TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Meijer, C., Huang, J., Sharma, S., Lazovik, E., & Chen, L. Y. (2025). TS-Inverse: A Gradient Inversion Attack Tailored for Federated Time Series Forecasting Models. In *Proceedings - 2025 IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2025* (pp. 110-124). IEEE. https://doi.org/10.1109/SaTML64287.2025.00014

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

TS-Inverse: A Gradient Inversion Attack Tailored for Federated Time Series Forecasting Models

Caspar Meijer*† caspar.meijer@tno.nl
Jiyue Huang* j.huang-4@tudelft.nl
Shreshtha Sharma† shreshtha.sharma@tno.nl
Elena Lazovik† elena.lazovik@tno.nl
Lydia Y. Chen*‡ lydiaychen@ieee.org

*Delft University of Technology, Delft, The Netherlands
†Dutch Organization for Applied Scientific Research (TNO), The Hague, The Netherlands
‡University of Neuchâtel, Neuchâtel, Switzerland

2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML) | 979-8-3315-1711-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/SATML64287.2025.00014

***Abstract*—Federated learning (FL) for time series forecasting (TSF) enables clients with privacy-sensitive time series (TS) data to collaboratively learn accurate forecasting models, for example, in energy load prediction. Unfortunately, privacy risks in FL persist, as servers can potentially reconstruct clients' training data through gradient inversion attacks (GIA). Although GIA is demonstrated for image classification tasks, little is known about time series regression tasks. In this paper, we first conduct an extensive empirical study on inverting TS data across 4 TSF models and 4 datasets, identifying the unique challenges of reconstructing both observations and targets of TS data. We then propose TS-Inverse, a novel GIA that improves the inversion of TS data by (i) learning a gradient inversion model that outputs quantile predictions, (ii) a unique loss function that incorporates periodicity and trend regularization, and (iii) regularization according to the quantile predictions. Our evaluations demonstrate a remarkable performance of TS-Inverse, achieving at least a 2x-10x improvement in terms of the sMAPE metric over existing GIA methods on TS data. Code repository: https://github.com/Capsar/ts-inverse**

***Index Terms*—Federated learning, time series, forecasting, gradient inversion attack**

I. Introduction

Federated learning (FL), a distributed machine learning framework in which a server and multiple clients collaboratively train a global model, is established as an effective method for training deep neural networks without centrally storing client training data. In this framework, clients independently train the model on their local data, and then transmit these updates to a central server. The server aggregates these updates and sends the new global model back to the clients who complete the global training round [1]. This learning paradigm has wide applications in industries that face privacy issues when collecting raw data, such as the energy distribution industry [2].

The energy industry actively researches data-driven technologies for load forecasting, a typical challenge in time series (TS) data, where historical data is used to predict future load patterns [2], [3]. Specifically the forecasting of individual loads on low-voltage (smart-)grids is a key subject of interest within the community as it allows the Distribution System Operators to optimize the grid on a local level [3], [4]. Local-level forecasting, while beneficial, faces a legal challenge: The EU general data protection regulation (GDPR) protects the privacy of household and corporate energy data, preventing its central aggregation for ML model training [5], [6]. This is where FL comes into play, allowing the data to remain at its source. Figure 1 illustrates an example scenario where households are equipped with smart meters that record private energy data, participating as clients in the FL system. The objective of the FL system is to build a global forecasting model that predicts individual household energy consumption, using the historical energy consumption data from all clients.

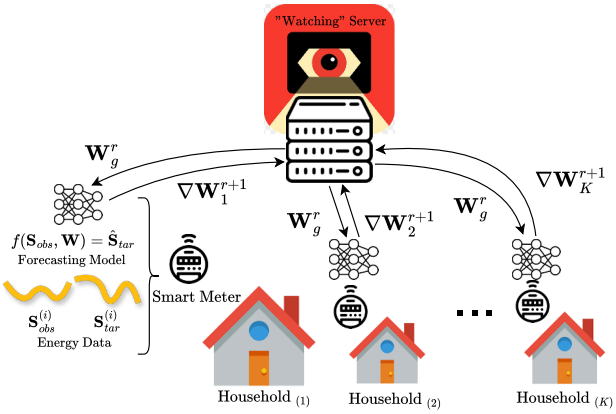


Fig. 1: Federated time series forecasting: a use case in energy demand forecasting with an *honest-but-curious* "watching" server. Each smart meter has a forecasting model and a sequence of energy data. The server distributes the global model parameters and retrieves the locally trained models.

Nonetheless, FL is not without privacy risks, such as membership inference, and property inference attacks [7], [8]. Given strict privacy regulations like GDPR, it is crucial to ensure that private energy data is not leaked through FL systems [6], just as with sensitive medical images [9]. This paper specifically addresses gradient inversion attacks (GIAs), where attackers aim to reconstruct private training data from client updates [10] by matching clients' gradients with so-

This research was partially supported by the SNSF project, Priv-GSyn (grant number 200021E_229204), and the Advanced Computing Engineering (ACE) Department at TNO.

979-8-3315-1711-3/25/$31.00 ©2025 IEEE
DOI 10.1109/SaTML64287.2025.00014

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

called dummy gradients. Previous research has demonstrated the feasibility of such attacks in image classification tasks, using model parameter updates from FL training to reconstruct training data [11]–[20]. Improving privacy attacks is essential, as understanding these enables a more rigorous evaluation of defense mechanisms, ultimately enhancing the overall security of privacy preserving systems.

In this paper, we address this gap by focusing on GIAs for federated time series forecasting (TSF) models. Inverting TS poses unique challenges compared to inverting image data with classification tasks. The training data of TS is composed of a sequence of observations and target values, whereas the existing GIA deals with input pairs of images and labels. Reconstructing TS thus requires inverting gradients into the sequences of observations and targets. The network architecture for TSF includes components to capture time dependency, e.g., Gated Recurrent Unit (GRU) [21] and 1-D temporal convolution (TCN) [22]. To demystify the inversion risk for TSF, we start with an empirical analysis by applying existing GIAs on combinations of 4 TSF models and 4 datasets. Based on the insights observed, we propose a novel and effective inversion framework, `TS-Inverse`, which utilizes additional knowledge and TS related characteristics for regularization. We also derive the analytical inversion of target values for the special case with a batch size of one. Extensive evaluation results against existing GIAs and ablation studies show that `TS-Inverse` is able to reconstruct both the observation and target values of time series, achieving low reconstructing errors.

Our contributions can be summarized as follows:

- We conduct the first empirical analysis of GIAs on federated TSF models, highlighting the challenges of inverting both observation and target sequences with respect to model architecture, gradient distance, and regularization. (Section III)
- We design `TS-Inverse`, which includes two innovative components: the gradient inversion model and TS-regularized inversion optimization. The gradient inversion model provides relevant quantile bounds by leveraging auxiliary data. The gradient distance is the L1-Norm with TS regularization incorporating periodicity, trend, and the learned quantile bounds of the gradient inversion model. Additionally, a one-shot analytical technique is used for reconstructing targets. (Section IV)
- We extensively evaluate `TS-Inverse` on four datasets and five forecasting architectures, demonstrating a 2x-10x reduction in reconstruction error on the non-RNN-based architectures. (Section V)

II. Background

A. Time Series Forecasting with Federated Learning

Most of the existing FL studies focus on image and text classification tasks [8], but its application to TS data is relatively under explored and primarily concentrated in the energy forecasting domain [23]–[25]. In the context of FL for TSF, Recurrent Neural Networks (RNNs) [26] and TCNs [22] are commonly used, opposed to traditional statistical models, due to their superior ability to capture complex temporal patterns in data. RNNs [26], specifically Long Short-Term Memory (LSTM) networks [27] and GRUs [21], are commonly used in federated settings for energy forecasting tasks [23], [28]. GRUs are an alternative to LSTMs that use fewer gating mechanisms to control flow of information, offering more computational efficiency while maintaining performance.

GRUs are designed to capture dependencies in sequential data by maintaining a hidden state. GRUs consist of two main gates: the update gate and the reset gate. The update gate controls the degree to which the hidden state is updated, while the reset gate determines how much of the previous hidden state should be forgotten. This gating mechanism allows GRUs to capture temporal dependencies efficiently, although it results in a many-to-one mapping from input sequences to hidden states.

Recently, non-recurrent architectures such as TCNs [22] have shown effectiveness in modeling sequence data for TSF tasks by using causal convolutions and dilations to capture long-range temporal dependencies. TCNs offer advantages in parallelization, training stability, and can maintain the same sequence length through zero-padding while using dilations to exponentially increase the receptive field, allowing them to model temporal dependencies more effectively than traditional convolutional neural networks (CNNs) and without the need for recurrence [29]. Alongside these, simpler Fully Connected Networks (FCNs) are also considered due to their ability to be trained on lower-quality hardware [25]. Despite advancements in transformer-based models for time series analysis, their application in federated TSF remains limited. Therefore, in this paper, we focus on the GRU and TCN architectures for federated TSF.

B. Non-IID Time Series Data in Federated Learning

Inherited from the nature of FL, training federated TSF models also involves non-identical and independently distributed client data [23], [24], [30]. Taking energy load forecasting as an example, each client, such as households and company buildings, possesses a distinct load profile. Each client, which is also the private individual, only contains its own unique load data. It results in the energy load of different clients varying in both peak/off hours and amount. This is different from cases where multiple private individuals can reside in a single client data silo. Although the non-iidness of load profiles and their accompanying convergence issues are beyond the scope of this paper, it is crucial to note that each client's load profile is private and must be protected against privacy leakage [31], [32]. Notably, the mentioned studies merely discuss differential privacy budgets and accuracy differences, without evaluating any actual attacks [31], [32].

C. Privacy and Inversion Attacks on Federated Learning

While adversarial attacks in FL aim to modify and disrupt learning tasks [8], privacy attacks focus on inferring

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

private information from the system [8]. Examples of such privacy attacks include membership inference attacks [33] and property inference attacks [34]. Among these privacy attacks, data reconstruction attacks are particularly concerning as they attempt to recover training samples from clients [11]–[20]. These attacks, known as GIAs, exploit gradients collected from clients, making them a significant threat to privacy in FL systems. To our knowledge there are no other studies that investigate the GIAs for TS data and models. Therefore, this paper will demonstrate 4 related studies on TS data and models in order to investigate their performance.

The four baselines examined in this study are DLG, InvG, LTI, and DIA. "Deep Leakage from Gradients" (DLG) [11] is the pioneering approach that demonstrated how training data could be reconstructed by minimizing the Euclidean distance between gradients, highlighting the feasibility of such attacks. Building upon DLG, "Inverting Gradient" (InvG) [14] introduced the use of cosine similarity and, image processing related, total variation regularization, which improved the reconstruction quality. "Learning to Invert" (LTI) [20] represents a more recent advancement, leveraging a model trained on an auxiliary dataset to directly map the client samples from gradients, making it robust even against defenses like differential privacy. Lastly, the "Dropout Inversion Attack" (DIA) [19] extended the attack capabilities to scenarios where dropout layers are used, effectively circumventing this commonly employed defense mechanism. Together, these baselines provide a comprehensive perspective on the effectiveness of GIAs in various federated learning contexts.

III. Empirical Inversion Analysis on Time Series Forecasting

In this section, we introduce and analyze the risk of inverting clients' private TS data in the context of federated forecasting by applying existing classification-based attacks. We first define TSF and outline the assumptions made regarding the FL setups and threat model. Afterward, we formally define the gradient inversion attacks under our forecasting scenarios. Finally, we analyze the feasibility and risks of existing image-based inversion attacks on different time series models.

A. Problem Definition

1) Time Series Forecasting: TS data are generally represented through discrete time steps, determined by a chosen sampling frequency, though time is inherently continuous. TSF is a regression task that predicts future values given a historical segment. In this work, we focus on multi-step TSF where the future segment consists of multiple timesteps. Formally, the historical segment, comprising H steps, is defined as a series of subsequent observation timesteps $obs = \{t \in \mathbb{Z} \mid -H < t \leq 0\}$, whereas the future segment, spanning F steps, is denoted as the targets $tar = \{t \in \mathbb{Z} \mid 0 < t \leq F\}$ as illustrated in Figure 2. We represent historical data of the sequence $\mathbf{S}$ as $\mathbf{S}_{obs} \in \mathbb{R}^{H \times d}$, and the forecasting target data as $\mathbf{S}_{tar} \in \mathbb{R}^{F \times d}$, where d indicates the number of distinct features. A time series is univariate when $d = 1$, and

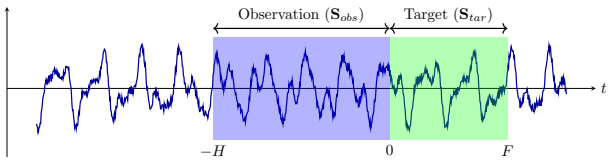


Fig. 2: Example of univariate time series forecasting.

multivariate when $d > 1$. Furthermore, $\mathbf{S}_t \in \mathbb{R}^d$ is used to denote a value vector at timestep t. We focus on the univariate case. For training the TSF task, we consider recent advanced deep learning models. The training objective of those models is to minimize the mean square errors (MSE) or mean absolute errors (MAE).

2) FL for Time Series Forecasting: A federated forecasting framework involves K clients, each having its own data silo with sequence data $\mathbf{S}_k$. Each client's sequence $\mathbf{S}_k$ is split up in multiple subsequent pairs of observation data $\mathbf{S}_{obs}^{(i)}$ and target data $\mathbf{S}_{tar}^{(i)}$, where i denotes the i'th sample from the sequence. A client's data silo contains a dataset with sequence samples organized as $\{\mathbf{S}_{obs}^{(i)}, \mathbf{S}_{tar}^{(i)}\}_{i=1}^{n_k}$, where n_k denotes the number of local sequence pairs. The clients jointly train a global forecasting model via a strategy determined by the server, e.g., learning rate and batch sizes.

The server initializes a global model with parameters $\mathbf{W}_g^r$, where g stands for the global model and r denotes the global training round. The server distributes $\mathbf{W}_g^r$ to the clients, who then follow federated stochastic gradient decent (FedSGD) [1] to further train the model using their own data. Specifically, the clients perform a single gradient update on a single batch, resulting in gradients $\nabla \mathbf{W}_k^r$. These are sent back to the server, which in turn, aggregates all model updates as $\mathbf{W}_g^{r+1} = \mathbf{W}_g^r - \alpha \left(\frac{1}{K} \sum_{k=1}^{K} \nabla \mathbf{W}_k^r \right)$, where α is the global learning rate.

B. Threat model

In line with the related studies, the threat model for our gradient inversion on TSF models assumes an *honest-but-curious* server. This means that while the server follows protocol rules, it is interested in extracting training data client information from the gradients data it receives. The attack considered is a *white-box* attack, where the FL strategy, model architecture, optimizer, loss function, observations, and target size ($\mathbb{R}^{d \times H}, \mathbb{R}^{d \times F}$), learning rate ($\alpha$), batch size ($\mathcal{B}$), etc. are all known to the adversary. The adversary has access to the model weights ($\mathbf{W}_g^r$) and aggregated gradients ($\nabla \mathbf{W}_k^r$) at communication time and aims to reconstruct each sample from the batched data, including both observations and targets ($\mathbf{S}_{obs}, \mathbf{S}_{tar}$). We also assume that the adversary has access to an auxiliary dataset ($\mathcal{D}_{aux}$) with a distribution overlapping with the client's data. Furthermore, the adversary has the computational capabilities to train models outside the FL framework.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

C. Attacking Time Series Forecasting

Inverting training data for TSF presents several new challenges compared to inverting training data for classification tasks. The training objectives and inputs differ: for image classification, the objective is cross-entropy loss with pairs of images and their class labels, whereas for time series forecasting, it is mean squared error (MSE) loss with pairs of observation and target series. In image classification, due to the cross-entropy loss, the labels can be analytically inferred from the gradients, as demonstrated by Zhao *et al.* [13] and Geng *et al.* [18]. Consequently, the inversion attack on a classification model primarily involves reconstructing the training images. However, inverting the training data for time series forecasting requires reconstructing both the input observations and the target values ($\mathbf{S}_{obs}$ and $\mathbf{S}_{tar}$) as both are of the same data modality. Furthermore, the analytical method of inferring the labels for the classification task is not directly applicable for a TSF task.

To recover the observation and target series data, the adversary first initializes a dummy input ($\widetilde{\mathbf{S}}_{obs}$) and target ($\widetilde{\mathbf{S}}_{tar}$) pair with the same dimensions as $\mathbf{S}_{obs}$ and $\mathbf{S}_{tar}$, respectively, and calculates the corresponding gradients:

$$\nabla\widetilde{\mathbf{W}} = \frac{\partial \mathcal{L}(f(\widetilde{\mathbf{S}}_{obs}, \mathbf{W}), \widetilde{\mathbf{S}}_{tar})}{\partial \mathbf{W}}, \tag{1}$$

where f is the global model and $\mathbf{W}$ are its parameters before applying the gradient updates. f is a twice differentiable neural network and $\mathcal{L}$ is the regression loss function used in optimizing f. These dummy observations and targets are optimized by minimizing a distance D between the dummy gradients $\nabla\widetilde{\mathbf{W}}$ from (1) and clients' gradients $\nabla\mathbf{W}$:

$$(\widetilde{\mathbf{S}}^{\star}_{obs}, \widetilde{\mathbf{S}}^{\star}_{tar}) = \underset{\widetilde{\mathbf{S}}_{obs}, \widetilde{\mathbf{S}}_{tar}}{\arg\min}\, D(\nabla\widetilde{\mathbf{W}}, \nabla\mathbf{W}). \tag{2}$$

In this work, we define the optimization problem for recovering both the observations ($\mathbf{S}_{obs}$) and the targets ($\mathbf{S}_{tar}$). Specifically, we use the gradient to recover data for both observations and targets. Additionally, the TS data is often pre-processed to be within the domain [0, 1], which helps to bound the reconstructed data. DLG [11] and others [13], [15], [16], [18] uses L2-Norm to match the gradients as:

$$D(\nabla\widetilde{\mathbf{W}}, \nabla\mathbf{W}) = \left\| \nabla\widetilde{\mathbf{W}} - \nabla\mathbf{W} \right\|_2. \tag{3}$$

Inverting Gradients [14] and others [17], [19] use the Cosine Similarity loss together with the image prior regularization, total variation, $\mathcal{R}_{\text{TV}}$, with hyperparameter λ_{TV}. The Cosine Similarity loss is defined as

$$\mathcal{D}_{\text{cosine}}(\nabla\widetilde{\mathbf{W}}, \nabla\mathbf{W}) = 1 - \frac{\nabla\widetilde{\mathbf{W}} \cdot \nabla\mathbf{W}}{\|\nabla\widetilde{\mathbf{W}}\|_2 \|\nabla\mathbf{W}\|_2} + \lambda_{\text{TV}}\mathcal{R}_{\text{TV}}. \tag{4}$$

D. Empirical Findings

We present our empirical findings by applying four existing attacks—DLG [11], InvG [14], DIA [19], and LTI [20]— on three models: CNN, FCN, and TCN for inverting the TS data from FL gradients. These attacks differ in their methodologies and gradient distance functions. Specifically, we discuss the impact of model architecture, loss function, and total variation for inversion performance. The experimental setups are detailed in Section V-A. We use the Symmetrical Mean Absolute Percentage Error (sMAPE↓) as the evaluation metric to compare all baselines, presented in Table II. Specifically, to demonstrate the impact of the total variation regularization ($\mathcal{R}_{\text{TV}}$) for InvG, we report the sMAPE results in Table I. Additionally, we also visualize the recovered observation and target data for easy perceptual comparison, by a single TS example in Figure 3. In-depth comparison is summarized in Table II and analyzed in Section V.

1) Model Architectures: The TCN architecture [22] includes dropout layers, which make the optimization attack more difficult [35]. DIA [19] counters this dropout defense by also optimizing the dropout masks that indicate which neurons are dropped during the training round. This is required in order to accurately reconstruct training data from architectures with dropout layers. Moreover, GRUs are designed to selectively remember and forget information, a process controlled through nonlinear functions governed by reset and update gates [21]. These gates determine the flow and modification of information over time, inherently leading to a loss of information and many-to-one mappings from input sequences to hidden states. From a gradient perspective, while gradients provide insights into how changes in inputs could affect the outputs, they only do so on average over the whole input space and do not offer direct mappings to individual inputs. This many-to-one relationship also applies to RNN architectures.

2) Gradient Distance Function of Attacks: The effects of the gradient distance function are most clear for the CNN and

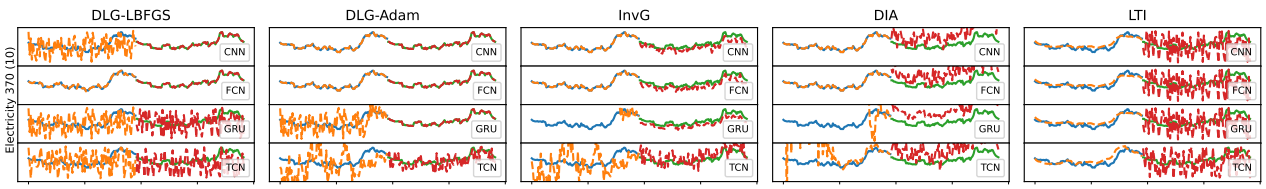


Fig. 3: Baseline reconstruction results with FCN, CNN, GRU-2-FCN, and TCN model architectures on the Electricity 370 dataset with seeds (10). In each graph, the blue and green lines are the ground truth observations and targets respectively. The orange and red lines are the respective reconstructed observations and targets.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

FCN architectures, according to Table II and Figure 3. The DLG attack, which uses L2-Norm to align gradients, effectively reconstructs the observations and targets. This is because L2-Norm measures the squared differences between gradients, inherently capturing both the magnitude and the direction of the gradients. In contrast, the InvG and DIA attacks rely on Cosine Similarity, which normalizes the gradients and focuses only on the direction, disregarding their magnitude. This focus on direction alone results in noisier targets, indicating that the magnitude of gradients is essential for accurate target reconstruction.

3) Total Variation: The application of total variation [14], is observed to have a counterproductive effect on the reconstruction of TS data, as shown in Table I. The regularization results in a higher sMAPE value when applied to TS. We applied the total variation separately to the observations and targets using hyperparameters $\lambda_{\text{TV}}^{obs}$ and $\lambda_{\text{TV}}^{tar}$, respectively. When increasing $\lambda_{\text{TV}}^{obs}$ while keeping $\lambda_{\text{TV}}^{tar} = 0$, the sMAPE increased for the observations, while for the targets it remained the same. In contrast, when increasing $\lambda_{\text{TV}}^{tar}$ while keeping $\lambda_{\text{TV}}^{obs} = 0$, the sMAPE for observations increased, while it decreased for targets. The total variation reduces noise in images, which helps improve the image reconstruction quality. However, this does not translate well to TS data. This is because TS data can be inherently noisier than images due to large differences between subsequent timesteps.

Highlights

- TCN and GRU architectures are more challenging to invert compared to CNN and FCN due to their inherent design features like dropout layers and gate mechanisms.
- To correctly recreate targets, the gradient distance has to take into account the magnitudes of the gradient values.
- Image regularization priors are not effective for time series data, which tends to be intrinsically noisier and exhibit larger differences between subsequent timesteps.

IV. TS-Inverse Framework

Following the findings from the empirical study, we propose a first-of-kind gradient inversion attack for federated TSF, named TS-Inverse, which inverts a batch of observation and target data of time series regression tasks from clients gradients. We illustrate the framework of TS-Inverse in Figure 4, consisting of three key steps. The first component of TS-Inverse is to train a gradient inversion model, i.e., f_{inv}, using auxiliary data to map model gradients to sequences of values that correspond to specific quantile ranges of the observations and targets, as depicted in Figure 7. The quantile ranges divide the data into intervals with equal probabilities, giving a representation of the observation and target distribution that belongs to the gradients. These sequences serve as upper and lower bounds for the learned quantile regularization. The second key feature is the inversion optimization, starting from the dummy time series. Different from the convention of minimizing L2-Norm and cosine distance in image inversion attacks, we advocate using L1-Norm loss and combine it with the proposed time series specific regularization terms, namely periodicity, trend, and quantile bounds. Furthermore, we provide an analytical one-shot inversion technique for the target sequence, under the scenario where the batch size is 1.

TABLE I: Comparison of the effects of total variation regularization on the InvG attack, measured using sMAPE (↓) with a CNN architecture. Evaluations are performed on the Electricity 370 and Proprietary Dataset for different combinations of $\lambda_{\text{TV}}^{obs}$ and $\lambda_{\text{TV}}^{tar}$ values. The experiments are run with a batch size $\mathcal{B} = 1$ and seeds 10, 43, and 28, with the best performances highlighted in bold. Standard deviations are provided as subscripts.

	Dataset	Electricity 370		Proprietary Dataset	
$\lambda_{\text{TV}}^{obs}$	$\lambda_{\text{TV}}^{tar}$	Observation	Target	Observation	Target
0	0	$\mathbf{0.002}_{0.00}$	$0.094_{0.06}$	$\mathbf{0.016}_{0.00}$	$1.143_{0.23}$
	0.001	$0.100_{0.01}$	$0.042_{0.04}$	$0.733_{0.16}$	$1.040_{0.25}$
	0.01	$0.797_{0.04}$	$\mathbf{0.009}_{0.00}$	$1.528_{0.07}$	$0.036_{0.02}$
0.001	0	$0.029_{0.01}$	$0.094_{0.06}$	$0.170_{0.02}$	$1.144_{0.23}$
	0.001	$0.067_{0.01}$	$0.042_{0.04}$	$0.467_{0.14}$	$1.042_{0.25}$
	0.01	$0.501_{0.06}$	$\mathbf{0.009}_{0.00}$	$1.394_{0.06}$	$\mathbf{0.028}_{0.01}$
0.01	0	$0.029_{0.00}$	$0.094_{0.06}$	$0.165_{0.02}$	$1.145_{0.23}$
	0.001	$0.067_{0.00}$	$0.042_{0.04}$	$0.472_{0.13}$	$1.042_{0.25}$
	0.01	$0.500_{0.06}$	$\mathbf{0.009}_{0.00}$	$1.398_{0.06}$	$\mathbf{0.028}_{0.01}$

A. Gradient Inversion Model

We assume the existence of an auxiliary dataset $\mathcal{D}_{aux}$ by the *honest-but-curious* server. This is ideally a subset of the time series of the data silos that are being attacked, but it can also be an out-of-distribution dataset [20]. The dataset is first pre-processed such that the samples have the same sequence length as the client's data samples through interpolation.

We define a gradient inversion model $f_{inv} : \mathbb{R}^m \rightarrow (\mathbb{R}^{H\times d\times Q}, \mathbb{R}^{F\times d\times Q})$, where m is the number of parameters in the forecasting model f, and Q is the number of quantile bounds. This model is designed to map gradients to quantile bounds, which are sequences of values corresponding to specific quantile levels of the TS observations and targets. Quantile levels (τ) are probability values between 0 and 1. For example, between the bounds corresponding to quantile levels 0.1 and 0.9 fall 80% of the sequences.

The model f_{inv} is trained on the gradients of the forecasting model f derived from batches of samples from $\mathcal{D}_{aux}$. A batch $(\mathbf{S}_{obs}^{\mathcal{B}}, \mathbf{S}_{tar}^{\mathcal{B}})$ consists of observation and target pairs from $\mathcal{D}_{aux}$, where $\mathcal{B}$ denotes the number of such pairs.

The model f_{inv} is optimized using the pinball loss with respect to the batched samples. It produces the outputs $\{\hat{\mathbf{S}}_{obs}^{\tau_1}, \hat{\mathbf{S}}_{obs}^{\tau_2}, \ldots, \hat{\mathbf{S}}_{obs}^{\tau_Q}\}$ and $\{\hat{\mathbf{S}}_{tar}^{\tau_1}, \hat{\mathbf{S}}_{tar}^{\tau_2}, \ldots, \hat{\mathbf{S}}_{tar}^{\tau_Q}\}$, where each τ_q represents a specific quantile level. These quantile predictions are repeated along the batch dimension to match $(\mathbf{S}_{obs}^{\mathcal{B}}, \mathbf{S}_{tar}^{\mathcal{B}})$. As a result the quantile predictions demonstrate the variability of the batch. The pinball loss, which is used for

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

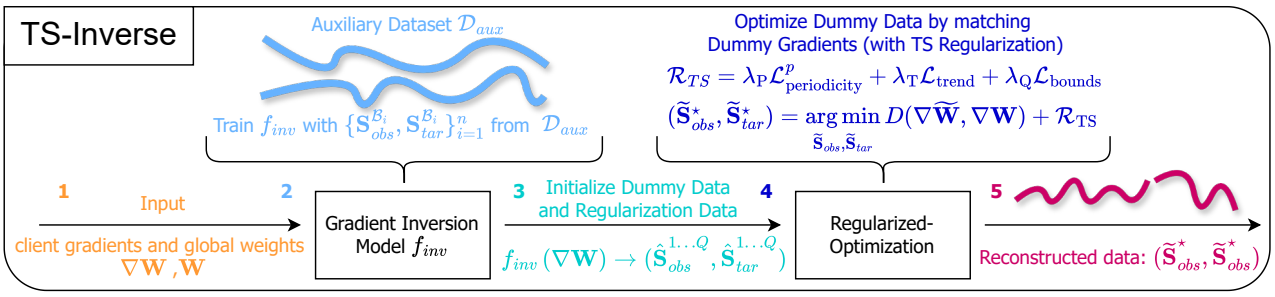


Fig. 4: Key steps of TS-Inverse framework

optimization, is defined as follows:

$$\mathcal{L}^{\tau}_{\text{pinball}}(\mathbf{S}, \hat{\mathbf{S}}) = \frac{1}{T}\sum_{t=1}^{T} \max\Big((\tau-1)(\mathbf{S}_t - \hat{\mathbf{S}}_t), \tau(\mathbf{S}_t - \hat{\mathbf{S}}_t)\Big), \tag{5}$$

where t is a timestep and T is the total number of timesteps in sequence $\mathbf{S}$, and $\hat{\mathbf{S}}$ represents the predicted quantile bound. The total loss is to apply (5) for each quantile level over the observations and targets as:

$$\mathcal{L}_{inv} = \sum_{q=1}^{Q} \frac{\mathcal{L}^{\tau_q}_{\text{pinball}}(\mathbf{S}^{\mathcal{B}}_{obs}, \hat{\mathbf{S}}^{\tau_q}_{obs}) + \mathcal{L}^{\tau_q}_{\text{pinball}}(\mathbf{S}^{\mathcal{B}}_{tar}, \hat{\mathbf{S}}^{\tau_q}_{tar})}{2}. \tag{6}$$

By using the pinball loss function, the model f_{inv} is effectively trained to provide quantile bounds for the entire batch of TS data.

B. Gradient Matching Distance Function

The cosine similarity loss only matches the hyperplane direction of the gradients, but does not take into account the magnitudes of the values. Matching the magnitude of the values is essential for gradient approximation. Otherwise, the optimization will be dominated by the large values, resulting in distortion of updating dummy data. One can add an additional term to the cosine similarity loss that takes into account the norm of the difference of the gradients [36], [37].

On the other hand, the L1 loss, i.e., the summed absolute errors, is a single-term metric that is robust to outliers, which is common in gradients. Additionally. it matches the magnitude of the gradient values and it is particularly compatible with models that utilize weights initialized from a normal distribution [15], such as the TCN architecture. Our choice is backed by empirical findings that show that using the L1-Norm leads to better reconstructions compared to the Cosine with or without the L2-Norm. The distance function is thus formulated as follows:

$$D(\nabla\widetilde{\mathbf{W}}, \nabla\mathbf{W}) = \left\|\nabla\widetilde{\mathbf{W}} - \nabla\mathbf{W}\right\|_1. \tag{7}$$

C. Regularizations: Periodicity, Trend and Quantile Bounds

In time series, there are data-specific characteristics that can be leveraged to regularize the reconstruction: periodicity and trend. The former refers to the repeating patterns or cycles in the data over a specific period, whereas the trend represents the long-term progression or direction in the data. Furthermore, the learned quantile ranges, outputted by f_{inv}, are used as soft bounds for the reconstructed data. We weight and combine those patterns into one regularization term, $\mathcal{R}_{TS}$:

$$\mathcal{R}_{TS} = \lambda_{\text{P}}\mathcal{L}^{p}_{\text{periodicity}} + \lambda_{\text{T}}\mathcal{L}_{\text{trend}} + \lambda^{obs}_{\text{Q}}\mathcal{L}^{obs}_{\text{bounds}} + \lambda^{tar}_{\text{Q}}\mathcal{L}^{tar}_{\text{bounds}}, \tag{8}$$

where λ_{P}, λ_{T}, λ^{obs}_{Q}, and λ^{tar}_{Q} are hyperparameters controlling the importance of periodicity, trend and learned quantile bounds regularizations, respectively.

1) Periodicity Regularization: Understanding the specific periodic nature of the data is crucial and the periodicity must be present in the combined observation and target sequence. To capture periodicity, we minimize the absolute error between each point and its corresponding point one period p apart. Here, p represents the number of timesteps for one full period or cycle. This regularization ensures that reconstructed data maintains the inherent cyclical patterns. Periodicity can vary widely depending on the dataset, exhibiting daily, weekly, or even yearly patterns. The regularization is defined as the mean absolute error between the periodic points:

$$\mathcal{L}^{p}_{\text{periodicity}}(\mathbf{S}) = \frac{1}{T-p}\sum_{t=1}^{T-p} |\mathbf{S}_t - \mathbf{S}_{t+p}|, \tag{9}$$

where p denotes the period length and T is the total number of timesteps in the sequence $\mathbf{S}$.

2) Trend Regularization: To enforce trend consistency, we fit a linear trend to the sequence and minimize the deviation of the actual sequence from this trend. This approach ensures that the trend is regularized over both the observations and targets combined, capturing the underlying trend dynamics in the data. The regularization is defined as the MAE between the sequence and the trend of the sequence, as such:

$$\mathcal{L}_{\text{trend}}(\mathbf{S}) = \|\mathbf{S} - \mathbf{S}_{\text{trend}}\|_1, \tag{10}$$

where the $\mathbf{S}_{\text{trend}}$ is calculated using linear regression over $\mathbf{S}$ as such: $\mathbf{S}_{\text{trend}} = \beta(s - \bar{s}) + \bar{\mathbf{S}}$. Here, s represents the time indices of the sequence, $\bar{s}$ is the mean of these time indices, $\bar{\mathbf{S}}$ is the mean of the sequence $\mathbf{S}$, and the slope β of the linear trend is given by: $\beta = \frac{\sum(s-\bar{s})\cdot(\mathbf{S}-\bar{\mathbf{S}})}{\sum(s-\bar{s})^2}$.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Comparison of sMAPE (↓) for baseline attacks and `TS-Inverse` on multiple datasets (Electricity 370, KDDCup, London Smartmeter, and Proprietary Dataset) and model architectures (CNN, FCN, TCN). Each attack is conducted over 5000 steps with a batch size $\mathcal{B} = 1$. The attack methods include 'DLG-LBFGS', 'DLG-Adam', 'InvG', 'DIA', 'LTI', '`TS-Inverse`', and '`TS-Inverse` $_{\text{one-shot}}$'. Experiments are run with seeds 10, 43, 28, 80, and 71, with the best performances highlighted in bold. Standard deviations are provided as subscripts.

	Dataset	Electricity 370		KDDCup		London Smartmeter		Proprietary Dataset	
Model	Attack Method	Observation	Target	Observation	Target	Observation	Target	Observation	Target
CNN	DLG-LBFGS	$0.632_{0.05}$	$0.001_{0.00}$	$1.454_{0.11}$	$0.030_{0.03}$	$1.429_{0.14}$	$0.023_{0.01}$	$1.294_{0.08}$	$0.011_{0.00}$
	DLG-Adam	$0.029_{0.03}$	$8.6e\text{-}05_{0.00}$	$1.349_{0.23}$	$0.029_{0.03}$	$0.993_{0.14}$	$0.006_{0.00}$	$1.013_{0.27}$	$0.005_{0.00}$
	InvG	$0.002_{0.00}$	$0.189_{0.13}$	$0.038_{0.02}$	$1.306_{0.30}$	$0.027_{0.00}$	$1.136_{0.26}$	$0.015_{0.00}$	$1.160_{0.19}$
	DIA	$0.002_{0.00}$	$0.542_{0.11}$	$0.073_{0.04}$	$1.293_{0.11}$	$0.094_{0.04}$	$1.501_{0.05}$	$0.018_{0.01}$	$1.484_{0.08}$
	LTI	$0.140_{0.04}$	$0.685_{0.07}$	$0.889_{0.15}$	$1.399_{0.28}$	$0.713_{0.18}$	$1.469_{0.06}$	$0.486_{0.05}$	$1.305_{0.09}$
	`TS-Inverse`	$\mathbf{0.002}_{0.00}$	$1.2e\text{-}05_{0.00}$	$\mathbf{0.017}_{0.01}$	$2.1e\text{-}04_{0.00}$	$\mathbf{8.1e\text{-}05}_{0.00}$	$2.4e\text{-}05_{0.00}$	$\mathbf{0.009}_{0.00}$	$7.1e\text{-}05_{0.00}$
	`TS-Inverse` $_{\text{one-shot}}$	$0.003_{0.00}$	$\mathbf{1.4e\text{-}07}_{0.00}$	$0.085_{0.05}$	$\mathbf{1.9e\text{-}06}_{0.00}$	$0.024_{0.01}$	$\mathbf{2.1e\text{-}06}_{0.00}$	$0.032_{0.01}$	$\mathbf{7.3e\text{-}07}_{0.00}$
FCN	DLG-LBFGS	$0.025_{0.01}$	$2.0e\text{-}04_{0.00}$	$0.168_{0.05}$	$0.002_{0.00}$	$0.498_{0.17}$	$0.003_{0.00}$	$0.181_{0.07}$	$0.002_{0.00}$
	DLG-Adam	$2.8e\text{-}06_{0.00}$	$1.2e\text{-}05_{0.00}$	$0.002_{0.00}$	$4.1e\text{-}04_{0.00}$	$0.004_{0.00}$	$2.9e\text{-}05_{0.00}$	$1.8e\text{-}05_{0.00}$	$3.5e\text{-}05_{0.00}$
	InvG	$4.7e\text{-}06_{0.00}$	$0.214_{0.15}$	$6.1e\text{-}05_{0.00}$	$1.261_{0.39}$	$7.8e\text{-}05_{0.00}$	$1.300_{0.10}$	$2.6e\text{-}05_{0.00}$	$1.108_{0.15}$
	DIA	$0.004_{0.00}$	$0.542_{0.10}$	$0.027_{0.02}$	$1.343_{0.19}$	$0.068_{0.05}$	$1.571_{0.07}$	$1.1e\text{-}04_{0.00}$	$1.551_{0.07}$
	LTI	$0.133_{0.03}$	$0.685_{0.07}$	$0.604_{0.17}$	$1.399_{0.28}$	$0.673_{0.07}$	$1.469_{0.06}$	$0.498_{0.03}$	$1.305_{0.09}$
	`TS-Inverse`	$\mathbf{1.7e\text{-}06}_{0.00}$	$8.4e\text{-}07_{0.00}$	$\mathbf{2.4e\text{-}06}_{0.00}$	$\mathbf{1.1e\text{-}06}_{0.00}$	$\mathbf{6.3e\text{-}06}_{0.00}$	$2.5e\text{-}06_{0.00}$	$\mathbf{6.4e\text{-}06}_{0.00}$	$4.4e\text{-}06_{0.00}$
	`TS-Inverse` $_{\text{one-shot}}$	$\mathbf{3.1e\text{-}07}_{0.00}$	$\mathbf{8.3e\text{-}08}_{0.00}$	$5.3e\text{-}06_{0.00}$	$1.5e\text{-}06_{0.00}$	$3.2e\text{-}05_{0.00}$	$\mathbf{1.4e\text{-}06}_{0.00}$	$6.6e\text{-}06_{0.00}$	$\mathbf{8.1e\text{-}07}_{0.00}$
TCN	DLG-LBFGS	$0.652_{0.06}$	$0.689_{0.04}$	$1.483_{0.06}$	$1.374_{0.28}$	$1.437_{0.10}$	$1.429_{0.09}$	$1.321_{0.13}$	$1.243_{0.09}$
	DLG-Adam	$0.726_{0.07}$	$0.362_{0.11}$	$1.320_{0.13}$	$1.072_{0.41}$	$1.136_{0.18}$	$1.079_{0.19}$	$1.141_{0.16}$	$1.028_{0.09}$
	InvG	$0.704_{0.05}$	$0.553_{0.20}$	$1.359_{0.16}$	$1.445_{0.19}$	$1.154_{0.18}$	$1.493_{0.11}$	$1.114_{0.12}$	$1.424_{0.11}$
	DIA	$1.088_{0.31}$	$0.719_{0.20}$	$0.728_{0.20}$	$1.229_{0.07}$	$1.076_{0.57}$	$1.576_{0.01}$	$1.282_{0.34}$	$1.473_{0.05}$
	LTI	$0.158_{0.05}$	$0.628_{0.06}$	$0.839_{0.13}$	$1.437_{0.30}$	$0.743_{0.08}$	$1.402_{0.06}$	$0.530_{0.13}$	$1.326_{0.09}$
	`TS-Inverse`	$0.097_{0.04}$	$0.007_{0.01}$	$\mathbf{0.509}_{0.13}$	$0.042_{0.05}$	$0.194_{0.22}$	$0.106_{0.19}$	$\mathbf{0.172}_{0.05}$	$0.021_{0.02}$
	`TS-Inverse` $_{\text{one-shot}}$	$\mathbf{0.089}_{0.03}$	$\mathbf{1.2e\text{-}07}_{0.00}$	$0.559_{0.21}$	$\mathbf{1.3e\text{-}06}_{0.00}$	$\mathbf{0.188}_{0.12}$	$\mathbf{1.8e\text{-}06}_{0.00}$	$0.251_{0.14}$	$\mathbf{7.6e\text{-}07}_{0.00}$

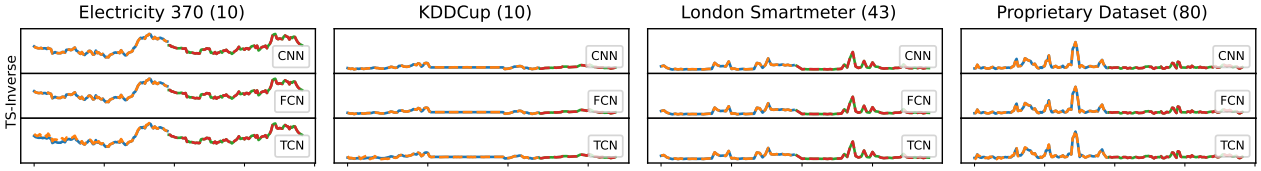


Fig. 5: `TS-Inverse` reconstruction results with FCN, CNN, and TCN model architectures on a sample from each dataset.

3) Quantile Bounds Regularization: The predictions made by the gradient inversion model are designed to function as soft bounds for the dummy data as illustrated in Figure 7. The bounds are defined as quantile pairs that are symmetrically spaced around the median. Within these bounds, no regularization is applied. However, if the dummy data falls outside this range, an absolute error is calculated for regularization. This approach intuitively ensures that the dummy data is encouraged to stay within realistic and statistically consistent bounds.

For a sequence $\mathbf{S}$ and bounds $\mathbf{S}_{\text{lower}}$ and $\mathbf{S}_{\text{upper}}$, the regularization loss is defined as:

$$\mathcal{L}_{\text{bounds}} = \sum_{q=1}^{Q} \left\| \max(0, \mathbf{S} - \mathbf{S}_{\text{upper}}) + \max(0, \mathbf{S}_{\text{lower}} - \mathbf{S}) \right\|_1, \tag{11}$$

where q indexes over the quantiles with $\mathbf{S}_{\text{lower}} = \mathbf{S}^{\tau_q}$ and $\mathbf{S}_{\text{upper}} = \mathbf{S}^{\tau_{(Q-q)}}$, respectively.

Putting Eq. (9) (10), and (11) into (8) and combining it with (7), `TS-Inverse` employs the following loss function to invert a batch of time series of observation and target from the client's gradients:

$$\mathcal{L}_{\text{total}} = D(\nabla\widetilde{\mathbf{W}}, \nabla\mathbf{W}) + \mathcal{R}_{TS}.$$

D. One-shot Target Reconstruction for $\mathcal{B} = 1$

In this section, we demonstrate how to reconstruct the target predictions of a fully connected layer f_{FC}, which in practice is the last layer of the forecasting model f. Given a fully connected layer f_{FC} with weights $\mathbf{W}$ and biases $\mathbf{b}$, and knowing the gradients $\nabla\mathbf{W}$ and $\nabla\mathbf{b}$ obtained with training inputs x and outputs y, we assume the objective function is the MSE $\mathcal{L}_{\text{MSE}} = \|\hat{y} - y\|_2$, where $\hat{y}$ are the layer's predictions. It is possible to reconstruct y one-shot for $\mathcal{B} = 1$ if $\nabla\mathbf{b}$ is invertible. The MSE loss is defined as: $\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$, where N is the number of elements. We can express the predicted output as $\hat{y} = \mathbf{W}x + \mathbf{b}$. The gradient with respect to the bias $\mathbf{b}$ and weights $\mathbf{W}$ are:

$$\nabla\mathbf{b} = \frac{2}{N}(\mathbf{W}x + \mathbf{b} - y) \tag{12}$$

$$\nabla\mathbf{W} = \frac{2}{N}(\mathbf{W}x + \mathbf{b} - y)x^T. \tag{13}$$

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

TABLE III: Dataset descriptions: sample frequency (FREQ.), observation length (O), future length (F), window size (W), and step size for attacked samples (S_{attack}) and for the auxilary samples (S_{aux}).

Dataset	FREQ.	H	F	W	S_{attack}	S_{aux}
Electricity 370	15 MIN	96	96	192	96	4
London SmartMeter	30 MIN	48	48	96	48	2
Proprietary	15 MIN	96	96	192	96	4
KDDCup 2018	HOUR	120	48	168	24	1

TABLE IV: Results of `TS-Inverse` without regularization for the TCN model architecture, measured using sMAPE (↓). Gradient distance functions include 'Cosine + L1', 'Cosine + L2', 'Cosine', 'L2', and 'L1'. Evaluations are performed on the Electricity 370 and London Smartmeter datasets for $\mathcal{B} \in [1, 2, 4]$. The experiments are run with seeds 10, 43, and 28, with the best performances highlighted in bold. Standard deviations are provided as subscripts.

	Dataset	Electricity 370		London Smartmeter	
$\mathcal{B}$	Grad. Distance D	Observation	Target	Observation	Target
1	Cosine + L1	$0.083_{0.01}$	**9.3e-08**$_{0.00}$	$0.212_{0.03}$	**2.3e-06**$_{0.00}$
	Cosine + L2	$0.290_{0.17}$	**9.3e-08**$_{0.00}$	$0.481_{0.30}$	**2.3e-06**$_{0.00}$
	Cosine	$0.377_{0.19}$	**9.3e-08**$_{0.00}$	$0.860_{0.56}$	**2.3e-06**$_{0.00}$
	L2	$0.236_{0.05}$	**9.3e-08**$_{0.00}$	$0.783_{0.49}$	**2.3e-06**$_{0.00}$
	L1	**0.081**$_{0.03}$	**9.3e-08**$_{0.00}$	**0.208**$_{0.14}$	**2.3e-06**$_{0.00}$
2	Cosine + L1	**0.362**$_{0.14}$	**0.053**$_{0.04}$	$0.301_{0.20}$	$0.261_{0.29}$
	Cosine + L2	$0.628_{0.24}$	$0.119_{0.03}$	$0.816_{0.36}$	$0.602_{0.46}$
	Cosine	$0.619_{0.18}$	$0.630_{0.40}$	$1.049_{0.25}$	$0.920_{0.33}$
	L2	$0.629_{0.32}$	$0.613_{0.40}$	$0.869_{0.43}$	$0.860_{0.36}$
	L1	$0.471_{0.08}$	$0.095_{0.06}$	**0.278**$_{0.20}$	**0.244**$_{0.27}$
4	Cosine + L1	$0.647_{0.11}$	$0.261_{0.17}$	$0.825_{0.35}$	**0.516**$_{0.40}$
	Cosine + L2	$0.715_{0.16}$	$0.901_{0.20}$	$1.231_{0.12}$	$0.881_{0.25}$
	Cosine	$0.738_{0.18}$	$0.818_{0.33}$	$1.132_{0.34}$	$0.933_{0.43}$
	L2	$0.742_{0.16}$	$0.811_{0.31}$	$1.264_{0.16}$	$1.040_{0.26}$
	L1	**0.554**$_{0.11}$	**0.131**$_{0.10}$	**0.777**$_{0.34}$	$0.539_{0.45}$

Equation (12) can be rearranged to solve for y:

$$y = \mathbf{W}x + \mathbf{b} - \nabla\mathbf{b} \cdot \frac{N}{2}. \tag{14}$$

Here, x is unknown, but it can be reconstructed by substituting y from (14) into (13), we get:

$$\begin{aligned}\nabla\mathbf{W} &= \frac{2}{N}\left(\mathbf{W}x + \mathbf{b} - \left(\mathbf{W}x + \mathbf{b} - \nabla\mathbf{b} \cdot \frac{N}{2}\right)\right)x^T \\ &= \nabla\mathbf{b} \cdot x^T.\end{aligned}$$

If $\nabla\mathbf{b}$ is invertible, such that $(\nabla\mathbf{b})^{-1}$ exists, then $\mathrm{x}^T = (\nabla\mathbf{b})^{-1}\nabla\mathbf{W}$. Finally, substituting x back into equation (14), we obtain:

$$y = \mathbf{W}(\nabla\mathbf{b})^{-1}\nabla\mathbf{W} + \mathbf{b} - \nabla\mathbf{b} \cdot \frac{N}{2}. \tag{15}$$

This demonstrates that the target y can be reconstructed in one shot for $\mathcal{B} = 1$, provided that $\nabla\mathbf{b}$ is invertible.

V. Evaluation

We first detail the experimental setup, followed by the attack accuracy of `TS-Inverse`, and an ablation study.

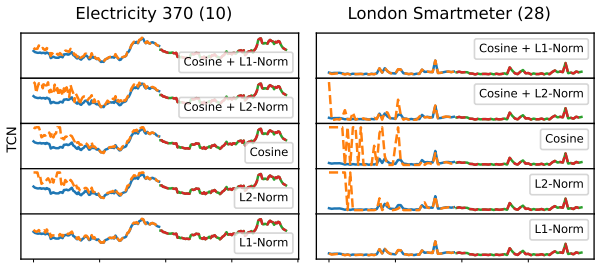


Fig. 6: `TS-Inverse`, reconstruction comparison on the TCN model architecture with different gradient distance functions and without regularization.

A. Experimental setup

1) Dataset: For the datasets we have selected a total of four datasets. Three electricity datasets: Electricity (370)[1], London Smartmeter[2], and a proprietary dataset. We have also opted for an additional dataset about air quality: KDDCup 2018 (Air Quality)[3]. Even though the air quality is not a privacy-sensitive dataset, it evaluates the method's ability to generalize. The energy datasets are sampled with 1 observation day and 1 future day to forecast. The KDDCup dataset has 5 observation days and 2 future days. The sample description of the datasets is summarized in Table III. All datasets are normalized using the min-max method bounding it between 0 and 1.

2) Data Pre-processing: We employ the rolling window method to preprocess the data, i.e., moving a window of fixed-width steps through the dataset, advancing by steps with each iteration. The data within this window is divided into observation and target segments $\mathbf{S}_{obs}$ and $\mathbf{S}_{tar}$ according to an observation length H and future length F. The result of the rolling window process is a dataset sample pairs of observations $\mathbf{S}_{obs}$ and targets $\mathbf{S}_{tar}$. The dataset-specific settings are described in Table III. The datasets are divided into training, validation, and test sets with ratios of 64%, 16%, and 20%, respectively, by recursively applying a 20% split ratio. For the auxiliary dataset $\mathcal{D}_{aux}$, the validation set is used.

3) Baselines: The baselines that have been evaluated are the DLG-LBFGS [11], DLG-Adam [11], InvG [14], DIA [19], and LTI [20]. The implementation and configuration details are consistent with their open-sourced repositories. Where LTI is trained for 250 epochs and the other methods attack with 5000 iterations.

4) Network Architectures: The FCN model has 3 FC layers with sigmoid acivation functions, the CNN model is based on LeNet with 1D Conv layers. The TCN architecture has a kernel size of 6, dilation factor of 2 and the number of levels is adjusted such that the receptive field is encompassing the observation sequence. The GRU-2-FCN and GRU-2-GRU both have the standard GRU implementation. All models apart

[1] https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014
[2] https://zenodo.org/records/4656091
[3] https://zenodo.org/records/4656756

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

from the GRU-2-GRU have a FC head that is outputting the final values. All hidden sizes of the architectures are 64.

The gradient inversion model f_{inv} has two modules for outputting observations and targets, each containing two residual blocks with hidden sizes of 768 and 512, respectively. Each residual block consists of fully connected layers, ReLU activations, batch normalization, and dropout for regularization, with an optional adaptation layer ensuring matching dimensions for identity mapping. Additionally, each module includes a fully connected layer that outputs the quantiles for its observation or target reconstruction. The model is trained for 75 epochs and the quantile levels are $\{0.1, 0.3, 0.7, 0.9\}$.

5) Evaluation Metrics: To measure the reconstruction quality the Symmetrical Mean Absolute Percentage Error (sMAPE ↓) is used, which is bounded between 0.0 and 2.0, and allows for better comparisons between datasets. This normalization is achieved by scaling the absolute difference between the actual values ($\mathbf{S}$) and the predicted values ($\hat{\mathbf{S}}$) by the sum of their absolute values. The formula for sMAPE is given by:

$$\text{sMAPE} = \frac{1}{N} \sum_{i=1}^{N} \frac{2 \left| \mathbf{S}_i - \hat{\mathbf{S}}_i \right|}{|\mathbf{S}_i| + \left| \hat{\mathbf{S}}_i \right|}$$

where N are the number of elements in sequence $\mathbf{S}$, and $|\cdot|$ is the absolute value function. When using other metrics, such as MSE and MAE, the ranks do not change.

B. Results Overview for `TS-Inverse`

The experimental results demonstrate that `TS-Inverse` outperforms the baseline methods in TS reconstruction across multiple datasets and architectures as shown in Table II. Specifically, `TS-Inverse` achieves the lowest sMAPE values for both observation and target sequences. Figures 5 and 7 illustrate how `TS-Inverse` captures underlying patterns and quantile bounds in the reconstructed sequences. The subsequent experiments investigate the impact of different gradient distance functions (Table IV) and assess the effectiveness of various regularization techniques, including quantile bounds, periodicity, and trend regularization (Tables V, VI, and VII). The combined effects of the various regularization techniques are displayed in Table VIII and Figure 10. These studies highlight the robustness and versatility of `TS-Inverse` in time series inversion tasks across different architectures and settings.

Effectiveness of the Proposed Gradient Distance Function

Table IV presents the sMAPE results for different gradient distance functions without additional regularization, focusing on capturing the effects of the gradient distances alone. The gradient distance functions tested include Cosine + L1-Norm, Cosine + L2-Norm, Cosine, L2, and L1. The results indicate that distance functions incorporating the L1-Norm perform better with lower sMAPE results compared to other functions. This trend is consistent across various datasets and is particularly evident in the TCN architecture, as shown in Figure 6, which displays individual reconstructions for $\mathcal{B} = 1$.

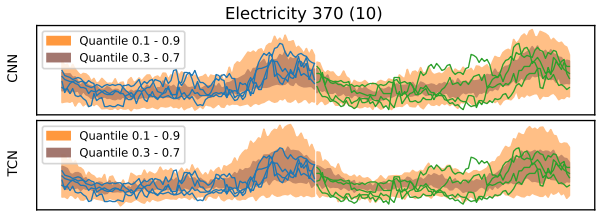


Fig. 7: `TS-Inverse` quantile predictions that function as prior in the learned regularization on Electricity 370 dataset for the CNN and TCN model architecture and $\mathcal{B} = 4$.

For the baseline architectures (FCN and CNN), the differences in sMAPE results among the distance functions are minimal.

Effectiveness of Regularization for Batch Reconstruction

The effects of the hyperparameters for the quantile bounds regularization are shown in Table V. The hyperparameters for the quantile bounds regularization are optimized through a grid search over the values $\lambda_Q^{obs} \in \{0, 0.1, 0.5, 1\}$ and $\lambda_Q^{tar} \in \{0, 0.1, 0.5, 1\}$. The results indicate that increasing the regularization term for observations leads to a decrease in the sMAPE metric, whereas the effect is less apparent for target sequences. From these results the optimal terms are found to be $\lambda_Q^{obs} = 1$ and $\lambda_Q^{tar} = 0.1$ for the Electricity 370 dataset.

Effectiveness of Periodicity Regularization

The impact of periodicity regularization on reconstruction performance is detailed in Table VI. We assessed various values of the periodicity regularization term $\lambda_{\text{P}} \in \{0.1, 0.5, 1, 2\}$ across three datasets: Electricity 370, London Smartmeter, and Proprietary. The results demonstrate that incorporating periodicity regularization significantly enhances the reconstruction quality, particularly for the Electricity 370 dataset. Specifically, setting $\lambda_{\text{P}} = 2$ yields the lowest sMAPE scores for both observations and targets, indicating superior reconstruction accuracy. This improvement is visually corroborated in Figure 8, which shows that the reconstructed sequences more closely align with the original data when periodicity regularization is applied. Notably, the optimal value of λ_{P} varies across datasets; for the London Smartmeter dataset, the best performance is achieved at $\lambda_{\text{P}} = 0.5$, while for the Proprietary dataset, $\lambda_{\text{P}} = 1$ yields the most favorable results. These findings suggest that the effectiveness of periodicity regularization is dataset-dependent, and appropriate tuning of λ_{P} is essential for maximizing reconstruction performance.

Effectiveness of Trend Regularization

We further explored the influence of trend regularization on the reconstruction results, as presented in Table VII. By varying the trend regularization parameter $\lambda_{\text{T}} \in \{0.1, 0.5, 1, 2\}$, we evaluated its effect on the same three datasets. The results indicate that trend regularization enhances the model's ability to capture underlying trends in the data, leading to improved reconstruction accuracy. For the Electricity 370 dataset, the

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

TABLE V: TS-Inverse results measured in sMAPE (↓) for the Electricity 370 dataset using quantile bounds regularization on observations (λ_Q^{obs}) and targets (λ_Q^{tar}). Results are shown for different hyperparameter values with the best performances in bold. Experiments are run with seeds 10 and 43, and the standard deviations are provided as subscripts.

λ_Q^{obs}	0		0.1		0.5		1	
λ_Q^{tar}	Observation	Target	Observation	Target	Observation	Target	Observation	Target
0	$0.515_{0.11}$	$0.068_{0.05}$	$0.504_{0.12}$	$0.185_{0.11}$	$0.207_{0.08}$	$0.061_{0.06}$	$0.203_{0.05}$	$0.060_{0.03}$
0.1	$0.567_{0.17}$	$0.048_{0.01}$	$0.449_{0.14}$	$0.105_{0.05}$	$0.284_{0.02}$	$0.237_{0.11}$	$\mathbf{0.144}_{0.03}$	$\mathbf{0.024}_{0.02}$
0.5	$0.568_{0.18}$	$0.067_{0.02}$	$0.368_{0.21}$	$0.026_{0.02}$	$0.259_{0.08}$	$0.112_{0.08}$	$0.207_{0.04}$	$0.131_{0.06}$
1	$0.514_{0.07}$	$0.053_{0.03}$	$0.522_{0.22}$	$0.342_{0.26}$	$0.226_{0.12}$	$0.040_{0.03}$	$0.196_{0.04}$	$0.077_{0.03}$

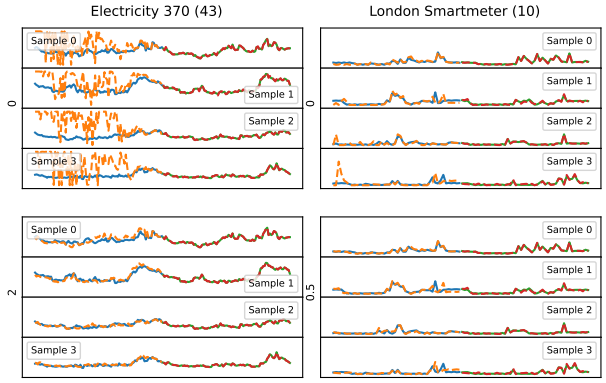


Fig. 8: TS-Inverse: The best periodicity regularization λ_P compared to no regularization. Rows are λ_P and Columns the dataset. Model: TCN, Datasets: Electricity 370, London Smartmeter, Batch size $\mathcal{B} = 4$, (seeds 10 and 43).

TABLE VI: TS-Inverse results measured in sMAPE (↓) for periodicity regularization with different λ_P values using the TCN model. The results are shown for three datasets: Electricity 370, London Smartmeter, and Proprietary. The best performances are highlighted in bold. Experiments are conducted with a batch size of $\mathcal{B} = 4$ using seeds 10 and 43, with standard deviations provided as subscripts.

Dataset	Elec. 370		London Smart.		Proprietary	
λ_P	Observation	Target	Observation	Target	Observation	Target
0	$0.515_{0.11}$	$0.068_{0.05}$	$0.603_{0.29}$	$0.250_{0.22}$	$0.956_{0.12}$	$0.197_{0.03}$
0.1	$0.400_{0.04}$	$0.146_{0.12}$	$0.838_{0.19}$	$0.620_{0.22}$	$0.670_{0.02}$	$0.369_{0.07}$
0.5	$0.306_{0.16}$	$0.219_{0.18}$	$\mathbf{0.467}_{0.18}$	$\mathbf{0.250}_{0.21}$	$0.396_{0.02}$	$0.214_{0.03}$
1	$0.355_{0.16}$	$0.375_{0.13}$	$0.714_{0.24}$	$0.538_{0.43}$	$\mathbf{0.372}_{0.01}$	$\mathbf{0.167}_{0.07}$
2	$\mathbf{0.163}_{0.05}$	$\mathbf{0.063}_{0.05}$	$0.658_{0.06}$	$0.289_{0.14}$	$0.452_{0.02}$	$0.193_{0.01}$

lowest sMAPE scores for observations are obtained at $\lambda_T = 2$, demonstrating the positive impact of stronger trend regularization. Figure 9 visually illustrates this improvement, showing that the reconstructed sequences better reflect the original data's trend when the regularization is applied. Similar trends are observed for the London Smartmeter and Proprietary datasets, with optimal performances at $\lambda_T = 0.5$ and $\lambda_T = 1$, respectively. These results underscore the importance of incorporating trend regularization in the reconstruction process, especially for datasets where trend components are prominent. Overall, careful tuning of λ_T is crucial to effectively leverage trend information and enhance reconstruction performance across different datasets.

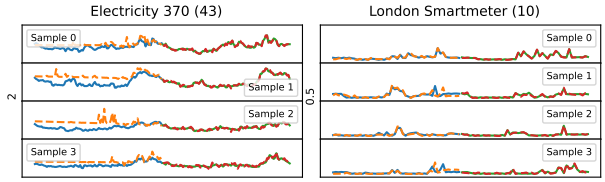


Fig. 9: TS-Inverse: The best trend regularization λ_T. Model: TCN, Datasets: Electricity 370, London Smartmeter, and Proprietary, Batch size $\mathcal{B} = 4$, (seeds 10 and 43)

TABLE VII: TS-Inverse results measured in sMAPE (↓) for trend regularization with different λ_T values using the TCN model. The results are shown for three datasets: Electricity 370, London Smartmeter, and Proprietary. The best performances are highlighted in bold. Experiments are conducted with a batch size of $\mathcal{B} = 4$ using seeds 10 and 43, with standard deviations provided as subscripts.

Dataset	Elec. 370		London Smart.		Proprietary	
λ_T	Observation	Target	Observation	Target	Observation	Target
0	$0.515_{0.11}$	$0.068_{0.05}$	$0.603_{0.29}$	$0.250_{0.22}$	$0.956_{0.12}$	$0.197_{0.03}$
0.1	$0.505_{0.15}$	$\mathbf{0.054}_{0.02}$	$0.853_{0.08}$	$0.416_{0.06}$	$0.955_{0.02}$	$0.443_{0.27}$
0.5	$0.392_{0.08}$	$0.152_{0.12}$	$\mathbf{0.495}_{0.21}$	$\mathbf{0.247}_{0.23}$	$0.736_{0.10}$	$0.482_{0.30}$
1	$0.419_{0.17}$	$0.324_{0.21}$	$0.685_{0.04}$	$0.568_{0.14}$	$\mathbf{0.475}_{0.12}$	$\mathbf{0.145}_{0.02}$
2	$\mathbf{0.281}_{0.05}$	$0.124_{0.06}$	$0.633_{0.01}$	$0.250_{0.21}$	$0.515_{0.06}$	$0.242_{0.07}$

Effectiveness of Combined Regularization

We investigated the combined effects of periodicity and trend regularization on reconstruction performance, as summarized in Table VIII. The parameters for both regularization terms, λ_P and λ_T, were tested with values $\{0.1, 0.5, 1\}$. The results indicate that the integration of periodicity and trend regularization significantly enhances reconstruction accuracy. Specifically, the best performance was achieved when both regularization methods were applied together, especially in the presence of quantile bounds. Figure 10 visually demonstrates the improvements in reconstructed sequences with combined regularization, highlighting the advantages of this approach in capturing complex patterns in the data. The findings underscore the necessity of tuning both λ_P and λ_T to optimize reconstruction across different datasets.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII: `TS-Inverse` results measured in sMAPE (↓) for the Electricity 370 dataset using periodicity and trend regularization ($\mathcal{R}_P + \mathcal{R}_T$) with and without quantile bounds regularization ($\mathcal{R}_Q$). Results are shown for different regularization parameter values (λ_P and λ_T) with the best performances in bold. Experiments are ran with seeds 10 and 43 and the standard deviations are provided as subscripts.

	λ_P	0.5		1		2	
λ_T	Regularization	Observation	Target	Observation	Target	Observation	Target
0.5	$\mathcal{R}_P + \mathcal{R}_T$	**0.154**$_{0.03}$	**0.118**$_{0.10}$	0.155$_{0.05}$	0.086$_{0.06}$	0.350$_{0.14}$	0.302$_{0.18}$
	$\mathcal{R}_P + \mathcal{R}_T + \mathcal{R}_Q$	0.273$_{0.12}$	0.287$_{0.19}$	**0.124**$_{0.00}$	**0.039**$_{0.03}$	0.174$_{0.02}$	0.199$_{0.01}$
1	$\mathcal{R}_P + \mathcal{R}_T$	0.197$_{0.04}$	0.075$_{0.02}$	0.350$_{0.20}$	0.330$_{0.29}$	0.162$_{0.01}$	0.081$_{0.01}$
	$\mathcal{R}_P + \mathcal{R}_T + \mathcal{R}_Q$	0.161$_{0.01}$	0.059$_{0.02}$	0.141$_{0.01}$	0.080$_{0.05}$	0.158$_{0.01}$	0.098$_{0.03}$
2	$\mathcal{R}_P + \mathcal{R}_T$	0.217$_{0.02}$	0.058$_{0.01}$	0.184$_{0.02}$	0.102$_{0.01}$	0.314$_{0.15}$	0.254$_{0.21}$
	$\mathcal{R}_P + \mathcal{R}_T + \mathcal{R}_Q$	0.184$_{0.02}$	0.082$_{0.02}$	0.132$_{0.02}$	0.051$_{0.04}$	0.124$_{0.03}$	0.114$_{0.09}$

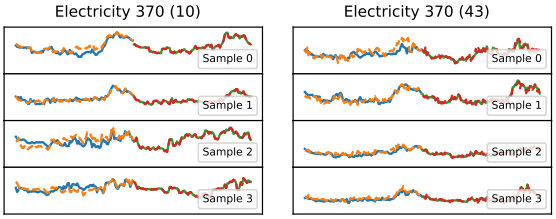


Fig. 10: `TS-Inverse` reconstructions with all regularization applied for the TCN model architecture and $\mathcal{B} = 4$.

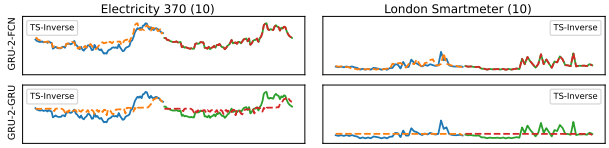


Fig. 11: `TS-Inverse` reconstructions on Electricity 370 and London Smartmeter datasets for the GRU-2-FCN and GRU-2-GRU architectures $\mathcal{B} = 1$ with $\lambda_P = 1$, $\lambda_T = 0.5$, $\lambda_Q^{obs} = 1$, and $\lambda_Q^{tar} = 0.1$.

C. Defenses

In addition to conventional FL defense strategies, such as pruning [38], differential privacy [39], and sign compression [40], we explore the defense capabilities of GRU-based forecasting models.

Foremost, Table IX illustrates the outcomes of these three defense mechanisms. The findings suggest that the LTI attack shows greater performance in attacking the defense mechanisms than the optimization-based attacks. We hypothesis that this is due to the fact that the Inversion model is trained with gradients on which the defense mechanisms are applied, and thus the model could potentially learn the defense characteristics and counter this.

Furthermore, we evaluate using GRU in the forecasting model to counter `TS-Inverse`. GRU-based forecasting models are more robust against attacks due to the many-to-one mapping caused by the recurrent design. We attack the GRU-2-FCN and GRU-2-GRU models, with reconstruction results shown in Figure 11. The results show that the targets associated with the FCN part remain vulnerable. We hypothesize that regularization causes the values to transfer to the observations, as the observations of the GRU-2-FCN architecture closely resemble the targets. This is backed by the results of employing a GRU-2-GRU architecture, which prevents detailed reconstruction of the observations and targets. However, the attack does seem to be able to extract the trend from the gradients. Overall, we the GRU-based architectures enhance the robustness against privacy attacks because of the many-to-one mapping design.

TABLE IX: Baselines and `TS-Inverse` results measured in sMAPE (↓) for the Electricity 370 dataset and FCN model under various defense strategies. Results are presented for Observations and Targets, with the best performances highlighted in bold. Experiments were conducted with seeds 10, 43 and 28, and the standard deviations are indicated as subscripts.

	Dataset	Electricity 370	
Defense	Attack Method	Observation	Target
none	DLG-LBFGS	0.033$_{0.00}$	3.1e-04$_{0.00}$
	DLG-Adam	**1.5e-06**$_{0.00}$	8.6e-06$_{0.00}$
	InvG	4.4e-06$_{0.00}$	0.279$_{0.06}$
	DIA	0.004$_{0.00}$	0.867$_{0.09}$
	LTI	0.069$_{0.01}$	0.042$_{0.01}$
	`TS-Inverse`	1.1e-05$_{0.00}$	**8.5e-08**$_{0.00}$
gauss	DLG-LBFGS	1.998$_{0.00}$	1.417$_{0.05}$
	DLG-Adam	1.910$_{0.01}$	1.457$_{0.04}$
	InvG	1.810$_{0.02}$	1.213$_{0.07}$
	DIA	1.805$_{0.04}$	1.377$_{0.04}$
	LTI	**0.222**$_{0.02}$	**0.231**$_{0.07}$
	`TS-Inverse`	0.593$_{0.04}$	1.179$_{0.13}$
prune	DLG-LBFGS	0.034$_{0.00}$	0.009$_{0.00}$
	DLG-Adam	**0.003**$_{0.00}$	0.009$_{0.00}$
	InvG	0.003$_{0.00}$	0.284$_{0.05}$
	DIA	0.003$_{0.00}$	0.861$_{0.09}$
	LTI	0.071$_{0.01}$	0.047$_{0.01}$
	`TS-Inverse`	0.011$_{0.01}$	**0.009**$_{0.00}$
sign	DLG-LBFGS	1.963$_{0.03}$	1.989$_{0.00}$
	DLG-Adam	1.929$_{0.01}$	1.951$_{0.02}$
	InvG	1.672$_{0.01}$	0.816$_{0.12}$
	DIA	1.672$_{0.01}$	1.261$_{0.13}$
	LTI	**0.324**$_{0.07}$	**0.184**$_{0.06}$
	`TS-Inverse`	1.049$_{0.06}$	1.230$_{0.14}$

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

TABLE X: Comparison of related gradient inversion attacks and `TS-Inverse`. The space indicates whether the attack optimizes the inputs ($\mathbf{x}$), outputs ($\mathbf{y}$), latent ($\mathbf{z}$), or generative model parameters (G_θ). The regularization terms mentioned are Total Variation, Clipping, Scaling, Dropout Masks, Batch and Group Normalization, Softmax Relaxation, Periodicity, Trend and Quantile Bounds. The FL Training Loss indicates with which loss the global model is trained.

Approach	Attack Name / Source	Type of attack	Space	Gradient Distance	Regularization	FL Training Loss
Pure	DLG [11]	Optimization	$(\mathbf{x}, \mathbf{y})$	L2	-	Cross Entropy
	iDLG [13]	Optimization	$(\mathbf{x})$	L2	-	Cross Entropy
	Inverting Gradients (InvG) [14]	Optimization	$(\mathbf{x})$	Cosine	TV	Cross Entropy
	SAPAG [12]	Optimization	$(\mathbf{x}, \mathbf{y})$	Weighted Gaussian Kernel	-	Cross Entropy
	R-GAP [41]	Recursive	$(\mathbf{x})$	-	-	Cross Entropy
	AGIC [42]	Optimization	$(\mathbf{x}, \mathbf{y})$	Cosine	TV	Cross Entropy
	DIA [19]	Optimization	$(\mathbf{x})$	Cosine	TV, Mask	Cross Entropy
	General DL [18], [43]	Optimization	$(\mathbf{x})$	L2	TV, Clip, Scale	Cross Entropy
	TAG [15]	Optimization	$(\mathbf{x}, \mathbf{y})$	L2 + α L1	-	Causal (Cross Ent.)
	TabLeak [44]	Optimization	$(\mathbf{x})$	Cosine	Softmax Relaxation, Ensemble	Cross Entropy
Informed	GradInversion [16]	(Latent) Optimization	$(\mathbf{z}, \mathbf{x})$	L2	TV, BN & Group	Cross Entropy
	GIAS / GIML [17]	(Latent) Optimization	$(\mathbf{z}, G_\theta)$	Cosine	TV, L2	Cross Entropy
	LTI [20]	Learning	$(\mathbf{x})$	Permutation Invariant CE or L2	-	-
	`TS-Inverse` (this work)	Learning & Optimization	$(\mathbf{x}, \mathbf{y})$	L1	Periodicity, Trend & QB	MSE

VI. Related Studies

Existing work on inversion attacks can be categorized into two types: pure optimization-based and informed optimization-based as presented in Table X. Both categories optimize dummy data such that their gradients match the observed gradients. The informed optimization-based attacks further leverage external knowledge (e.g., generative models) to assist the reconstruction process. The exploration of gradient inversion attacks reveals a rich and evolving landscape of techniques aimed at reconstructing training data from gradients. The foundational work "Deep Leakage from Gradients" (DLG) [11] set the stage by demonstrating the feasibility of these attacks using Euclidean distance minimization. The subsequent enhancement, "Improved Deep Leakage from Gradients" (iDLG) [13], advanced this by analytically reconstructing training labels, though it remained limited to a batch size of one. "Inverting Gradients" (InvG) [14] introduced cosine similarity and total variation regularization, improving the attack's robustness, particularly for image data. The "Self-Adaptive Privacy Attack from Gradients" (SAPAG) [12] tackled convergence issues arising from different weight initializations, further refining the optimization process with a Gaussian kernel-based distance function. The "Recursive Gradient Attack on Privacy" (R-GAP) [41] deviated from optimization-based methods, offering a faster, deterministic solution by recursively solving linear equations, albeit limited to single-sample batches. "See through Gradients" [16] and "Gradient Inversion with Generative Image Prior" [17] provided significant advancements in reconstruction quality and scalability to larger batches, leveraging auxiliary regularizations and generative models. The "Gradient Attack on Transformer-based Language Models" (TAG) [15] addressed the unique challenges of NLP models, using combined Euclidean and Manhattan distances for gradient comparison. The "Approximate Gradient Inversion Attack" (AGIC) [42] efficiently approximated multiple gradient updates in federated averaging, enhancing practicality for real-world federated learning setups. The "Dropout Inversion Attack" (DIA) [19] extended the attack surface by accounting for dropout layers, effectively neutralizing this common defense mechanism. "Improved Gradient Inversion Attacks and Defenses" [43] proposed a comprehensive framework addressing both FedSGD and FedAvg strategies, introducing zero-shot batch label inference and auxiliary regularizations for realistic image restoration. "Learning to Invert" (LTI) [20] presented a shift by using a model trained on an auxiliary dataset to predict client samples from gradients, overcoming traditional defenses like differential privacy. "Tabular Data Leakage" (TabLeak) [44] innovatively tackled the complexities of reconstructing tabular data, employing softmax relaxation and pooled ensembling for improved accuracy and reliability.

VII. Conclusion

This study presents the first empirical analysis and effective algorithm, `TS-Inverse`, of reconstructing TS data for federated TSF models. Our empirical analysis demonstrates that existing inversion methods, developed for image and text classification, are less effective for TSF due to the mismatch between gradient distance and the gradient magnitudes and TS-specific model architecture, e.g., TCN & GRU. To address these challenges, we propose `TS-Inverse`, which leverages a gradient inversion model and TS-specific characteristics as regularization during reconstruction. Specifically, the proposed TS-regularization include periodicity and trend. Furthermore, the gradient inversion model is trained with an auxiliary dataset to map gradients to quantile bounds of observations and targets. These bounds are also used to regularize the reconstruction. We extensively evaluate `TS-Inverse` on four data sets against five baselines, showing a 2x-10x improvement in reconstruction sMAPE and indicating the potential of GRU-based architecture in defending against inversion attacks.

Acknowledgments

This research was partly funded by the SNSF project, PrivGSyn 200021E_229204.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, A. Singh and X. (Zhu, Eds., ser. Proceedings of Machine Learning Research, vol. 54, PMLR, 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html (visited on 06/14/2024).

[2] J. Zhu, H. Dong, W. Zheng, S. Li, Y. Huang, and L. Xi, "Review and prospect of data-driven techniques for load forecasting in integrated energy systems," *Applied Energy*, vol. 321, p. 119 269, 2022, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2022.119269. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261922006262 (visited on 06/14/2024).

[3] N. Ahmad, Y. Ghadi, M. Adnan, and M. Ali, "Load forecasting techniques for power system: Research challenges and survey," *IEEE Access*, vol. 10, pp. 71 054–71 090, 2022. DOI: 10.1109/ACCESS.2022.3187839. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3187839 (visited on 01/10/2024).

[4] S. Čaušević, S. Sharma, S. B. Aziza, A. van der Veen, and E. Lazovik, "Lv grid state estimation using local flexible assets: A federated learning approach," in *27th International Conference on Electricity Distribution (CIRED 2023)*, 2023, pp. 1045–1049. DOI: 10.1049/icp.2023.0624. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10267781 (visited on 01/10/2024).

[5] S. Čaušević, R. Snijders, G. Pingen, and et al., "Flexibility prediction in smart grids: Making a case for federated learning," in *CIRED 2021 - The 26th International Conference and Exhibition on Electricity Distribution*, 2021, pp. 1983–1987. DOI: 10.1049/icp.2021.2196. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9692324 (visited on 01/10/2024).

[6] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the GDPR perspective," *Comput. Secur.*, vol. 110, p. 102 402, 2021. DOI: 10.1016/J.COSE.2021.102402. [Online]. Available: https://doi.org/10.1016/j.cose.2021.102402 (visited on 06/14/2024).

[7] L. Lyu, H. Yu, X. Ma, *et al.*, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 7, pp. 8726–8746, 2024. DOI: 10.1109/TNNLS.2022.3216981. [Online]. Available: https://doi.org/10.1109/TNNLS.2022.3216981 (visited on 01/22/2024).

[8] N. R. Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Inf. Fusion*, vol. 90, pp. 148–173, 2023. DOI: 10.1016/J.INFFUS.2022.09.011. [Online]. Available: https://doi.org/10.1016/j.inffus.2022.09.011 (visited on 06/05/2024).

[9] D. C. Nguyen, Q. Pham, P. N. Pathirana, *et al.*, "Federated learning for smart healthcare: A survey," *ACM Comput. Surv.*, vol. 55, no. 3, 60:1–60:37, 2023. DOI: 10.1145/3501296. [Online]. Available: https://doi.org/10.1145/3501296 (visited on 09/16/2024).

[10] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 7232–7241. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/3b3fff6463464959dcd1b68d0320f781-Abstract.html (visited on 06/14/2024).

[11] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 14 747–14 756. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html (visited on 11/10/2023).

[12] Y. Wang, J. Deng, D. Guo, *et al.*, "SAPAG: A self-adaptive privacy attack from gradients," *CoRR*, vol. abs/2009.06228, 2020. arXiv: 2009.06228. [Online]. Available: https://arxiv.org/abs/2009.06228 (visited on 01/18/2024).

[13] B. Zhao, K. R. Mopuri, and H. Bilen, "iDLG: Improved Deep Leakage from Gradients," *CoRR*, vol. abs/2001.02610, 2020. arXiv: 2001.02610. [Online]. Available: http://arxiv.org/abs/2001.02610 (visited on 11/10/2023).

[14] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?" In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html.

[15] J. Deng, Y. Wang, J. Li, *et al.*, "TAG: gradient attack on transformer-based language models," in *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds., Association for Computational Linguistics, 2021, pp. 3600–3610. DOI: 10.18653/V1/

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

2021.FINDINGS-EMNLP.305. [Online]. Available: https://doi.org/10.18653/v1/2021.findings-emnlp.305 (visited on 02/05/2024).

[16] H. Yin, A. Mallya, A. Vahdat, J. M. Álvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, Computer Vision Foundation / IEEE, 2021, pp. 16 337–16 346. DOI: 10.1109/CVPR46437.2021.01607. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Yin%5C_See%5C_Through%5C_Gradients%5C_Image%5C_Batch%5C_Recovery%5C_via%5C_GradInversion%5C_CVPR%5C_2021%5C_paper.html (visited on 11/10/2023).

[17] J. Jeon, J. Kim, K. Lee, S. Oh, and J. Ok, "Gradient inversion with generative image prior," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 29 898–29 908. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/fa84632d742f2729dc32ce8cb5d49733-Abstract.html (visited on 01/18/2024).

[18] J. Geng, Y. Mou, F. Li, *et al.*, "Towards general deep leakage in federated learning," *CoRR*, vol. abs/2110.09074, 2021. arXiv: 2110.09074. [Online]. Available: https://arxiv.org/abs/2110.09074 (visited on 02/01/2024).

[19] D. Scheliga, P. Maeder, and M. Seeland, "Dropout is NOT all you need to prevent gradient leakage," in *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, B. Williams, Y. Chen, and J. Neville, Eds., AAAI Press, 2023, pp. 9733–9741. DOI: 10.1609/AAAI.V37I8.26163. [Online]. Available: https://doi.org/10.1609/aaai.v37i8.26163 (visited on 04/22/2024).

[20] R. Wu, X. Chen, C. Guo, and K. Q. Weinberger, "Learning to invert: Simple adaptive attacks for gradient inversion in federated learning," in *Uncertainty in Artificial Intelligence, UAI 2023, July 31 - 4 August 2023, Pittsburgh, PA, USA*, R. J. Evans and I. Shpitser, Eds., ser. Proceedings of Machine Learning Research, vol. 216, PMLR, 2023, pp. 2293–2303. [Online]. Available: https://proceedings.mlr.press/v216/wu23a.html (visited on 11/10/2023).

[21] K. Cho, B. van Merrienboer, Ç. Gülçehre, *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, A. Moschitti, B. Pang, and W. Daelemans, Eds., ACL, 2014, pp. 1724–1734. DOI: 10.3115/V1/D14-1179. [Online]. Available: https://doi.org/10.3115/v1/d14-1179 (visited on 05/01/2024).

[22] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018. arXiv: 1803.01271. [Online]. Available: http://arxiv.org/abs/1803.01271 (visited on 01/29/2024).

[23] M. N. Fekri, H. Patel, K. Grolinger, and V. Sharma, "Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network," *Applied Energy*, vol. 282, p. 116 177, Jan. 2021, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2020.116177. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261920315804 (visited on 01/17/2024).

[24] M. Savi and F. Olivadese, "Short-term energy consumption forecasting at the edge: A federated learning approach," *IEEE Access*, vol. 9, pp. 95 949–95 969, 2021. DOI: 10.1109/ACCESS.2021.3094089. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3094089 (visited on 10/11/2023).

[25] A. Duttagupta, J. Zhao, and S. Shreejith, "Exploring lightweight federated learning for distributed load forecasting," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm IEEE, Glasgow, United Kingdom, October 31 - November 3, 2023*, IEEE, 2023, pp. 1–6. DOI: 10.1109/SMARTGRIDCOMM57358.2023.10333889. [Online]. Available: https://doi.org/10.1109/SmartGridComm57358.2023.10333889.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, ISSN: 1476-4687. DOI: 10.1038/323533a0. [Online]. Available: https://www.nature.com/articles/323533a0 (visited on 06/08/2024).

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6795963 (visited on 06/08/2024).

[28] M. A. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid CNN-LSTM model for short-term individual household load forecasting," *IEEE Access*, vol. 8, pp. 180 544–180 557, 2020. DOI: 10.1109/ACCESS.2020.3028281. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.3028281 (visited on 12/19/2023).

[29] P. Lara-Benítez, M. Carranza-García, J. M. Luna-Romera, and J. C. Riquelme, "Temporal convolutional networks applied to energy-related time series forecasting," *Applied Sciences*, vol. 10, no. 7, p. 2322, Jan. 2020, ISSN: 2076-3417. DOI: 10.3390/app10072322.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.

[Online]. Available: https://www.mdpi.com/2076-3417/10/7/2322 (visited on 01/22/2024).

[30] H. Bousbiat, R. Bousselidj, Y. Himeur, *et al.*, "Crossing roads of federated learning and smart grids: Overview, challenges, and perspectives," *CoRR*, vol. abs/2304.08602, 2023. DOI: 10.48550/ARXIV.2304.08602. arXiv: 2304.08602. [Online]. Available: https://doi.org/10.48550/arXiv.2304.08602 (visited on 10/11/2023).

[31] J. D. Fernández, S. P. Menci, C. M. Lee, A. Rieger, and G. Fridgen, "Privacy-preserving federated learning for residential short-term load forecasting," *Applied Energy*, vol. 326, p. 119 915, Nov. 2022, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2022.119915. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261922011722 (visited on 06/05/2024).

[32] E. U. Soykan, Z. Bilgin, M. A. Ersoy, and E. Tomur, "Differentially private deep learning for load forecasting on smart grid," in *2019 IEEE Globecom Workshops, Waikoloa, HI, USA, December 9-13, 2019*, IEEE, 2019, pp. 1–6. DOI: 10.1109/GCWKSHPS45667.2019.9024520. [Online]. Available: https://doi.org/10.1109/GCWkshps45667.2019.9024520 (visited on 06/05/2024).

[33] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, IEEE, 2019, pp. 739–753. DOI: 10.1109/SP.2019.00065. [Online]. Available: https://doi.org/10.1109/SP.2019.00065 (visited on 06/05/2024).

[34] F. Mo, A. Borovykh, M. Malekzadeh, H. Haddadi, and S. Demetriou, "Layer-wise characterization of latent information leakage in federated learning," *CoRR*, vol. abs/2010.08762, 2020. arXiv: 2010.08762. [Online]. Available: https://arxiv.org/abs/2010.08762 (visited on 06/05/2024).

[35] Y. Zheng, "Dropout against deep leakage from gradients," *CoRR*, vol. abs/2108.11106, 2021. arXiv: 2108.11106. [Online]. Available: https://arxiv.org/abs/2108.11106 (visited on 04/22/2024).

[36] Z. Jiang, J. Gu, M. Liu, and D. Z. Pan, "Delving into effective gradient matching for dataset condensation," in *IEEE International Conference on Omni-layer Intelligent Systems, COINS 2023, Berlin, Germany, July 23-25, 2023*, IEEE, 2023, pp. 1–6. DOI: 10.1109/COINS57856.2023.10189244. [Online]. Available: https://doi.org/10.1109/COINS57856.2023.10189244 (visited on 06/03/2024).

[37] T. Zhang, Y. Zhang, K. Wang, *et al.*, "Two trades is not baffled: Condensing graph via crafting rational gradient matching," *CoRR*, vol. abs/2402.04924, 2024. DOI: 10.48550/ARXIV.2402.04924. arXiv: 2402.04924. [Online]. Available: https://doi.org/10.48550/arXiv.2402.04924 (visited on 06/04/2024).

[38] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, M. Palmer, R. Hwa, and S. Riedel, Eds., Association for Computational Linguistics, 2017, pp. 440–445. DOI: 10.18653/V1/D17-1045. [Online]. Available: https://doi.org/10.18653/v1/d17-1045.

[39] M. Abadi, A. Chu, I. Goodfellow, *et al.*, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16, Vienna, Austria: Association for Computing Machinery, 2016, pp. 308–318, ISBN: 9781450341394. DOI: 10.1145/2976749.2978318. [Online]. Available: https://doi.org/10.1145/2976749.2978318.

[40] J. Bernstein, Y. Wang, K. Azizzadenesheli, and A. Anandkumar, "SIGNSGD: compressed optimisation for non-convex problems," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, J. G. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 559–568. [Online]. Available: http://proceedings.mlr.press/v80/bernstein18a.html.

[41] J. Zhu and M. B. Blaschko, "R-GAP: recursive gradient attack on privacy," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=RSU17UoKfJF.

[42] J. Xu, C. Hong, J. Huang, L. Y. Chen, and J. Decouchant, "AGIC: approximate gradient inversion attack on federated learning," in *41st International Symposium on Reliable Distributed Systems, SRDS 2022, Vienna, Austria, September 19-22, 2022*, IEEE, 2022, pp. 12–22. DOI: 10.1109/SRDS55811.2022.00012. [Online]. Available: https://doi.org/10.1109/SRDS55811.2022.00012.

[43] J. Geng, Y. Mou, Q. Li, *et al.*, "Improved gradient inversion attacks and defenses in federated learning," *IEEE Trans. Big Data*, vol. 10, no. 6, pp. 839–850, 2024. DOI: 10.1109/TBDATA.2023.3239116. [Online]. Available: https://doi.org/10.1109/TBDATA.2023.3239116 (visited on 11/10/2023).

[44] M. Vero, M. Balunovic, D. I. Dimitrov, and M. T. Vechev, "Tableak: Tabular data leakage in federated learning," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 2023, pp. 35 051–35 083. [Online]. Available: https://proceedings.mlr.press/v202/vero23a.html.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 06,2026 at 12:44:44 UTC from IEEE Xplore. Restrictions apply.