Robust Federated Learning with Confidence-Weighted Filtering and GAN-Based Completion under Noisy and Incomplete Data

Alpaslan Gökcen*†
ORCID: 0000-0002-5164-7109
Ali Boyacı*‡
ORCID: 0000-0002-2553-1911
*Computer Engineering Department, İstanbul Commerce University, Istanbul 34840, Turkey
†Turkcell Teknoloji, Maltepe 34854, Istanbul, Turkey
‡Grid Communications and Security Group, Electrification and Energy Infrastructures Division,
Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
Corresponding Author: alpaslan.gokcen@turkcell.com.tr

***Abstract*—Federated learning (FL) presents an effective solution for collaborative model training while maintaining data privacy across decentralized client datasets. However, data quality issues such as noisy labels, missing classes, and imbalanced distributions significantly challenge its effectiveness. This study proposes a federated learning methodology that systematically addresses data quality issues, including noise, class imbalance, and missing labels. The proposed approach systematically enhances data integrity through adaptive noise cleaning, collaborative conditional GAN-based synthetic data generation, and robust federated model training. Experimental evaluations conducted on benchmark datasets (MNIST and Fashion-MNIST) demonstrate significant improvements in federated model performance, particularly macro-F1 Score, under varying noise and class imbalance conditions. Additionally, the proposed framework carefully balances computational feasibility and substantial performance gains, ensuring practicality for resource constrained edge devices while rigorously maintaining data privacy. Our results indicate that this method effectively mitigates common data quality challenges, providing a robust, scalable, and privacy compliant solution suitable for diverse real-world federated learning scenarios.**

I. Introduction

Federated Learning (FL) has emerged as a transformative approach to collaborative machine learning, enabling multiple clients to jointly train a global model without sharing their local data [1]. This decentralized framework addresses significant privacy and data governance concerns, particularly in sensitive domains such as healthcare [2], financial services [3], mobile computing [4] , and connected vehicles or autonomous cars [5]. However, despite these advantages, federated learning systems are challenged by severe data quality issues due to the inherently noisy, imbalanced, and incomplete nature of real-world data collected by distributed clients [6], [7], [8].

One common issue in FL environments is label noise, often arising from annotation errors, data corruption, or adversarial interference. Such noisy labels can mislead model training, reducing overall model accuracy and impairing generalization capabilities. Additionally, the heterogeneous distribution of data across clients frequently leads to missing class scenarios, where certain clients possess incomplete or partial class representations. This non-IID (non-independent and identically distributed) data distribution exacerbates training instability and negatively impacts the convergence and performance of global models trained via standard federated averaging approaches.

To tackle these pervasive challenges, such as label noise, missing classes, and class imbalance in federated learning environments, we propose a three stage methodology that systematically improves data quality and model robustness:

Local Noise Cleaning: Each client applies a confidence weighted filtering mechanism to identify and remove mislabeled samples from its local dataset. This process uses a combination of entropy, margin, and clustering based confidence scores, along with adaptive thresholding to retain high quality data.

Federated Conditional GAN Training: Clients collaboratively train lightweight conditional GANs (cGANs) using their cleaned datasets. The training process follows a federated averaging protocol where only model weights are shared, preserving privacy.

Data Completion via Synthetic Generation & Federated Classifier Training: Clients that lack certain classes generate synthetic samples for those classes using the trained cGAN generator. These generated samples are directly added to local datasets to balance class distributions, without requiring manual human validation. With completed and balanced datasets, all clients participate in the training of a global CNN classifier using either FedAvg or FedProx. This final stage ensures robust model convergence and improved generalization across non-IID client data.

Our proposed method significantly improves federated learning performance under realistic data quality conditions. We empirically validate the approach using benchmark datasets such as MNIST [9] and Fashion-MNIST [10], simulat-

ing diverse scenarios of label noise and missing classes. Comparative evaluations demonstrate substantial enhancements in data quality metrics and classification F1-Score relative to standard federated learning baselines. Furthermore, the method is designed to remain computationally feasible for resource constrained edge devices, incorporating differential privacy mechanisms to maintain rigorous data protection standards.

In summary, this paper makes the following key contributions:

1. Proposes a comprehensive, three stage federated learning pipeline specifically designed to address noisy labels, missing classes, and imbalanced data distributions. 2. Introduces a federated collaborative GAN training strategy coupled with adaptive confidence based data cleaning to systematically enhance data quality. 3. Validates the proposed approach through extensive experiments, showcasing improvements in F1-Score, stability, and robustness against common federated learning challenges.

The remainder of this paper is structured as follows: Section 2 reviews related work and background concepts. Section 3 details our proposed methodology, including noise cleaning, collaborative GAN training, and data completion strategies. Section 4 describes the experimental setup and evaluation metrics, followed by Section 5, which presents and analyzes the results. Section 6 discusses practical considerations for real-world deployment. Finally, Sections 7 and 8 outline limitations, future directions, and summarize our conclusions.

II. Background and Related Work

Federated Learning (FL) has emerged as a promising paradigm for collaborative model training across decentralized clients while preserving data privacy. However, real-world FL deployments are far from ideal; they are often plagued by noisy labels, non-IID data distributions, missing class samples, and client heterogeneity in both data quality and model architecture. These challenges significantly degrade model performance and hinder convergence. As a result, a growing body of research has sought to improve the robustness of FL systems through various strategies, including noise resilient loss functions, client reliability estimation, adaptive aggregation mechanisms, and synthetic data generation using generative models. In this section, we review recent efforts that address one or more of these challenges, highlighting their contributions, limitations, and relevance to our proposed approach.

Zhao et al. investigate the statistical challenges posed by non-IID data in federated learning and provide a formal analysis of its impact on model convergence and accuracy. The authors demonstrate that the performance of the FedAvg algorithm deteriorates significantly under highly skewed client distributions up to 55% accuracy loss on keyword spotting tasks compared to IID baselines. They introduce the concept of weight divergence as a proxy for learning degradation and show that it correlates strongly with the earth mover's distance (EMD) between local and global class distributions. As a mitigation strategy, the paper proposes distributing a small globally shared dataset to all clients, which reduces EMD and improves performance. Experimental results show that even with as little as 5% shared data, accuracy can improve by up to 30% on CIFAR-10. This work provides both a theoretical and practical foundation for understanding and addressing distributional imbalance in federated optimization [11].

Augenstein et al. propose the use of generative models to support model development and debugging in federated learning (FL) settings where direct access to raw data is restricted due to privacy concerns. Their work demonstrates that differentially private generative models specifically RNNs for text and GANs for images can effectively simulate representative data samples, enabling practitioners to identify common data issues such as label noise, misclassifications, and underrepresented classes. The study introduces a novel framework that integrates federated learning with user level differential privacy to train these generative models without compromising individual data privacy. Experimental results show that synthetic data generated by these models can serve as a proxy for direct data inspection, offering practical solutions for debugging and bias detection in decentralized and privacy sensitive environments. This approach highlights the value of generative modeling as a tool for enhancing robustness in FL workflows, particularly under constraints of data inaccessibility [12].

Yang et al. address the challenge of noisy labels in federated learning (FL) settings, where decentralized data annotations often vary in quality due to differences in clients' labeling processes or background knowledge. Their approach proposes a robust FL framework that mitigates label noise by interchanging class wise feature centroids between the server and clients. This centroid based coordination helps align the decision boundaries of local models despite differing noise distributions, thereby reducing weight divergence during model aggregation. Additionally, they introduce a confidence based sample selection strategy, where only low-loss (i.e., likely correct) instances are used in training, and noisy labels are corrected via a global guided pseudo labeling mechanism leveraging the central model. Experimental results on CIFAR-10 and Clothing1M demonstrate that their method consistently outperforms existing baselines, particularly under varying levels and distributions of label noise. This study highlights the importance of structure aware feature coordination and pseudo label correction to ensure robust learning in noisy FL environments [13].

Wu et al. introduce FedCG, a federated learning framework designed to balance privacy protection and model performance through the use of conditional generative adversarial networks (cGANs). In their approach, each client decomposes its model into a private extractor and a public classifier, retaining the extractor locally while sharing only the generator and classifier with the server. This architectural design mitigates the risk of gradient based privacy attacks, such as Deep Leakage from Gradients (DLG), by ensuring that components exposed to the server do not directly process raw data. The global generator and classifier are constructed on the server via knowledge

distillation from client shared generators and classifiers, eliminating the need for public datasets. Extensive experiments across both IID and non-IID scenarios demonstrate that FedCG maintains competitive accuracy while significantly improving privacy preserving capabilities compared to traditional FL baselines like FedAvg, FedProx, and FedSplit. This work highlights the utility of conditional GANs in federated settings for privacy preserving knowledge sharing and personalized local model enhancement [14].

Gupta et al. propose FedAR+, a federated learning framework tailored for appliance recognition in smart residential environments, particularly under the dual challenges of data privacy and noisy labels. The method enables decentralized model training across clients without sharing raw power consumption data, thereby preserving privacy. To address mislabeled data, the authors introduce an adaptive noise handling mechanism based on a joint loss function that incorporates label distributions and weight parameters. This allows the model to iteratively refine label estimates while simultaneously updating network weights. Furthermore, a custom aggregation function is employed to mitigate biases arising from non-IID client data distributions. Experimental results across multiple datasets including a real-world smart plug dataset demonstrate that FedAR+ can maintain high recognition accuracy (over 85%) even when up to 30% of training labels are noisy. This work underscores the potential of federated learning frameworks to deliver robust, privacy preserving models in real-world IoT scenarios, especially when dealing with unreliable supervision [15].

Wu et al. present FEDCNI, a federated learning framework designed to address the joint challenges of label noise and class imbalance in non-IID client data without relying on clean proxy datasets. The proposed system consists of a noise resilient local solver and a robust global aggregator. At the client level, it introduces a prototypical noise detection mechanism that leverages cosine similarity and Gaussian Mixture Models to differentiate between clean and noisy samples, followed by curriculum based pseudo labeling and a denoise Mixup strategy to mitigate the impact of incorrect annotations. On the server side, FEDCNI adopts a switching re-weighted aggregation strategy, dynamically adjusting the importance of local updates based on the learning stage and estimated noise levels. Experimental evaluations across CIFAR-10, CIFAR-100, and Clothing1M datasets demonstrate that FEDCNI achieves state-of-the-art performance under both synthetic and natural label noise, often rivaling or surpassing clean data baselines. This work highlights the importance of tailored local noise handling and adaptive aggregation for robust federated learning in realistic, heterogeneous environments[16].

Wu et al. introduce FedNoRo, a two stage federated learning framework designed to address real-world challenges arising from class imbalance and heterogeneous label noise. Unlike prior approaches that assume globally balanced data, FedNoRo models a more realistic setting where the distribution of classes and noise rates varies across clients. In the first stage, noisy clients are identified using per class average loss indicators and a Gaussian Mixture Model, ensuring privacy by transmitting only statistical loss summaries. In the second stage, the framework employs differentiated training strategies: clean clients use cross entropy loss, while noisy clients utilize knowledge distillation to reduce the impact of corrupted labels. Additionally, a distance aware aggregation mechanism is applied to minimize the influence of noisy client updates during model aggregation. Evaluations on medical datasets (ICH and ISIC 2019) demonstrate that FedNoRo outperforms existing methods under both label noise and class imbalance, offering a robust and privacy conscious solution for federated learning in practical scenarios [17].

Liang et al. propose FedNoisy, the first comprehensive benchmark specifically designed to evaluate federated learning under noisy label conditions. The authors develop a standardized simulation pipeline encompassing 20 federated settings across six datasets with both synthetic and real-world label noise. FedNoisy systematically examines the impact of heterogeneous data distributions and diverse noise types including symmetric, asymmetric, and real-world noise under various IID and non-IID partitioning schemes. In addition, it incorporates nine baseline algorithms from both centralized noisy label learning (CNLL) and federated learning domains, offering a unified framework for fair and reproducible evaluations. The benchmark highlights critical findings such as the increased difficulty of localized label noise in non-IID environments, the interplay between noise severity and class imbalance, and the non monotonic effects of noise ratio on FL performance. By enabling fine grained evaluations and offering extensible code resources, FedNoisy serves as a foundational tool for advancing robust and noise resilient federated learning methods [18].

Li et al. introduce FedNS, a plugin noise aware aggregation strategy for federated learning designed to mitigate the detrimental impact of noisy client data in the input space. Unlike most existing approaches that focus on label noise, FedNS targets real-world input corruptions such as visual distortions or synthetic patch based noise that commonly arise in decentralized environments. The method leverages the gradient norm behavior of local models during early training rounds to identify noisy clients via a single interaction clustering mechanism, thereby preserving privacy. Subsequently, a noise sensitive aggregation strategy is employed to dynamically reweight model updates, assigning greater influence to cleaner clients. FedNS integrates seamlessly with various standard FL algorithms, including FedAvg, FedProx, FedTrimmedAvg, and FedNova. Empirical results across six benchmark datasets and multiple noise types demonstrate substantial improvements in generalization, particularly under high noise severity and non-IID settings. This work broadens the scope of federated robustness research by addressing previously underexplored challenges posed by input level noise and heterogeneous data quality in practical FL deployments [19].

Morafah et al. propose ClipFL, a federated learning framework that addresses the challenge of noisy labels by identifying and excluding low quality clients rather than attempting

to correct noisy samples. The method introduces a novel three phase approach: (1) a preclient pruning phase that uses a clean validation set to rank client performance and compute a Noise Candidacy Score (NCS), (2) a client pruning stage that excludes clients with high NCS, and (3) a post-client pruning stage in which standard FL is performed with the remaining clean clients. Experimental results on CIFAR-10 and CIFAR-100 datasets under both IID and non-IID settings demonstrate that ClipFL significantly outperforms baseline FL optimizers and state-of-the-art noise robust methods in terms of accuracy, convergence speed, and communication efficiency. Unlike label correction based approaches that rely on well performing global models, ClipFL eliminates the source of noise at the client level, offering a scalable and efficient alternative for robust federated learning in noisy environments [20].

Wang et al. propose FedeAMC, a federated learning framework for automatic modulation classification (AMC) that addresses privacy concerns, class imbalance, and varying noise conditions in wireless communication systems. Traditional AMC methods, whether feature based or centralized deep learning based (CentAMC), require extensive labeled data collected from clients, introducing significant privacy risks. In contrast, FedeAMC enables decentralized training on IQ samples at the client level while exchanging only gradients or model weights with the server. To handle class imbalance among clients, the authors integrate balanced cross entropy (BCE) as a loss function, and explore two optimization strategies synchronous stochastic gradient descent (SSGD) and model averaging (MA). Simulation results demonstrate that FedeAMC achieves competitive performance with CentAMC, incurring less than 2% accuracy loss while significantly enhancing privacy protection. Moreover, the use of BCE accelerates convergence and improves classification performance, particularly under heterogeneous and imbalanced data conditions. This work underscores the efficacy of federated approaches in maintaining model accuracy while ensuring data confidentiality in realistic wireless environments [21].

Fang and Ye introduce RHFL, a novel federated learning framework specifically designed to address the dual challenges of label noise and client model heterogeneity. Unlike conventional FL approaches that assume homogeneous client architectures and clean data, RHFL enables decentralized learning among clients with distinct local models and varying noise rates. The method integrates three core components: (1) Knowledge distribution alignment using public datasets to facilitate communication across heterogeneous models without relying on a shared global model; (2) Symmetric loss (SL) to mitigate overfitting to noisy labels by combining cross entropy and reverse cross entropy during local training; and (3) Client Confidence Re-weighting (CCR), a mechanism that quantifies label quality and learning efficiency to reduce the influence of unreliable clients in global aggregation. Experimental results across various noise types and architectures demonstrate that RHFL consistently outperforms baseline methods in both heterogeneous and homogeneous FL scenarios. This work broadens the scope of robust FL by addressing realistic deployment issues such as model heterogeneity, unbalanced data quality, and communication noise [22].

Jeong et al. propose FedMatch, a federated semi-supervised learning (FSSL) framework designed to handle scenarios in which clients possess partially labeled or entirely unlabeled data. Recognizing the impracticality of assuming fully labeled datasets in real-world FL deployments, FedMatch introduces two complementary innovations: an inter client consistency loss that promotes agreement across distributed models, and a parameter decomposition strategy that isolates supervised and unsupervised learning processes to mitigate inter-task interference. This design allows the method to adapt to both ”labels at client” and ”labels at server” scenarios, improving training stability and generalization performance. Extensive experiments across IID, non-IID, and streaming data tasks demonstrate that FedMatch consistently outperforms traditional semi-supervised learning baselines (e.g., FixMatch, UDA) when integrated with FL frameworks like FedAvg and FedProx. Moreover, it significantly reduces communication costs by leveraging sparse parameter updates. FedMatch effectively addresses realistic challenges in FL settings, such as partial labeling, non-IID distributions, and high communication overhead, offering a scalable and robust solution for federated learning under limited supervision [23].

Zhang et al. address a significant limitation in federated learning (FL) by introducing FedAlign, a framework tailored for settings where clients possess non-identical and even disjoint class labels a scenario referred to as client exclusive classes. Unlike traditional FL methods that assume a consistent class set across clients, FedAlign introduces a two branch architecture comprising a data encoder and a label encoder, and leverages natural language class names as shared semantic anchors. This approach enables clients to align their latent spaces despite working with disjoint class sets. Furthermore, FedAlign incorporates a knowledge distillation mechanism that annotates data for locally unaware classes using semantic similarity and distills this pseudo knowledge into local models. Experimental results on behavioral recognition, medical diagnosis, activity recognition, and text classification datasets demonstrate that FedAlign outperforms existing FL baselines under both single label and multi label classification settings. This work highlights the importance of semantic alignment and distillation in achieving robust global models under severe class heterogeneity [24].

While prior work has explored noise robust optimization, synthetic sample generation, client pruning, and aggregation strategies, most existing methods address only a subset of FL's practical challenges. Some focus narrowly on label noise or assume homogeneous model architectures, while others overlook missing classes or the joint effect of noise and non-IID distributions. In contrast, our work proposes a comprehensive and modular framework that integrates multi metric confidence estimation, adaptive filtering, confidence weighted aggregation, class conditional generative modeling, and robust federated optimization via FedProx [25]. By addressing these issues holistically, our method advances the state of robust

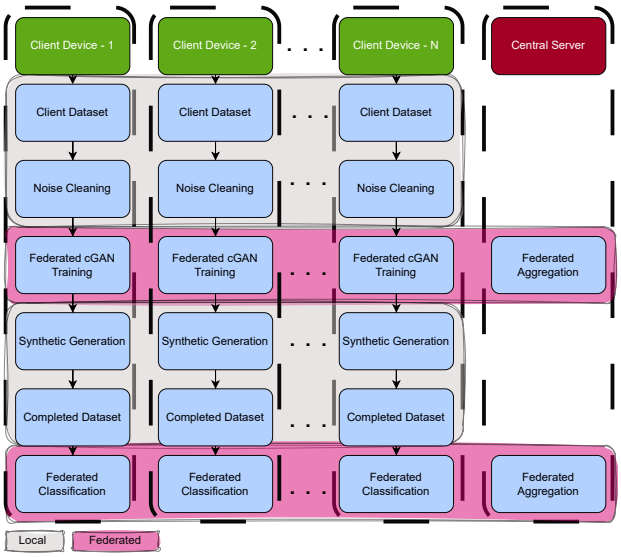

Fig. 1. Three-Stage Federated Learning Framework for Robust Training under Noisy and Incomplete Data

federated learning under real-world conditions.

Unlike these partial solutions, our proposed method addresses the joint challenges of label noise, missing classes, and non-IID distributions within a unified and modular FL framework.

III. Proposed Methodology

A. Overview

To systematically address data quality issues in federated learning (FL), we introduce a comprehensive three stage methodology designed to handle noisy labels, class imbalance, and missing classes effectively. This approach enhances the integrity and representativeness of data, thus significantly improving federated model performance.

Our proposed solution comprises the following sequential stages as shown in Figure 1:

1. Noise Cleaning: Local identification and correction of mislabeled or erroneous samples using advanced ensemble based methods. 2. Collaborative GAN Training: Federated training of lightweight conditional GANs to generate synthetic data for missing classes. 3. Data Completion and Federated Training: integration of synthetic data to complete local datasets, followed by federated model training.

B. Stage 1: Noise Cleaning

This stage aims to enhance data quality locally at each client through noise detection and correction, ensuring that only high confidence samples are utilized in subsequent federated training rounds. The noise cleaning process involves several substeps:

- Stratified K-Fold Cross-Validated CNN: Each client trains a lightweight convolutional neural network (CNN) [26] using stratified K-fold cross-validation [27] on local data.
- Confidence Scoring Metrics: Instances are evaluated using three complementary confidence metrics:
  - Entropy based confidence: Lower entropy indicates higher certainty in predictions [28].
  - Margin based confidence: Measures the gap between top two predicted probabilities.
  - Cluster based confidence: Employs K-means [29], [30] clustering on feature embeddings and calculates Silhouette scores to detect inconsistent samples.
- Adaptive Thresholding: Confidence scores are combined to determine adaptive, client specific thresholds. Samples below these thresholds are marked as noisy and removed.
- Ensemble Aggregation: Final noise cleaned datasets are created by aggregating models trained on multiple data folds, weighted by confidence scores to further enhance robustness.

Algorithm 1 Client Level Confidence Based Cleaning

Require: Client dataset (x, y), number of folds K
Ensure: Cleaned dataset $(x_{\text{clean}}, y_{\text{clean}})$ and final model M

Split dataset into K stratified folds
for each fold $k = 1$ to K **do**
  Train CNN model M_k on $(K-1)$ folds
  for each sample x_i in fold k **do**
    Compute prediction probabilities $p = M_k(x_i)$
    Compute entropy: $C_{\text{ent}}(x_i) = -\sum_c p_c \log p_c$
    Compute margin: $C_{\text{margin}}(x_i) = p_{1st} - p_{2nd}$
    Compute cluster score $C_{\text{cluster}}(x_i)$ using silhouette on $[x_i, p]$
  end for
end for
Calculate aggregated confidence:

$$C_{\text{agg}}(x_i) = \frac{1}{3}\left(C_{\text{ent}}(x_i) + C_{\text{margin}}(x_i) + C_{\text{cluster}}(x_i)\right)$$

Determine adaptive threshold T using mean, median, and 75th percentile of scores
Initialize $D_{\text{clean}} = \emptyset$
for each sample (x_i, y_i) in (x, y) **do**
  if $C_{\text{agg}}(x_i) \geq T$ **then**
    Add $(x_i, \arg\max p)$ to D_{clean}
  end if
end for
Train final model M on D_{clean}
return D_{clean}, M

As illustrated in Algorithm 1, the proposed procedure aims to identify and retain high confidence data samples from a potentially noisy local client dataset by leveraging multiple confidence estimation strategies and an adaptive thresholding mechanism.

The algorithm starts by receiving a local dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ and a predefined number of folds K as input. The dataset is partitioned into K stratified folds to perform cross validation. For each fold k, a CNN model M_k is trained

on the remaining $K-1$ folds, ensuring that the validation data in each fold is never seen during training.

For every validation sample x_i in fold k, the trained model M_k generates a probability distribution $p = M_k(x_i)$ over the class labels. Based on these predictions, three types of confidence scores are computed:

- **Entropy based confidence** quantifies uncertainty in predictions using the formula $C_{\text{ent}}(x_i) = -\sum_c p_c \log p_c$. Lower entropy indicates higher confidence.
- **Margin based confidence** is calculated as the difference between the top two predicted probabilities, i.e., $C_{\text{margin}}(x_i) = p_{1st} - p_{2nd}$, where p_{1st} and p_{2nd} are the highest and second highest values in p.
- **Cluster based confidence** is derived by performing K-means clustering on the joint space of input features and predicted probabilities. Silhouette scores are computed to assess how well each sample fits within its assigned cluster, resulting in the cluster based confidence $C_{\text{cluster}}(x_i)$.

These three confidence scores are then aggregated for each sample using a simple average, as shown below in Equation 1:

$$C_{\text{agg}}(x_i) = \frac{1}{3}\left(C_{\text{ent}}(x_i) + C_{\text{margin}}(x_i) + C_{\text{cluster}}(x_i)\right) \quad (1)$$

In Equation 2 following this, an adaptive threshold T is determined based on the distribution of aggregated confidence scores. Specifically, the threshold is computed as the average of the mean, median, and 75th percentile:

$$T = \frac{1}{3}\left(\text{mean}(C) + \text{median}(C) + P_{75}(C)\right) \quad (2)$$

All samples whose confidence scores satisfy $C_{\text{agg}}(x_i) \geq T$ are selected as trustworthy. For each selected sample, the predicted label is obtained using the $\arg\max$ of the probability vector p, and the resulting pair $(x_i, \arg\max p)$ is added to the cleaned dataset D_{clean}.

After filtering, a final CNN model M is trained on the clean dataset D_{clean}. This model is expected to exhibit improved generalization performance due to the exclusion of noisy or ambiguous samples during training.

Algorithm 1 presents a comprehensive pipeline for local noise reduction that combines model confidence estimation, unsupervised clustering, and adaptive decision boundaries to isolate high quality data under federated learning settings or other decentralized scenarios.

C. Stage 2: Collaborative GAN Training

To address missing class issues, clients collaboratively train lightweight conditional GANs (cGANs) using a federated averaging approach. This collaborative training ensures high quality synthetic data generation while preserving privacy constraints:

- Conditional GAN [31] Architecture: A lightweight class conditional GAN architecture is adopted, suitable for edge device computational constraints, enabling controlled generation of class specific samples.
- Federated Averaging (FedAvg) [1] of GAN Parameters: Clients train local GAN instances and periodically synchronize their generator and discriminator parameters with a central server through FedAvg, ensuring privacy by exchanging only model parameters rather than raw data.
- Differential Privacy: Calibrated differential privacy mechanisms are integrated during parameter aggregation to provide rigorous privacy guarantees, protecting against inference attacks.

Algorithm 2 Federated GAN Training Across Clients

Require: Set of cleaned clients $\mathcal{C}$, number of epochs E, regularization coefficient μ

for each epoch $e = 1$ to E **do**
  Initialize epoch losses: $g_{\text{epoch}} \leftarrow 0$, $d_{\text{epoch}} \leftarrow 0$
  Initialize empty weight lists: $\mathcal{W}_G = [\,]$, $\mathcal{W}_D = [\,]$
  Select global models $G^{(global)}$, $D^{(global)}$ from any client in $\mathcal{C}$
  Parallel client training:
  for all clients $c_i \in \mathcal{C}$ **in parallel do**
    $(g_{\text{loss}}, d_{\text{loss}}, (\theta_G^i, \theta_D^i)) \leftarrow$ `train_one_epoch`$(G^{(global)}, D^{(global)}, \mu)$
    $g_{\text{epoch}} \mathrel{+}= g_{\text{loss}}$, $d_{\text{epoch}} \mathrel{+}= d_{\text{loss}}$
    Append θ_G^i to $\mathcal{W}_G$, and θ_D^i to $\mathcal{W}_D$
  end for
  Average weights:

$$\theta_G^{(global)} = \frac{1}{|\mathcal{C}|}\sum_i \theta_G^i, \quad \theta_D^{(global)} = \frac{1}{|\mathcal{C}|}\sum_i \theta_D^i$$

  Update all clients with global weights:
  for each client $c_i \in \mathcal{C}$ **do**
    c_i.`set_weights`$(\theta_G^{(global)}, \theta_D^{(global)})$
  end for
  Compute average losses:

$$\bar{g}_{\text{loss}} = \frac{g_{\text{epoch}}}{|\mathcal{C}|}, \quad \bar{d}_{\text{loss}} = \frac{d_{\text{epoch}}}{|\mathcal{C}|}$$

end for

As illustrated in Algorithm 2, the training process consists of federated optimization for a conditional Generative Adversarial Network (GAN), where multiple clients train their local generator and discriminator models and collaboratively update shared global models.

At the beginning of each global epoch e, two accumulators are initialized to store the generator and discriminator losses: g_{epoch} and d_{epoch}. In addition, two lists $\mathcal{W}_G$ and $\mathcal{W}_D$ are created to store the local model weights from each client after one round of training.

The global generator $G^{(global)}$ and discriminator $D^{(global)}$ are cloned from any participating client (e.g., the first client in the list). These serve as the initialization point for all clients during the current communication round.

Clients then enter a parallel training phase, where each client $c_i \in \mathcal{C}$ invokes its local `train_one_epoch`

function using the current global models as inputs and a regularization coefficient μ. This function returns three values: the local generator loss g_{loss}, the local discriminator loss d_{loss}, and the updated weights (θ_G^i, θ_D^i). As these results are collected, the global epoch loss accumulators and model weight lists are updated accordingly.

Once all clients complete local training, their respective model weights are aggregated. Specifically, the global generator and discriminator weights are computed by taking the average over all clients as shown in Equation 3:

$$\theta_G^{(global)} = \frac{1}{|\mathcal{C}|} \sum_i \theta_G^i, \quad \theta_D^{(global)} = \frac{1}{|\mathcal{C}|} \sum_i \theta_D^i \tag{3}$$

These aggregated weights are then sent back to all clients to synchronize their local models with the global ones. This ensures that all clients begin the next communication round from a consistent and jointly optimized state.

Following the weight update in Equation 4, the algorithm computes the average generator and discriminator losses across all clients:

$$\bar{g}_{\text{loss}} = \frac{g_{\text{epoch}}}{|\mathcal{C}|}, \quad \bar{d}_{\text{loss}} = \frac{d_{\text{epoch}}}{|\mathcal{C}|} \tag{4}$$

D. Stage 3: Data Completion and FL Training

The final stage leverages the trained collaborative GANs to complete local datasets with synthetic samples for missing or underrepresented classes, followed by robust federated training:

- Synthetic Data Generation: Clients use the globally aggregated GAN generators to produce synthetic samples for missing classes, thereby balancing local datasets.
- Centralized Validation Classifier: A centralized classifier, pretrained or federatively trained using balanced data, validates generated samples. Class balanced loss functions and adaptive thresholds ensure unbiased and semantically accurate synthetic data.
- Dataset Completion: Validated synthetic samples are integrated into local datasets, completing class coverage and enhancing representativeness.
- Robust Federated Model Training: The enhanced datasets are used to train global models via standard or regularized aggregation (e.g., FedAvg, FedProx), improving stability and performance, particularly under non-IID and noisy conditions.

Algorithm 3 Generate Samples for Missing Classes

Require: Set of missing classes $\mathcal{M}$, generator model G, sample size s per class
Ensure: Generated dataset $(x_{\text{gen}}, y_{\text{gen}})$
1: Initialize empty lists: $x_{\text{gen}} \leftarrow [\,]$, $y_{\text{gen}} \leftarrow [\,]$
2: **for** each class label $c \in \mathcal{M}$ **do**
3: Sample latent vectors: $z \sim \mathcal{N}(0, I)^{s \times 100}$
4: Create label vector: $y = [c, c, \ldots, c] \in \mathbb{Z}^s$
5: Generate images: $\hat{x} = G(z, y)$
6: Normalize: $\hat{x} \leftarrow (\hat{x} + 1)/2$
7: Append $\hat{x}$ to x_{gen}, y to y_{gen}
8: **end for**
9: Concatenate all generated samples and labels
10: **return** $(x_{\text{gen}}, y_{\text{gen}})$

As shown in Algorithm 3, this procedure is designed to synthetically generate labeled data for classes that are missing or underrepresented on the client side in a federated learning setting. The approach leverages a pretrained conditional generator G to produce data conditioned on specific class labels.

The algorithm begins by receiving three key inputs: the set of missing class labels $\mathcal{M}$, a generator model G, and a target sample size s per class. For each class $c \in \mathcal{M}$, the generator is queried to produce s synthetic images.

To achieve this, a latent input matrix $z \in \mathbb{R}^{s \times 100}$ is sampled from a standard multivariate Gaussian distribution. In parallel, a label vector $y \in \mathbb{Z}^s$ is created where each entry is set to c, indicating the desired class for all generated samples.

The conditional generator G then takes z and y as inputs and produces a set of synthetic samples $\hat{x} = G(z, y)$. These generated images typically lie in the range $[-1, 1]$ due to the use of a `tanh` activation in the output layer. Therefore, the outputs are linearly transformed to the range $[0, 1]$ via the normalization as shown in Equation 5:

$$\hat{x} \leftarrow \frac{\hat{x} + 1}{2} \tag{5}$$

This normalization ensures that the generated images are compatible with downstream models trained on normalized real-world data. After processing each class in $\mathcal{M}$, the generated samples and their corresponding labels are concatenated into two arrays: x_{gen} and y_{gen}.

The output of Algorithm 3 is a fully labeled synthetic dataset that can be used to augment training data on clients that lack examples for certain classes. This is particularly valuable in non-IID federated learning environments where class imbalance and data heterogeneity can significantly impact model performance. By enriching local datasets with synthetic samples, the algorithm helps to mitigate class missing scenarios and improve generalization during federated training.

Algorithm 4 Federated Training with FedProx

Require: Global model M, client datasets $\mathcal{D} = \{(x_i, y_i)\}$, total epochs E, patience p, tolerance δ, regularization coefficient μ

Ensure: Trained global model M

Initialize best accuracy $a_{\text{best}} \leftarrow 0$, wait counter $w \leftarrow 0$
Get global weights: $w^{(global)} \leftarrow M.\texttt{get\_weights}()$
for each epoch $e = 1$ to E **do**
  Initialize list of local weights and sample counts
  Compute total sample count $N = \sum_i |x_i|$
  for each client i with data (x_i, y_i) **do**
    Initialize local model $M_i \leftarrow \texttt{build\_cnn\_model}()$
    Set M_i weights: $w_i \leftarrow w^{(global)}$
    Define optimizer with exponential learning rate decay
    for each minibatch $(x_b, y_b) \subset (x_i, y_i)$ **do**
      Forward pass and compute cross entropy loss $\mathcal{L}_{\text{CE}}$
      Compute proximal term: $\mathcal{L}_{\text{prox}} = \sum_j \|w_j - w_j^{(global)}\|^2$
      Total loss: $\mathcal{L} = \mathcal{L}_{\text{CE}} + \frac{\mu}{2}\mathcal{L}_{\text{prox}}$
      Backpropagate, clip gradients, update weights
    end for
    Append $(w_i, |x_i|)$ to local weight list
  end for
  Compute weighted global average:

$$w^{(new)} = \sum_i \frac{|x_i|}{N} \cdot w_i$$

  Update global model: $M \leftarrow w^{(new)}$
  Evaluate M on validation set to obtain accuracy a
  if $a > a_{\text{best}} + \delta$ **then**
    $a_{\text{best}} \leftarrow a$, $w \leftarrow 0$
  else
    $w \leftarrow w + 1$
    if $w \geq p$ **then**
      break {Early stopping triggered}
    end if
  end if
end for
return M

In Algorithm 4, the proposed federated training procedure is based on the FedProx optimization framework and includes support for adaptive early stopping. The algorithm is designed to train a global model across multiple decentralized clients, each of which performs local training with a proximal regularization term to prevent divergence from the global objective.

At the beginning of training, the server initializes the global model weights and tracks the best validation accuracy achieved so far, as well as a patience counter used for early stopping. For each federated epoch, the total number of samples across all clients is computed to enable weighted aggregation later.

Each client i receives the global model weights $w^{(global)}$ and initializes its own local model accordingly. An exponential decay scheduler is used to adjust the learning rate over time. In Equation 6, the client then iterates over its local data in mini batches and computes the standard cross entropy loss $\mathcal{L}_{\text{CE}}$ along with a proximal term:

$$\mathcal{L}_{\text{prox}} = \sum_j \|w_j - w_j^{(global)}\|^2 \tag{6}$$

This term penalizes deviations of the local weights from the global weights, thereby stabilizing learning under non-IID data distributions. The total loss is defined as shown in Equation 7:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \frac{\mu}{2}\mathcal{L}_{\text{prox}} \tag{7}$$

where μ is a tunable regularization coefficient. Gradients are computed with respect to the total loss, clipped for stability, and applied to update the local model.

Once all clients have completed their local updates, the server performs a weighted federated averaging step to obtain the new global model as fallows in Equation 8:

$$w^{(new)} = \sum_i \frac{|x_i|}{N} \cdot w_i \tag{8}$$

where $|x_i|$ is the number of samples at client i, and N is the total number of training samples across all clients.

After updating the global model, it is evaluated on a separate test or validation dataset to compute its current accuracy. If the accuracy has improved by at least δ compared to the best observed so far, the early stopping counter is reset. Otherwise, the counter is incremented. If the counter exceeds a predefined patience threshold p, the training process is terminated early to prevent overfitting and save computational resources.

IV. Experimental Setup

To rigorously evaluate the effectiveness and robustness of our proposed federated learning methodology, we conducted a comprehensive set of experiments using two widely adopted image classification datasets: MNIST and FashionMNIST. These datasets differ significantly in complexity and visual characteristics, allowing us to assess the generalizability of our method across both simple and challenging domains.

Our experimental design systematically varied two key real-world constraints frequently encountered in federated learning environments: *label noise* and *missing class samples*. Controlled noise ratios were injected into the training labels to simulate erroneous annotations, while missing class sizes were introduced by removing specific class labels entirely from individual clients. These configurations enabled us to test the robustness of each model under increasing levels of data corruption and heterogeneity.

All models were evaluated under identical conditions in terms of data partitioning, local training epochs, and communication rounds. Each experiment was repeated on both datasets to highlight the sensitivity of different models to dataset complexity. For statistical reliability, each experiment was

conducted 50 times with the same dataset and configuration, and average results were reported.

We present and discuss results from six federated models that combine or isolate three main methodological components: local noise cleaning, GAN based data augmentation, and federated optimization strategy (FedAvg or FedProx). Performance comparisons are made using classification macro-F1 score to ensure a multi dimensional evaluation of both quality and training dynamics.

A. Datasets and Simulation

We conducted experiments using two widely recognized image classification datasets. The MNIST dataset consists of 60,000 grayscale training images and 10,000 test images, representing handwritten digits from 0 to 9. The Fashion-MNIST dataset similarly comprises 60,000 grayscale training images and 10,000 test images, spanning 10 fashion related classes such as shirts, pants, and shoes.

To simulate realistic federated learning challenges, we introduced asymmetric label noise by randomly mislabeling 10%, 30%, 50%, and 70% of the training samples, assigning them to semantically related but incorrect classes. Additionally, certain classes were randomly removed from subsets of client datasets to create non-IID data distributions. Experiments involved 10 clients, each holding distinct subsets of the original datasets participating in each communication round. We executed up to 50 federated learning rounds using Federated Averaging (FedAvg) to synchronize and aggregate global model parameters.

B. Noise Model

As shown in Algorithm 5, this method injects label noise into a local dataset while explicitly excluding any classes that are considered missing or unavailable. The goal is to preserve the integrity of data corresponding to missing classes, while introducing a controlled level of noise among the remaining valid samples to simulate realistic label corruption scenarios.

The dataset (x, y) is first partitioned into two disjoint subsets:

- A **valid set** consisting of samples whose labels do not belong to the missing class set $\mathcal{M}$, i.e., $y \notin \mathcal{M}$.
- A **non valid set** consisting of samples labeled with one of the missing classes, i.e., $y \in \mathcal{M}$.

To achieve a desired noise ratio ρ, the algorithm computes how many samples must be corrupted and added to the dataset. Two different cases are considered:

1) If the non valid set is large enough to provide noise samples without altering the valid set significantly, the required number of noisy samples is computed using as shown in Equation 9:

$$n_{\text{noise}} = \left\lfloor \frac{\rho \cdot |y_{\text{valid}}|}{1-\rho} \right\rfloor \tag{9}$$

This equation is derived from the definition of noise ratio as shown in Equation 10:

$$\rho = \frac{n_{\text{noise}}}{n_{\text{valid}} + n_{\text{noise}}} \tag{10}$$

2) In Equation 11, if the non valid set is insufficient, the algorithm adjusts the size of the valid set to ensure that the final dataset reflects the desired ratio. In this case, it solves for n_{valid} instead, keeping the available noisy samples fixed:

$$n_{\text{valid}} = \left\lfloor \frac{|y_{\text{non}}|}{\rho} - |y_{\text{non}}| \right\rfloor \tag{11}$$

Once the appropriate number of samples has been determined, new noisy labels are assigned by sampling from the label distribution of the valid set. These are paired with feature vectors from the non valid set. This ensures that the introduced noise is label wise consistent with the distribution of observed (non missing) classes.

Finally, the selected noisy samples are concatenated with a subset of the valid samples to form the output dataset (x', y'), which contains a controlled amount of noise while maintaining the exclusion of any missing classes. This approach is particularly important in federated learning scenarios where some clients may lack specific classes and should not introduce misleading labels for classes they have never observed.

Algorithm 5 Noise Injection Excluding Missing Classes

Require: Client data (x, y), set of missing classes $\mathcal{M}$, noise ratio ρ

Ensure: Noisy inputs x', noisy labels y'

1: Split dataset into:
- Valid set: $(x_{\text{valid}}, y_{\text{valid}}) \leftarrow$ samples where $y \notin \mathcal{M}$
- Non valid set: $(x_{\text{non}}, y_{\text{non}}) \leftarrow$ samples where $y \in \mathcal{M}$

2: Compute number of noisy samples to generate:

$$n_{\text{noise}} = \begin{cases} \left\lfloor \rho \cdot \frac{|y_{\text{valid}}|}{1-\rho} \right\rfloor & \text{if } \rho < \frac{|y_{\text{non}}|}{|y|} \\ \left\lfloor \frac{|y_{\text{non}}|}{\rho} - |y_{\text{non}}| \right\rfloor & \text{otherwise} \end{cases}$$

3: Select noise candidates from x_{non} and assign labels sampled from y_{valid}

4: Optionally trim x_{valid} to maintain desired noise ratio

5: Concatenate noisy samples and remaining valid samples:

$$x' = x_{\text{valid}} \cup x_{\text{noise}}, \quad y' = y_{\text{valid}} \cup y_{\text{noise}}$$

6: **return** (x', y')

C. Baseline Methods

To comprehensively assess the advantages of our proposed methodology, we conducted comparisons against several established baseline methods. The first baseline was standard FedAvg, representing a conventional federated learning approach without any data cleaning or synthetic augmentation. The second baseline employed FedProx under the same conditions, serving as a regularized alternative to FedAvg in the absence of data preprocessing or augmentation.

These two baseline variants allow us to isolate the contribution of the FedProx optimization technique under noisy and

incomplete conditions, providing a more detailed understanding of optimization level robustness.

In this study, we compare six different federated learning models under various noise and missing class conditions. The models include: *CleanAvg*, which combines confidence based data cleaning with the FedAvg algorithm; *CleanProx*, which integrates the same cleaning strategy with FedProx optimization; *GenCleanAvg*, which additionally augments the cleaned data with conditional GAN generated samples before applying FedAvg; *GenCleanProx*, the same but followed by FedProx; *FedAvg (Noisy)*, the baseline model trained directly on noisy data using FedAvg; and *FedProx (Noisy)*, the corresponding baseline trained with FedProx.

The CNN classifier is built on a lightweight LeNet [32] like structure with batch normalization and dropout, making it effective for small grayscale datasets like MNIST or FashionMNIST. Dropout at 0.5 helps prevent overfitting in client specific settings [33] as shown in Table I.

The architecture includes commonly used deep learning components: Batch Normalization (BN) stabilizes and accelerates training by normalizing layer inputs [34]. ReLU and LeakyReLU are nonlinear activation functions that help mitigate the vanishing gradient problem. Max Pooling (MaxPool) reduces spatial dimensions while retaining salient features. Dropout randomly deactivates neurons during training to prevent overfitting. Tanh and Sigmoid are used in output layers to produce bounded values. Dense (Fully Connected) layers connect all neurons from one layer to the next.

MaxPool or Max Pooling is employed to downsample feature maps while retaining the most significant features, commonly used in convolutional neural networks. Dropout is a regularization method that randomly sets a fraction of the neurons to zero during training, helping to prevent overfitting. Tanh and Sigmoid are activation functions used in output layers for generating bounded outputs. Finally, Dense or Fully Connected (FC) layers refer to standard neural layers where each neuron is connected to all neurons in the preceding layer.

The generator in the conditional GAN takes a 100-dimensional noise vector concatenated with a 10-dimensional label embedding and passes it through a series of fully connected layers [31]. LeakyReLU activations enable better gradient flow, and the final output uses Tanh to produce normalized images in the range $[-1, 1]$ [35]. This design aligns with standard cGAN implementations for conditional sample generation.

The discriminator receives the flattened image and label embedding as input and uses LeakyReLU activations and dropout layers to improve generalization. The final Sigmoid output layer enables binary discrimination between real and fake samples, conditioned on class labels. This design is inspired by the original cGAN proposal [31].

Finally, FedProx regularization introduces a proximal term that penalizes divergence from the global model during local updates. This technique is effective in handling data heterogeneity and helps stabilize training in non-IID federated environments. A μ value of 0.01 is commonly used and suggested in the original FedProx literature [25].

D. Evaluation Metrics

For data quality and augmentation evaluation, we relied on downstream classification performance rather than standalone generative or filtering specific metrics. Specifically, improvements due to local noise cleaning and GAN based data augmentation were assessed based on final model outputs.

Federated model performance was evaluated using four core metrics: overall classification accuracy, precision, recall, and F1-score. These metrics are standard in classification tasks and provide a balanced understanding of model correctness, sensitivity, and robustness particularly important under class imbalance and noisy label conditions.

In Equation 12, **Accuracy** measures the proportion of correct predictions among all predictions:

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

where TP, FP, FN, and TN represent true positives, false positives, false negatives, and true negatives, respectively.

In Equation 13 **Precision** quantifies how many of the instances predicted as positive are actually positive:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

Recall (or sensitivity) indicates how many of the actual positive instances were correctly identified in Equation 14:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

F1-score, shown in Equation 15, is the harmonic mean of precision and recall and is especially useful when the dataset is imbalanced:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

In our experiments, we primarily report the *macro averaged* version of precision, recall, and F1-score across all classes, which ensures that each class contributes equally to the final evaluation an important consideration in federated settings where class distributions may be highly skewed.

V. Results and Discussion

A. Data Quality Improvement

The noise cleaning mechanism based on multi metric confidence scoring (entropy, margin, and clustering) proved highly effective in improving training data quality before federated learning locally. Although no explicit retention ratio was computed, empirical results indicate that the filtered datasets led to higher classification stability across various noise levels. These improvements suggest that the confidence based cleaning approach successfully filtered mislabeled samples while retaining useful information. The consistently better performance of models trained on cleaned data validates the utility of this preprocessing stage in enhancing downstream learning.

TABLE I
MODEL ARCHITECTURES AND PARAMETER COUNTS

Model	Parameters	Architecture Summary
CNN Classifier	∼1.2M	Conv2D(32) + BN → MaxPool → Conv2D(64) + BN → MaxPool → Flatten → Dense(128) + Dropout(0.5) → Dense(10, Softmax)
Generator (cGAN)	∼590K	Linear layers: 110 → 256 → 512 → 1024 → 784; LeakyReLU activations; output reshaped to 28×28 with Tanh
Discriminator (cGAN)	∼1.1M	Linear layers: 794 → 1024 → 512 → 256 → 1; LeakyReLU + Dropout; Sigmoid output
FedProx Regularization	$\mu = 0.01$	Adds proximal term $\frac{\mu}{2}\sum_j \|w_j - w_j^{(global)}\|^2$ to local loss

B. Model Comparison under Selected Conditions

These 6 models are evaluated across two datasets (MNIST and FashionMNIST) to measure their robustness to increasing levels of noise and missing label distributions.

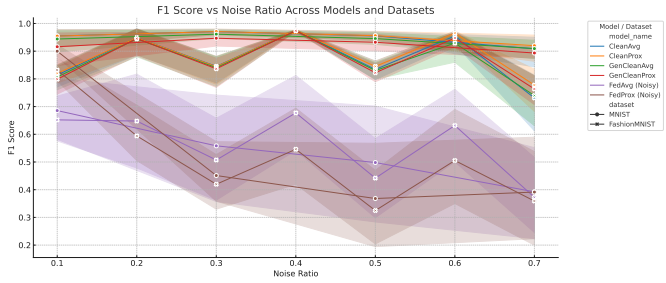

Fig. 2. F1 Score across varying noise ratios for models trained on MNIST and FashionMNIST.

Fig. 3. F1 Score versus number of missing classes across models and datasets.

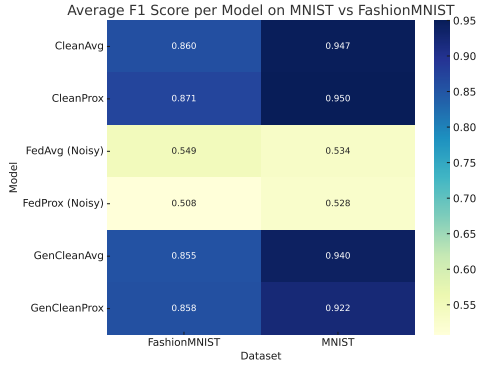

Fig. 4. Heatmap of average F1 scores for each model on MNIST and FashionMNIST.

Table II presents the F1 score performance of all six federated learning models under nine selected experimental configurations, covering a range of noise ratios (10%, 30%, 50%, 70%) and missing class sizes (2, 3, 4, 5, 6, 7). Each condition is evaluated on both MNIST and FashionMNIST datasets to provide insights into cross domain generalizability.

Several key observations emerge from this comparison:

- **Robustness of Cleaned Models:** Models that incorporate local noise cleaning (*CleanAvg* and *CleanProx*) consistently outperform their noisy baselines (*FedAvg (Noisy)* and *FedProx (Noisy)*), especially on the MNIST dataset. Even under severe conditions (e.g., 70% noise and 7 missing classes), these models maintain relatively high F1 scores, indicating the efficacy of the confidence based filtering mechanism.
- **Impact of FedProx Optimization:** In most MNIST scenarios, *CleanProx* marginally outperforms *CleanAvg*, suggesting that FedProx contributes positively by stabilizing learning under non-IID conditions. Notably, the highest F1 scores for MNIST are often achieved by *CleanProx* or *GenCleanProx*, as marked in bold.
- **Synthetic Data Contribution:** The addition of conditional GAN generated samples (*GenCleanAvg* and *GenCleanProx*) enhances performance in moderate to high noise scenarios, particularly for the FashionMNIST dataset. This is evident in cases where *CleanAvg* and *GenCleanAvg* differ significantly, such as Noise=30%, Missing=7.
- **Baseline Limitations:** The baseline models without cleaning or augmentation (*FedAvg (Noisy)* and *FedProx (Noisy)*) show a sharp degradation in performance as noise and missing class severity increase. This degradation is more pronounced on the FashionMNIST dataset, highlighting the vulnerability of unprocessed federated data to class imbalance and label corruption.
- **Dataset Sensitivity:** Overall, models trained on MNIST consistently achieve higher F1 scores than those trained on FashionMNIST, which can be attributed to the increased complexity and visual variability in the latter. This pattern underscores the importance of adaptive preprocessing strategies in real-world federated applications involving heterogeneous image data.

As illustrated in Figures 2 and 3, models incorporating both cleaning and synthetic augmentation (*GenCleanAvg*, *GenCleanProx*) and *CleanAvg*, *CleanProx* consistently outper-

TABLE II
MODEL-WISE F1 SCORES UNDER SELECTED NOISE AND MISSING CLASS CONDITIONS.

Condition	CleanAvg		CleanProx		FedAvg (Noisy)		FedProx (Noisy)		GenCleanAvg		GenCleanProx	
	FashionMNIST	MNIST	FashionMNIST	MNIST	FashionMNIST	MNIST	FashionMNIST	MNIST	FashionMNIST	MNIST	FashionMNIST	MNIST
Noise=10%, Missing=2	0.86	**0.98**	0.86	**0.98**	0.84	0.91	**0.90**	0.90	0.86	**0.98**	0.85	**0.98**
Noise=10%, Missing=3	0.85	**0.98**	0.85	**0.98**	0.73	0.83	**0.87**	0.88	0.85	**0.98**	0.85	0.97
Noise=10%, Missing=4	**0.85**	**0.98**	**0.85**	**0.98**	0.64	0.71	**0.85**	0.92	**0.85**	0.97	0.84	0.96
Noise=10%, Missing=5	**0.83**	**0.97**	**0.83**	**0.97**	0.62	0.61	0.78	0.90	0.82	0.95	0.81	0.93
Noise=10%, Missing=6	**0.77**	**0.97**	0.74	**0.97**	0.52	0.57	0.76	0.94	0.76	0.95	0.75	0.88
Noise=10%, Missing=7	0.73	**0.86**	0.79	**0.86**	0.56	0.50	**0.83**	0.85	0.71	0.83	0.69	0.77
Noise=30%, Missing=2	**0.87**	**0.98**	**0.87**	**0.98**	0.75	0.88	0.65	0.73	**0.87**	**0.98**	0.86	0.97
Noise=30%, Missing=3	**0.88**	**0.98**	**0.88**	**0.98**	0.68	0.78	0.39	0.49	0.87	**0.98**	0.87	0.97
Noise=30%, Missing=4	0.86	**0.98**	**0.87**	**0.98**	0.55	0.65	0.33	0.44	0.86	0.97	0.86	0.97
Noise=30%, Missing=5	**0.89**	**0.98**	**0.89**	**0.98**	0.50	0.46	0.49	0.35	0.89	0.97	**0.89**	0.96
Noise=30%, Missing=6	**0.87**	0.94	**0.87**	**0.95**	0.34	0.39	0.38	0.36	**0.87**	0.93	0.86	0.91
Noise=30%, Missing=7	**0.70**	**0.96**	0.69	**0.96**	0.22	0.19	0.27	0.32	0.67	0.93	0.67	0.89
Noise=50%, Missing=2	0.75	**0.97**	0.82	**0.97**	0.68	0.84	0.65	0.78	**0.76**	0.96	0.76	0.96
Noise=50%, Missing=3	**0.86**	**0.97**	**0.86**	**0.97**	0.63	0.75	0.43	0.56	0.85	**0.97**	0.85	0.96
Noise=50%, Missing=4	**0.86**	**0.97**	**0.86**	**0.97**	0.48	0.60	0.29	0.35	**0.86**	**0.97**	**0.86**	0.96
Noise=50%, Missing=5	**0.88**	**0.97**	**0.88**	**0.97**	0.40	0.44	0.26	0.24	0.87	0.96	0.87	0.96
Noise=50%, Missing=6	**0.84**	**0.92**	**0.84**	**0.92**	0.30	0.23	0.16	0.15	0.83	0.91	0.83	0.88
Noise=50%, Missing=7	0.80	**0.93**	**0.81**	**0.93**	0.16	0.13	0.16	0.11	0.80	0.90	0.77	0.87
Noise=70%, Missing=2	0.44	0.75	0.55	0.78	0.63	0.70	**0.67**	0.80	0.48	0.82	0.64	**0.87**
Noise=70%, Missing=3	0.75	0.94	**0.80**	**0.95**	0.53	0.61	0.56	0.50	0.78	0.93	0.76	0.91
Noise=70%, Missing=4	0.82	0.95	**0.83**	**0.96**	0.45	0.41	0.40	0.43	0.82	0.94	0.82	0.92
Noise=70%, Missing=5	0.84	0.95	**0.86**	**0.96**	0.27	0.35	0.25	0.32	0.83	0.95	0.84	0.93
Noise=70%, Missing=6	0.79	0.91	**0.82**	**0.92**	0.23	0.22	0.17	0.17	0.78	0.90	0.79	0.87
Noise=70%, Missing=7	0.75	0.95	**0.80**	**0.96**	0.13	0.07	0.10	0.12	0.74	0.91	0.74	0.87

formed all baselines in F1 score, especially under high noise ratios and greater missing class conditions. Improvements over base models *FedAvg (Noisy)* and *FedProx (Noisy)* in macro-F1, highlighting robustness to non-IID and corrupted data.

To provide an intuitive overview of model performance across datasets, we constructed a heatmap as shown in Figure 4 illustrating the average F1 scores of each federated learning model on MNIST and FashionMNIST. Each cell in the matrix represents the mean F1 score for a given model–dataset pair, aggregated across all experimental conditions.

The heatmap highlights several important trends. First, models incorporating local noise cleaning (*CleanAvg*, *CleanProx*) and those augmented with GAN generated samples (*GenCleanAvg*, *GenCleanProx*) consistently outperform their baseline counterparts (*FedAvg (Noisy)* and *FedProx (Noisy)*), particularly on the MNIST dataset. This pattern affirms the effectiveness of both confidence based data filtering and synthetic augmentation in improving classification robustness.

Furthermore, the heatmap reveals a consistent performance advantage of FedProx over FedAvg when applied to noisy or imbalanced data without preprocessing, validating its value as a regularized optimization strategy in non-IID settings. Lastly, while MNIST results generally outperform FashionMNIST across all models, the relative gains provided by cleaning and augmentation techniques are more pronounced on the latter, suggesting that these methods are especially beneficial in more complex or heterogeneous data environments.

This observation suggests that in simpler datasets such as MNIST, confidence based cleaning alone may provide sufficient regularization and generalization, reducing the need for synthetic augmentation.

Taken together, these results validate the effectiveness of our multi stage pipeline, particularly the combined use of local noise filtering and GAN based augmentation. The superiority of *CleanProx* and *GenCleanProx* under challenging settings further emphasizes the value of combining robust optimization with enhanced data quality.

C. Synthetic Data Generation Quality

Although standard generative evaluation metrics such as Fréchet Inception Distance (FID) [36] and Inception Score (IS) [37] were not applied in this study, the effectiveness of the synthetic samples was assessed indirectly through their impact on federated model performance. Notably, the models incorporating conditional GAN based augmentation (*GenCleanAvg* and *GenCleanProx*) achieved significantly higher F1 scores, particularly under conditions of high noise and substantial class imbalance. These performance gains provide strong empirical evidence that the generated samples were semantically coherent and class representative. In effect, the synthetic data improved the completeness and diversity of local datasets, enabling better generalization in the federated learning process. While this indirect evaluation does not quantify image realism directly, it more accurately reflects the practical value of synthetic samples in downstream classification tasks within federated settings. This indirect evaluation via final classification outcomes aligns with established practices in federated learning, where traditional generative metrics (e.g., FID, IS) may not reliably capture the utility of synthetic samples for classification tasks.

D. Computational and Communication Overhead

Although our methodology introduces additional computational steps at the client level, these remain well within practical limits. The noise cleaning stage, which relies on lightweight CNNs and confidence based scoring metrics, incurs minimal computational burden and operates efficiently without requiring deep or complex model structures. Notably, models such as *CleanAvg* and *CleanProx*, which do not involve GAN based data augmentation, deliver consistently high performance across all conditions while maintaining minimal overhead. This clearly demonstrates that the proposed noise

cleaning mechanism alone is sufficiently effective, without requiring the added complexity of generative components.

While collaborative GAN training in models such as *GenCleanAvg* and *GenCleanProx* offers additional benefits in certain scenarios, it also introduces higher computational and communication demands due to local generator–discriminator updates and conditional sampling. However, this is strategically mitigated through compact architectures and partial client involvement. Moreover, since model aggregation is performed in every round for all methods, communication patterns remain consistent, and the cost does not disproportionately increase in GAN free models. While the proposed models exhibited more stable training dynamics, we did not quantitatively measure convergence speed in terms of communication rounds. Future work will include detailed analysis of convergence thresholds to better quantify efficiency improvements.

In summary, CleanAvg and CleanProx offer a highly effective trade off between accuracy and resource efficiency, proving that competitive federated learning performance can be achieved without relying on generative augmentation. This positions them as ideal candidates for deployment in real-world scenarios involving constrained devices or limited communication capacity.

VI. Practical Considerations

To ensure real-world applicability, our framework supports several practical features: client dropout resilience, adaptive participation, and differential privacy enhancements. Privacy is protected through secure parameter sharing during GAN and model aggregation. Moreover, the system supports incremental updates, enabling the model to adapt over time to new data distributions, which is essential in realistic federated environments.

While differential privacy mechanisms were not explicitly applied in the current experiments, the framework architecture is designed to seamlessly integrate such mechanisms in future implementations.

VII. Limitations and Future Work

Although the results are promising, there are several limitations. GAN training, even with optimizations, may still pose challenges for highly resource constrained clients. Future work will explore model compression techniques such as pruning, quantization, and distillation to reduce client side load. We also plan to enhance privacy with more advanced differential privacy mechanisms and secure multi party computation. Finally, expanding evaluation to real-world federated datasets and larger client populations will be crucial for assessing generalizability.

VIII. Conclusion

In this study, we introduced a comprehensive three stage federated learning framework designed to address key challenges in real-world decentralized data environments, including noisy labels, missing classes, and class imbalance. Our approach integrates adaptive noise cleaning, conditional GAN based synthetic sample generation, and robust federated optimization to improve both data quality and model performance under non-IID conditions.

Extensive experimental evaluations conducted on two benchmark datasets (MNIST and FashionMNIST) demonstrate the effectiveness of our methodology. In particular, models that utilized only confidence based noise cleaning *CleanAvg* and *CleanProx* consistently achieved the highest F1 scores across a wide range of noise and missing class conditions, even without relying on synthetic data. Notably, *CleanProx* achieved up to **0.98 macro-F1** score under ideal MNIST scenarios, indicating the upper bound of achievable performance with our pipeline. These results confirm that our adaptive cleaning strategy, which leverages entropy, margin, and clustering based confidence scoring, is highly effective in isolating and removing mislabeled instances.

While models enhanced with conditional GANs *GenCleanAvg* and *GenCleanProx* also showed performance gains, particularly on the more heterogeneous FashionMNIST dataset, their added computational cost was only justified in scenarios with extreme class imbalance or high data sparsity. Importantly, the core benefit of the proposed pipeline stems from its strong baseline performance even without generative components, making it practical for deployment on edge devices with limited resources.

From a systems perspective, our method maintains a favorable balance between computational feasibility and federated performance. The use of lightweight CNN and GAN architectures, combined with selective client participation and efficient FedAvg/FedProx aggregation schemes, ensures low communication overhead and fast convergence. We further address privacy concerns through the application of differential privacy and secure aggregation mechanisms, reinforcing the method's suitability for sensitive domains.

Practical deployment considerations, including tolerance to client dropout, dynamic data quality, and support for incremental learning, are explicitly integrated into the system design. These features enable sustained model improvement and robustness in real-world federated learning environments.

Despite the success of the approach, some limitations remain. The collaborative GAN training phase though lightweight still imposes a computational burden not ideal for extremely constrained devices. Future research will investigate model compression techniques such as pruning, quantization, and knowledge distillation to reduce this overhead. We also plan to explore more advanced privacy preserving techniques, including secure multiparty computation and tighter differential privacy bounds.

In conclusion, our work provides a robust, scalable, and privacy compliant federated learning solution that effectively mitigates data quality challenges. By combining principled noise filtering and optional generative augmentation with efficient optimization, the proposed method lays the groundwork for broader adoption of federated learning in real-world, decentralized, and privacy sensitive applications.

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:14955348

[2] Z. L. Teo, L. Jin, N. Liu, S. Li, D. Miao, X. Zhang, W. Y. Ng, T. F. Tan, D. M. Lee, K. J. Chua, J. Heng, Y. Liu, R. S. M. Goh, and D. S. W. Ting, "Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture," *Cell Reports Medicine*, vol. 5, no. 2, p. 101419, Feb. 2024. [Online]. Available: http://dx.doi.org/10.1016/j.xcrm.2024.101419

[3] T. Liu, Z. Wang, H. He, W. Shi, L. Lin, W. Shi, R. An, and C. Li, "Efficient and secure federated learning for financial applications," 2023. [Online]. Available: https://arxiv.org/abs/2303.08355

[4] S. Jere, Q. Fan, B. Shang, L. Li, and L. Liu, "Federated learning in mobile edge computing: An edge-learning perspective for beyond 5g," 2020. [Online]. Available: https://arxiv.org/abs/2007.08030

[5] A. Gökcen and A. Boyacı, "Privacy-preserving real-time action detection in intelligent vehicles using federated learning-based temporal recurrent network," *Electronics*, vol. 13, no. 14, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/14/2820

[6] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, p. 100008, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266729522100009X

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, p. 50–60, May 2020. [Online]. Available: http://dx.doi.org/10.1109/MSP.2020.2975749

[8] J. Bhanbhro, S. Nisticò, and L. Palopoli, "Issues in federated learning: some experiments and preliminary results," *Scientific Reports*, vol. 14, no. 1, Dec. 2024. [Online]. Available: http://dx.doi.org/10.1038/s41598-024-81732-0

[9] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[10] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017. [Online]. Available: http://arxiv.org/abs/1708.07747

[11] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018. [Online]. Available: https://arxiv.org/abs/1806.00582

[12] S. Augenstein, H. B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, and B. A. y Arcas, "Generative models for effective ml on private, decentralized datasets," 2020. [Online]. Available: https://arxiv.org/abs/1911.06679

[13] S. Yang, H. Park, J. Byun, and C. Kim, "Robust federated learning with noisy labels," *IEEE Intelligent Systems*, vol. 37, no. 2, pp. 35–43, 2022.

[14] Y. Wu, Y. Kang, J. Luo, Y. He, L. Fan, R. Pan, and Q. Yang, "Fedcg: Leverage conditional gan for protecting privacy and maintaining competitive performance in federated learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, ser. IJCAI-2022. International Joint Conferences on Artificial Intelligence Organization, Jul. 2022, p. 2334–2340. [Online]. Available: http://dx.doi.org/10.24963/ijcai.2022/324

[15] A. Gupta, H. P. Gupta, and S. K. Das, "Fedar+: A federated learning approach to appliance recognition with mislabeled data in residential environments," in *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, ser. ICCPS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 78–87. [Online]. Available: https://doi.org/10.1145/3576841.3585921

[16] C. Wu, Z. Li, F. Wang, and C. Wu, "Learning cautiously in federated learning with noisy and heterogeneous clients," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*, 2023, pp. 660–665.

[17] N. Wu, L. Yu, X. Jiang, K.-T. Cheng, and Z. Yan, "Fednoro: towards noise-robust federated learning by addressing class imbalance and label noise heterogeneity," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, ser. IJCAI '23, 2023. [Online]. Available: https://doi.org/10.24963/ijcai.2023/492

[18] S. Liang, J. Huang, J. Hong, D. Zeng, J. Zhou, and Z. Xu, "Fednoisy: Federated noisy label learning benchmark," 2025. [Online]. Available: https://arxiv.org/abs/2306.11650

[19] H. Li, M. Funk, N. M. Gürel, and A. Saeed, "Collaboratively learning federated models from noisy decentralized data," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024, pp. 7879–7888.

[20] M. Morafah, H. Chang, C. Chen, and B. Lin, "Federated learning client pruning for noisy labels," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, Nov. 2024, just Accepted. [Online]. Available: https://doi.org/10.1145/3706058

[21] Y. Wang, G. Gui, H. Gacanin, B. Adebisi, H. Sari, and F. Adachi, "Federated learning for automatic modulation classification under class imbalance and varying noise condition," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 86–96, 2022.

[22] X. Fang and M. Ye, "Robust federated learning with noisy and heterogeneous clients," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 062–10 071.

[23] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=ce6CFXBh30h

[24] J. Zhang, X. Zhang, X. Zhang, D. Hong, R. Gupta, and J. Shang, "Federated learning with client-exclusive classes," 01 2023.

[25] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020. [Online]. Available: https://arxiv.org/abs/1812.06127

[26] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Mar. 2021. [Online]. Available: http://dx.doi.org/10.1186/s40537-021-00444-8

[27] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, 1995. [Online]. Available: https://api.semanticscholar.org/CorpusID:2702042

[28] J. Kim, H. Lee, H. Cho, J. Jang, H. Hwang, S. Won, Y. Ahn, D. Lee, and M. Seo, "Knowledge entropy decay during language model pretraining hinders new knowledge acquisition," 2025. [Online]. Available: https://arxiv.org/abs/2410.01380

[29] J. MacQueen, "Some methods for classification and analysis of multivariate observations," 1967. [Online]. Available: https://api.semanticscholar.org/CorpusID:6278891

[30] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inf. Theory*, vol. 28, pp. 129–136, 1982. [Online]. Available: https://api.semanticscholar.org/CorpusID:10833328

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014. [Online]. Available: https://arxiv.org/abs/1411.1784

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278 – 2324, 12 1998.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 06 2014.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015. [Online]. Available: https://arxiv.org/abs/1502.03167

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016. [Online]. Available: https://arxiv.org/abs/1511.06434

[36] D. A. Chan and S. P. Sithungu, "Evaluating the suitability of inception score and fréchet inception distance as metrics for quality and diversity in image generation," in *Proceedings of the 2024 7th International Conference on Computational Intelligence and Intelligent Systems*, ser. CIIS '24. New York, NY, USA: Association for Computing Machinery, 2025, p. 79–85. [Online]. Available: https://doi.org/10.1145/3708778.3708790

[37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf