40]	0.92 ± 0.044	0.92 ± 0.043	0.92 ± 0.042	0.92 ± 0.043
	RVE-PFL	**0.99 ± 0.082**	**0.99 ± 0.072**	**0.99 ± 0.081**	**0.99 ± 0.086**

The complexity of a dataset can have a significant impact on the efficiency of approaches. The Fashion-MNIST dataset is more intricate than the MNIST dataset, as the images contain more intricate shapes and textures. In addition, the dataset contains greater variation within each class, making accurate classification more challenging. The CIFAR-10 dataset, on the other hand, is more complex than both MNIST and Fashion-MNIST due to its use of colour images and intricate shapes and textures. Even greater variation exists within each class, making the dataset extremely difficult to accurately classify.

The empirical results align with these observations, as evidenced by Tables I, II. The complexity of a dataset hurts the performance of NbAFL, Fed-CDP, and Fed-αCDP, as these algorithms struggle to achieve high accuracy on datasets such as CIFAR-10 and Fashion-MNIST. Specifically, NbAFL's performance does not surpass 41% on the CIFAR-10 dataset and 81% on the Fashion-MNIST dataset. However, the Fed-Avg and RVE-PFL continue to achieve reasonable performance on the CIFAR-10 dataset, with an average performance of 97% and 99%, respectively. Interestingly, when it comes to the MNIST and Fashion-MNIST datasets, RVE-PFL and Fed-Avg perform similarly and achieve the same level of reasonable accuracy with an average performance of 99%. On the other hand, the performance of NbAFL, Fed-CDP, and Fed-αCDP is

degraded, despite the relative simplicity of the MNIST dataset. This is due to the additive noise used in NbAFL, Fed-CDP, and Fed-αCDP. Interestingly, ResSFL and Dropout achieve intermediate levels of accuracy on test data, ranging from 92.5% to 97%. Overall, RVE-PFL has proven to be robust, achieving superior performance without being affected by the complexity of the dataset. This highlights the effectiveness of RVE-PFL in handling complex datasets and demonstrates its potential for use in real-world scenarios.

b) Privacy leakage: MIFace, DLG, and iDLG MI attacks are employed to test the resilience of RVE-PFL to mitigate the MI attack. The success of MI attacks is contingent upon determining if recovered data reveals sensitive information about a target label. Thus, we investigate the attack performance both quantitatively and qualitatively. Qualitative analysis of the attack performance occurs via visual inspection of the attack outputs, and quantitative analysis is based on analysis of four metrics; MSE, PSNR, SSIM, and FID. These are individually outlined above. Table III shows a comparison of the robustness of RVE-PFL against MI attacks using MIFace, in comparison to the state-of-the-art approaches: FedAvg, NbAFL, Fed-CDP, Fed-αCDP, ResSFL, and Dropout. Additionally, this table demonstrates the efficacy of the MI attacks and their ability to infer sensitive information about the participants in FL, based on the model structure and trained parameters.

It is evident from Figure 4 that the MIFace attack can successfully target deep learning models in the context of FL and expose sensitive features related to the participants' private data. Specifically, in the case of FedAvg, ResSFL, and Dropout, Figure 4 provides visual evidence of the attack's ability to infer meaningful patterns related to the participants' private data. However, in the case of NbAFL, Fed-CDP, Fed-αCDP, and RVE-PFL, the attack is hardly able to infer visually meaningful features close to the ground truth. Also, it became evident that the MI attack has the potential to reveal training data information embedded in the model parameters obtained through ReSFL and Dropout. Additionally, distinct features of the training data were discernible in the reconstructed images derived from the model parameters trained using ResSFL and Dropout. The robustness of RVE-PFL is further evident in Figure 5 which displays the reconstructed training data through DLG and iDLG model inversion attacks. Our observations indicate that RVE-PFL does not exhibit any discernible patterns related to the training data.

At this point, The significance of the quantitative metrics for evaluating model inversion attacks becomes evident for demonstrating that although some images in Figure 4 may not have obvious patterns, they still contain sensitive information that is quantitatively similar to the ground truth. Therefore, Table III presents the quantitative metrics for the model inversion attack.

Based on the quantitative measurements presented in Table III, we can see that RVE-PFL achieves higher MSE and FID rates but lower PSNR and SSIM rates across all three datasets. This exemplifies the inferior quality of the reconstructed images obtained from the MI attack when RVE-PFL was used in the development of the FL system. These results also imply that RVE-PFL outperforms NbAFL, Fed-CDP, and Fed-αCDP in terms of privacy preservation, as it produces greater MSE and FID and lower PSNR and SSIM, regardless of the dataset's complexity. As demonstrated in the section on performance and convergence analyses, RVE-PFL also maintains an outstanding level of model utility while limiting privacy leakage. In addition, RVE-PFL has an outstandingly high level of model utility, as we have shown in the part of the discussion devoted to performance and convergence analysis.

Table IV presents the improvement rates of privacy-preserving evaluation metrics for RVE-PFL versus the FedAvg, NbAFL, Fed-CDP, Fed-αCDP, ResSFL, and Dropout. It is observed that RVE-PFL outperforms FedAvg in terms of privacy preservation, as indicated by the average results. Specifically, we observe a 41.93% increase in MSE, a 15.93% decrease in PSNR, a 45.99% decrease in SSIM, and a 16.92% increase in FID. Notably, RVE-PFL demonstrates higher improvement rates over FedAvg compared to NbAFL, Fed-CDP, and Fed-αCDP, likely because FedAvg lacks additional mechanisms for privacy preservation.

The results also show that RVE-PFL outperformed the NbAFL, Fed-CDP, and Fed-αCDP approaches in terms of all four privacy metrics. Hence, RVE-PFL achieved an average improvement of 25.07% in MSE, a decrease of 8.05% in PSNR, a decrease of 37.95% in SSIM, and an increase of 13.18% in FID compared to the NbAFL approach. These results indicate that RVE-PFL is more effective in preserving privacy and more robust in mitigating MI attacks than the NbAFL approach. Specifically, on MNIST, RVE-PFL achieved an improvement rate of 3.125% for MSE and 2.18% for FID. On Fashion-MNIST, the improvement rates were 23.08% for MSE and 14.02% for FID, while on CIFAR-10, they were 50% for MSE and 24.34% for FID. In contrast, RVE-PFL achieved improvement rates for PSNR and SSIM for all three datasets, which implies that the reconstructed images obtained by the MIFace attack on RVE-PFL were of lower quality than those obtained on NbAFL. Notably, RVE-PFL outperforms ResSFL and Dropout in enhancing privacy preservation metrics. Specifically, when juxtaposed with ResSFL, RVE-PFL attains a substantial average improvement rate of 19.70% in MSE, 11.92% in PSNR, 52.33% in SSIM, and 21.23% in FID across MNIST, Fashion-MNIST, and Cifar-10 datasets. Similarly, in comparison with Dropout, RVE-PFL demonstrates a noteworthy average improvement rate of 25.18% in MSE, 11.01% in PSNR, 46.98% in SSIM, and 17.42% in FID.

The results also show that the level of improvement varies depending on the complexity of the dataset. The CIFAR-10 dataset shows the highest improvement rates, while MNIST has the lowest improvement rates. This implies that the robustness and effectiveness of RVE-PFL will mitigate the model inversion attacks and consequently preserve privacy despite the complexity of the dataset and the deep learning model.

Table IV reveals interesting findings when comparing RVE-PFL with Fed-CDP and Fed-αCDP approaches. RVE-PFL demonstrates significantly higher privacy improvement rates compared to Fed-CDP, Fed-αCDP, ResSFL, and Dropout, as well as outperforms NbAFL in terms of privacy preservation. These results suggest that Fed-CDP and Fed-αCDP, which rely on per-example DP, might not be sufficient to

TABLE III

PRIVACY-PRESERVING EVALUATION METRICS BETWEEN THE GROUND TRUTH IMAGES AND THE RECONSTRUCTED IMAGES BY MIFACE ATTACK

Dataset	Approach	Privacy Metrics			
		MSE ↑	PSNR ↓	SSIM ↓	FID ↑
MNIST	FedAvg [45]	0.21	6.85	0.22	6.31
	NbAFL [6]	0.32	5.05	0.07	6.41
	Fed-CDP [37]	0.16	7.95	0.13	4.16
	Fed-αCDP [38]	0.17	7.85	0.12	4.21
	ResSFL [41]	0.26	6.6	0.17	5.40
	Dropout [40]	0.27	6.5	0.15	5.46
	RVE-PFL	**0.33**	**4.78**	**0.02**	**6.55**
Fashion-MNIST	FedAvg [45]	0.12	9.34	0.14	3.9
	NbAFL [6]	0.13	9.57	0.17	4.07
	Fed-CDP [37]	0.12	9.51	0.13	3.87
	Fed-αCDP [38]	0.12	9.40	0.13	3.92
	ResSFL [41]	0.13	9.12	0.15	3.46
	Dropout [40]	0.12	9.25	0.14	3.86
	RVE-PFL	**0.16**	**8.78**	**0.12**	**4.64**
Cifar-10	FedAvg [45]	0.09	10.91	0.13	3.47
	NbAFL [6]	0.08	11.22	0.1	3.57
	Fed-CDP [37]	0.07	11.49	0.14	3.36
	Fed-αCDP [38]	0.07	11.35	0.13	3.40
	ResSFL [41]	0.11	10.20	0.16	4.10
	Dropout [40]	0.10	10.29	0.15	3.96
	RVE-PFL	**0.12**	**9.76**	**0.09**	**4.44**

Fig. 4. Visual results of the MIFace attack on the three datasets.

safeguard against privacy leakage through MI attacks while also struggling to maintain satisfactory model utility, as evidenced by the performance metrics presented in Tables I and II, and Figure 3. Hence, RVE-PFL emerges as a promising alternative, offering improved privacy protection without compromising on model performance, making it a more robust choice for privacy-preserving federated learning scenarios.

It is noteworthy that the positive values in Table IV represent improvements in the positive direction, while the negative values indicate improvements in the negative direction. For example, in the MSE column, all values are positive, indicating that RVE-PFL has higher MSE values than NbAFL, Fed-CDP, and Fed-αCDP, which is desirable. On the other hand, in the SSIM column, all values are negative, indicating that RVE-PFL has lower SSIM values than NbAFL, Fed-CDP, Fed-αCDP, ResSFL, and Dropout which is desirable for privacy preservation.

c) Computational cost: The RVE-PFL approach entails a computational cost of 41.04 million floating-point operations (FLOPs) and encompasses 20.52 million parameters, demanding an approximate memory allocation of 78.06 megabytes (MB). In contrast, the baseline FL incurs 4.1 million FLOPs with 0.824 million parameters, necessitating a memory space of 3.15 MB. Despite the incremental computational load of RVE-PFL, it manifests a commendable equilibrium between utility and privacy preservation. Notably, it remains well-suited for deployment on resource-constrained IoT devices, demonstrating a judicious memory footprint of 78.06 MB. This underscores its aptitude for scenarios characterized by limited computational resources.

d) The impact of adding more clients: We assessed the performance of RVE-PFL across various numbers of FL participants on the Fashion-MNIST dataset, as detailed in Table VI. Our findings indicate that as the number of clients increases, RVE-PFL maintains comparable privacy measures with a marginal decrease in accuracy.

e) Comparison with related encoding-based approaches: Table V presents a comparison of the performances of

TABLE IV
IMPROVEMENT RATES FOR PRIVACY-PRESERVING EVALUATION METRICS OF RVE-PFL

Approach	Dataset	Privacy Metrics			
		MSE ↑	PSNR ↓	SSIM ↓	FID ↑
FedAvg [45]	MNIST	57.14%	-30.24%	-90.91%	3.80%
	Fashion-MNIST	33.33%	-5.99%	-14.29%	18.97%
	CIFAR-10	33.33%	-10.56%	-30.77%	28.00%
	Average	**41.93%**	**-15.93%**	**-45.99%**	**16.92%**
NbAFL [6]	MNIST	3.125%	-5.35%	-71.43%	2.18%
	Fashion-MNIST	23.08%	-8.23%	-29.41%	14.02%
	CIFAR-10	50%	-11.58%	-10%	24.34%
	Average	**25.07%**	**-8.05%**	**-37.95%**	**13.18%**
Fed-CDP [37]	MNIST	106.25%	-39.37%	-84.62%	57.45%
	Fashion-MNIST	33.33 %	-7.57%	-7.69%	20.57%
	CIFAR-10	71.43%	-15.02%	-35.71%	32.14%
	Average	**70.67%**	**-20.65%**	**-42.00%**	**36.72%**
Fed-αCDP [38]	MNIST	94.12%	-39.11%	-83.33%	55.82%
	Fashion-MNIST	33.33%	-6.38%	-7.69%	18.37%
	CIFAR-10	71.43%	-13.99%	-30.77%	30.59%
	Average	**66.29%**	**-19.83%**	**-40.60%**	**34.26%**
ResSFL [41]	MNIST	26.92%	-27.58%	88.24%	21.30%
	Fashion-MNIST	23.08%	-3.87%	-25.0%	34.10%
	CIFAR-10	9.09%	-4.31%	-43.75%	8.29%
	Average	**19.70%**	**-11.92%**	**-52.33%**	**21.23%**
Dropout [40]	MNIST	22.22%	-26.46%	-86.67%	19.93%
	Fashion-MNIST	33.33%	-1.42%	-14.29%	20.21%
	CIFAR-10	20.0%	-5.16%	-40.0%	12.12%
	Average	**25.18%**	**-11.01%**	**-46.98%**	**17.42%**

TABLE V
COMPARISON OF PERFORMANCES OF RVE-PFL AND ENCODING-BASED APPROACHES

Approach / Dataset	**MNIST**	**Fashion-MNIST**	**Cifar-10**
Classification-Compliant Autoencoder [18]	97.63	N/A	80.36
Distributed Encoders [9]	99.45	81.82	85.02
InstaHide [33]	98.2±0.2	N/A	91.4±0.2
RVE-PFL	**0.99±0.001**	**0.99±0.008**	**0.99±0.053**

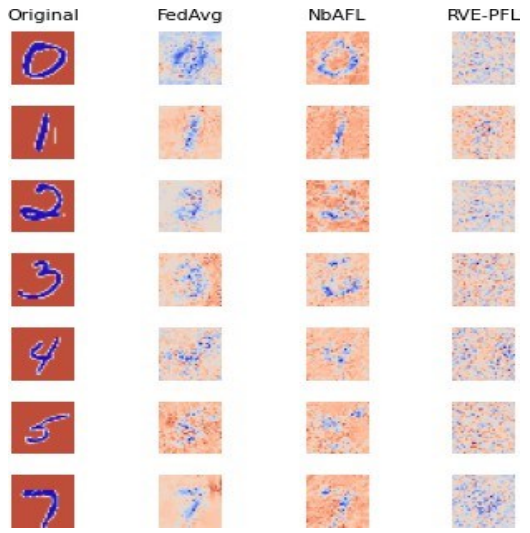


Fig. 5. Visual representations of training data reconstructions obtained through DLG and iDLG model inversion attacks.

TABLE VI
PERFORMANCE AND ROBUSTNESS OF RVE-PFL ACROSS MULTIPLE CLIENTS ON THE FASHION-MNIST DATASET

M	Acc	Privacy Metrics			
		MSE ↑	PSNR ↓	SSIM ↓	FID ↑
10	99.01%	0.165	8.78	0.125	4.64
40	98.98%	0.163	8.75	0.126	4.61
70	98.97%	0.164	8.74	0.124	4.63
100	98.98%	0.162	8.77	0.125	4.62
150	98.97%	0.163	8.73	0.123	4.64

RVE-PFL and related approaches on different datasets named MNIST, Fashion-MNIST, and Cifar-10. The approaches included in the comparison are the Classification-Compliant Autoencoder [18], Distributed Encoders [9], InstaHide [33], and RVE-PFL. In terms of accuracy, RVE-PFL consistently achieves high performance across all three datasets, with an accuracy of 0.99±0.001 for MNIST, 0.99±0.008 for Fashion-MNIST, and 0.99±0.053 for Cifar-10.

Comparatively, the Classification-Compliant Autoencoder achieves an accuracy of 97.63% on MNIST and 80.36% on Cifar-10, while no value is reported for Fashion-MNIST. Distributed Encoders perform well with an accuracy of 99.45%

on MNIST, 81.82% on Fashion-MNIST, and 85.02% on Cifar-10. InstaHide shows an accuracy of 98.2±0.2% on MNIST and 91.4±0.2% on Cifar-10, with no reported value for Fashion-MNIST. Overall, the results indicate that RVE-PFL achieves competitive accuracy on all three datasets when compared to the related approaches, demonstrating its effectiveness in PFL and protection against MI attacks.

VII. Conclusion

We have introduced an RVE-PFL approach for enhancing the privacy of FL systems that uses a personalised variational encoder to protect against MI attacks while preserving model utility. RVE-PFL consists of two components: personalised encoding and the FL classifier. The former transforms client-private data into a probabilistic latent space, while the latter is locally trained using the latent space and globally aggregated by the server. RVE-PFL has demonstrated satisfactory performance in terms of privacy protection and model utility. Specifically, our approach has proven effective at mitigating inversion attacks, a common privacy risk in FL applications and machine learning applications in general. However, as concerns about data privacy continue to grow, techniques that can effectively protect sensitive data while still maintaining the model's utility will become increasingly valuable. Thus, RVE-PFL is a step in this direction, and we believe it has the potential to inform future efforts to develop FL systems that are more secure and effective. In the future, the proposed approach should be further evaluated and expanded to include not only image datasets but also tabular and text data. This expansion would allow the application of the RVE-PFL approach in a wider range of use cases and provide additional insights into its effectiveness and generalizability. As part of future work, it would be valuable to evaluate the robustness of our approach against membership and property inference attacks.

References

[1] Polaris Market Research. (2021). *Federated Learning Market Size, Share, Trends, Growth, Demand, Forecast To 2028*. Accessed: Apr. 1, 2023. [Online]. Available: https://www.polarismarketresearch.com/industry-analysis/federated-learning-market

[2] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart., 2021.

[3] M. S. Jere, T. Farnan, and F. Koushanfar, "A taxonomy of attacks on federated learning," *IEEE Secur. Privacy*, vol. 19, no. 2, pp. 20–28, Mar. 2021.

[4] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Comput. Surv.*, vol. 55, no. 4, pp. 1–35, 2022.

[5] Y. Liu et al., "ML-doctor: Holistic risk assessment of inference attacks against machine learning models," in *Proc. 31st USENIX Secur. Symp.*, Aug. 2022, pp. 4525–4542.

[6] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[7] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf.*, 2020, pp. 493–506.

[8] A. Triastcyn and B. Faltings, "Federated learning with Bayesian differential privacy," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 2587–2596.

[9] Y. Zhang, H. Salehinejad, J. Barfett, E. Colak, and S. Valaee, "Privacy preserving deep learning with distributed encoders," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2019, pp. 1–5.

[10] M. Keshk, B. Turnbull, N. Moustafa, D. Vatsalan, and K. R. Choo, "A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5110–5118, Aug. 2020.

[11] A. Caciularu and D. Burshtein, "Unsupervised linear and nonlinear channel equalization and decoding using variational autoencoders," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 1003–1018, Sep. 2020.

[12] Z. Zhang, Z. Tianqing, W. Ren, P. Xiong, and K.-K.-R. Choo, "Preserving data privacy in federated learning through large gradient pruning," *Comput. Secur.*, vol. 125, Feb. 2023, Art. no. 103039.

[13] T. Stevens, C. Skalka, C. Vincent, J. Ring, S. Clark, and J. Near, "Efficient differentially private secure aggregation for federated learning via hardness of learning with errors," in *Proc. 31st USENIX Secur. Symp.*. Boston, MA, USA, 2022, pp. 1379–1395. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/stevens

[14] L. Lyu et al., "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 10, 2022, doi: 10.1109/TNNLS.2022.3216981.

[15] Q. Li et al., "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, Apr. 2023.

[16] X. Jiang, X. Zhou, and J. Grossklags, "Privacy-preserving high-dimensional data collection with federated generative autoencoder," *Proc. Privacy Enhancing Technol.*, vol. 2022, no. 1, pp. 481–500, Jan. 2022.

[17] R. M. Alguliyev, R. M. Aliguliyev, and F. J. Abdullayeva, "Privacy-preserving deep learning algorithm for big personal data analysis," *J. Ind. Inf. Integr.*, vol. 15, pp. 1–14, Sep. 2019.

[18] Y. Ma, Y. Yao, X. Liu, and N. Yu, "Privacy-preserving collaborative learning with scalable image transformation and autoencoder," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 1031–1036.

[19] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1322–1333.

[20] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox,and R. Garnett, Eds. Curran Associates, 2019, pp. 1–11. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf

[21] B. Zhao, K. Reddy Mopuri, and H. Bilen, "IDLG: Improved deep leakage from gradients," 2020, *arXiv:2001.02610*.

[22] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 9587–9603, 2022, doi: 10.1109/TNNLS.2022.3160699.

[23] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: An efficient federated learning framework for heterogeneous mobile clients," in *Proc. 27th Annu. Int. Conf. Mobile Comput. Netw.*, 2021, pp. 420–437.

[24] K. Pillutla, K. Malik, A.-R. Mohamed, M. Rabbat, M. Sanjabi, and L. Xiao, "Federated learning with partial model personalization," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 17716–17758.

[25] P. Pu Liang et al., "Think locally, act globally: Federated learning with local and global representations," 2020, *arXiv:2001.01523*.

[26] D. Bui et al. (2020). *Federated User Representation Learning*. [Online]. Available: https://openreview.net/forum?id=Syl-_aVtvH

[27] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–36, Mar. 2021.

[28] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 7232–7241.

[29] R. Mayer and A. Ekelhart, "Macro-level inference in collaborative learning," in *Proc. 12th ACM Conf. Data Appl. Secur. Privacy*, Apr. 2022, pp. 373–375.

[30] J. Xu, C. Hong, J. Huang, L. Y. Chen, and J. Decouchant, "AGIC: Approximate gradient inversion attack on federated learning," in *Proc. 41st Int. Symp. Rel. Distrib. Syst. (SRDS)*, 2022, pp. 12–22.

[31] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 148–162.

[32] S. Truex et al., "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, Nov. 2019, pp. 1–11.

[33] Y. Huang, Z. Song, K. Li, and S. Arora, "InstaHide: Instance-hiding schemes for private distributed learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4507–4518.

[34] A. Madi, O. Stan, A. Mayoue, A. Grivet-Sébert, C. Gouy-Pailler, and R. Sirdey, "A secure federated learning framework using homomorphic encryption and verifiable computing," in *Proc. Reconciling Data Anal., Autom., Privacy, Secur., Big Data Challenge*, 2021, pp. 1–8.

[35] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "HybridAlpha: An efficient approach for privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, Nov. 2019, pp. 13–23.

[36] Z. Zhang, L. Zhang, Q. Li, K. Wang, N. He, and T. Gao, "Privacy-enhanced momentum federated learning via differential privacy and chaotic system in industrial cyber–physical systems," *ISA Trans.*, vol. 128, pp. 17–31, Sep. 2022.

[37] W. Wei, L. Liu, Y. Wut, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2021, pp. 797–807.

[38] W. Wei, L. Liu, J. Zhou, K.-H. Chow, and Y. Wu, "Securing distributed SGD against gradient leakage threats," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 7, pp. 2040–2054, Jul. 2023.

[39] M. Naseri, J. Hayes, and E. De Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," 2020, *arXiv:2009.03561*.

[40] Z. He, T. Zhang, and R. B. Lee, "Attacking and protecting data privacy in edge–cloud collaborative inference systems," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9706–9716, Jun. 2021.

[41] J. Li, A. S. Rakin, X. Chen, Z. He, D. Fan, and C. Chakrabarti, "ResSFL: A resistance transfer framework for defending model inversion attack in split federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10184–10192.

[42] S. J. Oh, B. Schiele, and M. Fritz, *Towards Reverse-Engineering Black-Box Neural Networks*. Berlin, Germany: Springer, 2022, pp. 121–144, doi: 10.1007/978-3-030-28954-6_7.

[43] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture IE*, vol. 2, no. 1, pp. 1–18, 2015.

[44] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[45] B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[46] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.