Understanding the Impact of Data Noise in Federated Learning: [Experiments & Analysis]

JINMING HU* and JIAHAO GU*, University of Toronto, Canada
KENTA PLOCH, University of Toronto, Canada
HAO WANG, Stevens Institute of Technology, USA
JINGXIAN WANG, National University of Singapore, Singapore
WENTAO WU, Microsoft, USA
QIZHEN ZHANG, University of Toronto, Canada

Federated learning (FL) has emerged as a popular paradigm for distributed machine learning over decentralized data. A typical FL training task involves a fleet of client devices with private data and a centralized server for aggregating the global model. Data generated by FL clients, e.g., smart phones, vehicles, and cameras, is prone to noise. While the impact of data noise on centralized learning (CL) is well understood, to our best knowledge there is a lack of a systematic study from this point of view for FL. In this paper, we fill this gap by presenting an empirical investigation to provide a deeper understanding regarding the impact of data noise on FL. Our study is enabled by `DataNoiseGenerator`, an open-source and extensible toolkit that we developed for the injection of controlled data noise across five diverse data modalities: image, video, audio, text, and tabular data. We then carry out extensive experiments based on the noisy data generated by `DataNoiseGenerator`, and our experimental evaluation results reveal that FL is *significantly more vulnerable* to data noise compared to CL, in terms of the quality of the trained ML models. This gap between FL and CL widens as the intensity of data noise and the proportion of noisy FL clients increase. We further present a detailed analysis to diagnose the root cause of this increased sensitivity of FL to data noise. Our analysis finds that the aggregation performed by the FL server can *amplify* divergent updates from FL clients trained on noisy data, thereby hindering global model convergence. We conclude that data quality issues are a fundamental challenge for deploying robust FL systems and demand novel *decentralized* data cleaning mechanisms.

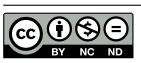
CCS Concepts: • **Information systems → Data cleaning**; • **Computer systems organization → Client-server architectures**.

Additional Key Words and Phrases: Federated Learning, Data Noise

ACM Reference Format:
Jinming Hu, Jiahao Gu, Kenta Ploch, Hao Wang, Jingxian Wang, Wentao Wu, and Qizhen Zhang. 2026. Understanding the Impact of Data Noise in Federated Learning: [Experiments & Analysis]. *Proc. ACM Manag. Data* 4, 3 (SIGMOD), Article 2467 (June 2026), 24 pages. https://doi.org/10.1145/3802124

*Co-first authors randomly ordered. Both contributed equally to this research.

Authors' Contact Information: Jinming Hu, jinminghu@cs.toronto.edu; Jiahao Gu, jiahao.gu@mail.utoronto.ca, University of Toronto, Toronto, Ontario, Canada; Kenta Ploch, University of Toronto, Toronto, Ontario, Canada, kenta.ploch@mail.utoronto.ca; Hao Wang, Stevens Institute of Technology, New York, USA, hwang9@stevens.edu; Jingxian Wang, National University of Singapore, Singapore, wang@nus.edu.sg; Wentao Wu, Microsoft, Redmond, USA, Wentao.Wu@microsoft.com; Qizhen Zhang, University of Toronto, Toronto, Ontario, Canada, qz@cs.toronto.edu.



ACM 2836-6573/2026/6-ART2467
https://doi.org/10.1145/3802124

1 Introduction

Federated learning (FL) [40, 72] has emerged as a popular, if not standard, paradigm for distributed machine learning (ML) over decentralized data, powering applications from medical image analysis to mobile keyboard prediction. A typical FL system includes a central server that coordinates with a multitude of clients. FL clients can run on edge devices, such as smart phones, laptops, or Internet of Things (IoT) sensors, that are located in different geographical regions, with private data that are typically not allowed to leave their compliance zones for data privacy or regulation reasons. As a result, FL clients can train models using *only* local private data and transmit local model updates (e.g., gradients or model weights) to the server. The server then *aggregates* these local updates to produce an improved *global* model, which is sent back to the clients for the next round of local model training and updates.

It is well known that the quality of the trained ML model is sensitive to the noise presented in the training data. Intensive and extensive studies have been conducted on this aspect, coined "data-centric ML/AI" in the literature, and numerous techniques have been proposed to improve data quality for ML training [28, 54]. However, the focus of this line of research has been on centralized learning (CL), and the impact of noisy data on FL remains elusive. Existing FL systems operate under the idealized assumption that client data is clean and well-formed. In contrast, real-world data are often collected "in the wild" and are inherently imperfect. For example, the data generation and collection processes on edge devices can suffer from a wide range of noise sources, stemming from hardware limitations (e.g., sensor noise in images or videos, microphone distortion in audio), environmental factors (e.g., poor illumination for videos, ambient sound for audios), and human errors (e.g., typos in input text from smartphone users). This disconnect between the assumption of clean data underlying today's FL systems and the reality of noisy data in the real world poses a significant threat to the robustness and reliability of these FL systems, often leading to unexpected poor performance after their deployment.

The existing literature on FL robustness has mainly concentrated on two fronts: security against malicious adversaries (e.g., Byzantine or model poisoning attacks) [13, 58] and tolerance to label noise (i.e., incorrect annotations) [14, 23, 34, 35, 65–67]. However, it overlooks a more common challenge: the presence of natural or artificial non-malicious noise induced by the data generation and collection process itself. This type of noise corrupts the input data rather than its labels, and we refer to it with the term "data (generation) noise" in this paper. This leaves critical questions unanswered: **How does the performance of FL degrade under increasing levels of data noise? How does this degradation compare to that of CL under identical conditions? And, most importantly, what are the underlying mechanisms within the FL process that amplify this degradation?** As illustrated in Figure 1, the decentralized nature of FL, coupled with potential non-IID data distributions, could further exacerbate these questions.

Answering these questions is crucial for building robust FL systems in practice, but it presents several significant challenges. First, establishing a controlled experimental setup is non-trivial. A principled framework is needed to inject a wide range of common types of data noise in a modality-aware manner, with unified control over parameters such as the intensity of noise and the proportion of noisy FL clients. Second, ensuring that the findings are generalizable requires a comprehensive experimental effort, spanning multiple data modalities, model architectures, and diverse data noise configurations. Third, moving beyond surface-level observations to explain *why* FL being more sensitive to data noise compared to CL, which is one of our main findings, is a major analytical hurdle. It requires decomposing the complex FL pipeline and designing insightful metrics to diagnose how data noise impacts each individual stage, from local training to global aggregation.

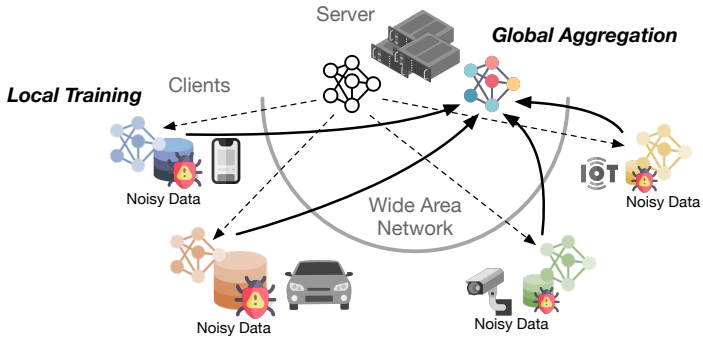


Fig. 1. **The vulnerability of federated learning (FL) to real-world data noise. Clients with noisy data (e.g., from vehicles, cameras, phones) can produce divergent model updates. The server, agnostic to the underlying data quality, aggregates these "polluted" model updates, leading to a significantly degraded global model. We identify two key "amplification" points in the FL training pipeline that increases the vulnerability of FL to data noise: (1) *local amplification*, where local models overfit to noise in private data owned by FL clients, leading to divergence of local model updates, and (2) *global amplification*, where the server combines divergent local model updates via "blind" model aggregation, leading to unstable training and poor convergence.**

In this paper, we systematically address these challenges to provide the first comprehensive study on the impact of data noise in FL. Our key contributions are the following.

- **Data noise injection toolkit.** We design and implement `DataNoiseGenerator`, a flexible and extensible open-source toolkit to enable the injection of controlled data noise into a variety of data modalities. It provides a unified framework for the research community to benchmark the robustness of FL systems in the presence of data noise.
- **Systematic empirical study.** We conduct the first large-scale systematic study comparing the impact of data noise on FL and CL. Our experiments cover five diverse data modalities: image, video, audio, text, and tabular data, with controlled noise injected by `DataNoiseGenerator`.
- **Key finding: FL is more vulnerable to data noise compared to CL.** Our empirical investigation reveals that FL is significantly more vulnerable to data noise than CL. This gap between FL and CL widens as the intensity of data noise and the proportion of noisy FL clients increase.
- **In-depth analysis of the root cause for the vulnerability of FL.** We provide a detailed analysis to pinpoint the root cause of this increased vulnerability or sensitivity of FL to data noise. We identify that the model aggregation by the FL server amplifies the negative impact of divergent updates from FL clients with noisy data, hindering global model convergence, especially under non-IID data distributions.

Paper Organization. The remainder of this paper is organized as follows. Section 2 provides a background on FL. Section 3 illustrates the design and implementation of our `DataNoiseGenerator` framework for the injection of various data noises for various data modalities. We describe the experimental methodology in Section 4 and present the results of the experimental evaluation on the vulnerability of FL to data noise in Section 5. In Section 6, we provide an in-depth analysis of the experimental results to identify the core reasons for the vulnerability of FL. We further examine the impact of data noise with different configurations of the FL training pipeline in Section 7. In

Section 8, we summarize the key takeaways from our study and discuss their implications for future research. We review related work in Section 9 and conclude the paper in Section 10.

Reproducibility. The artifacts that we developed for our study in this paper are available at **https://github.com/fardatalab/fl_data_quality**, including the `DataNoiseGenerator` toolkit, the example usage scripts, the data partitioning scripts, the ML model configurations, and the implementations of the FL training workflows.

2 Background and Problem Formulation

We provide the necessary background on the architecture of federated learning (FL) and its centralized learning (CL) counterpart. We then formally define data (generation) noise and motivate why the unique structure of FL makes it particularly susceptible to data noise, thereby formulating the core problem studied in this paper.

2.1 Architecture of Federated Learning

Federated learning (FL) is a decentralized machine learning (ML) paradigm designed to train a shared global model across numerous clients without centralizing their local data [41]. This approach preserves data privacy by design, as only model updates, not the raw data itself, are transmitted over the network.

The most common FL algorithm is federated averaging (FedAvg) [41]. A typical training round in FL proceeds in four steps:

(1) **Distribution.** The central server sends the current global model weights (w_t) to a selected subset K of clients.
(2) **Local training.** Each selected client k trains the model on its local dataset D_k for several epochs, producing an updated local model w_{t+1}^k.
(3) **Transmission.** Clients send their updated model weights w_{t+1}^k back to the server.
(4) **Aggregation.** The server aggregates the received updates to form a new global model, w_{t+1}. In FedAvg, this is a weighted average based on the size of each client's dataset ($n_k = |D_k|$):

$$w_{t+1} \leftarrow \sum_{k=1}^{|K|} \frac{n_k}{N} w_{t+1}^k, \text{where } N = \sum_{k=1}^{|K|} n_k. \tag{1}$$

This process repeats until the global model converges. Beyond the challenge of data noise, FL systems must also contend with non-IID (non-identically and independently distributed) data, communication bottlenecks, and heterogeneity among clients.

2.2 Centralized Learning as a Baseline

In contrast to FL, centralized learning (CL) represents the traditional ML approach where data are centralized and a single model is trained on top of the centralized data. CL serves as a useful reference baseline for FL, as it typically yields an ML model that is no worse than the one trained by FL. Therefore, the CL model quality serves as an *upper bound* of the corresponding FL model quality. Comparing FL to CL also allows us to isolate and quantify the performance impact introduced by the federated training process.

2.3 Data Noise in Federated Learning

In machine learning, noise can manifest in two primary forms: **label noise**, where the output labels are incorrect, and **data (generation) noise**, which refers to corruption in the input data itself. While label noise has been studied in the context of FL, our work in this paper focuses on the more pervasive but less understood impact of data noise on FL. This type of noise, caused by

various reasons such as sensor errors, environmental factors, or data entry mistakes, directly alters the data representation that the model learns from.

Data noise is an especially pressing concern in FL, as data are often generated on decentralized edge devices. This unique architecture of FL implies that it may exhibit an increased sensitivity or vulnerability to data noise compared to CL. We hypothesize three core mechanisms that can contribute to this vulnerability:

- **Server blindness to data quality.** In CL, a central authority can inspect, clean, and preprocess the raw data. In FL, the server is *blind* to the quality of client data. It can only observe the downstream effects of data noise through model updates, making detection and correction of noisy data challenging.
- **Local noise amplification.** During local training, an FL client updates the ML model for multiple epochs based on its private data. In the presence of data noise, the model can overfit to these corruptions. This process can *amplify* the effect of data noise, resulting in "polluted" model updates that diverge from desired model updates based on clean training data.
- **Compounding effect of non-IID data.** The non-IID nature of FL can exacerbate the impact of data noise. If a client holding the majority of data for a specific class label also contains noisy data, the ability of the global model to learn that class can be severely compromised. Unlike in CL, where noise might be canceled out over the entire global dataset, the impact of noisy data in FL can be highly concentrated, leading to performance degradation.

This hypothesized vulnerability of FL to data noise motivates our investigation. Although the general impact of data noise on ML is known, its specific ramifications for FL have not yet been explored. We aim to systematically quantify this vulnerability and dissect its root causes through a large-scale comparative study.

3 `DataNoiseGenerator`: A Toolkit for Injecting Data Noise

To evaluate of the impact of data noise on FL, we design and implement `DataNoiseGenerator`, an open-source framework for injecting a variety of data noise into datasets of different modalities. Beyond the scope of this study, `DataNoiseGenerator` can also serve as a reusable toolkit for the broad research community to benchmark the robustness of ML systems to data quality issues. We developed `DataNoiseGenerator` with the following properties.

Modality Awareness. Noise is intrinsically different between modalities. For example, image noise often involves pixel-level distortions, e.g., blur and overexposure, while noise in text documents occur at the character or token level, e.g., misspelled words and duplicates. Therefore, `DataNoiseGenerator` generates different noise for data in different modalities: images, videos, texts, audios, and tabular data. For each data type, we implement a set of noise functions parameterized with noise-specific knobs. This ensures that the injected noise is both realistic and relevant to the specific data domain.

Fine-grained Controls. `DataNoiseGenerator` provides fine-grained controls over the noise injection process along three dimensions:

(1) *Noise Proportion* $P \in [0, 1]$. This parameter controls the scope of the noise. It controls the *fraction* of the dataset whose data is subjected to noise, as shown in Table 1. This allows us to simulate scenarios ranging from a few isolated noisy data to widespread data quality issues.
(2) *Noise Intensity* $\mathcal{I}$. This parameter controls the magnitude of the corruption applied to each individual noisy data sample. It provides an intuitive knob to tune the *severity* of the noise, from minor to intense distortions.

Table 1. **Noise proportion P for each data modality. P denotes the fraction of data elements corrupted.**

Modality	Unit of corruption controlled by P
Image	Image files
Video	Video files (entire sequence)
Audio	Audio files
Text	10-word blocks
Tabular	Data points (rows)

(3) *Sample Coverage*. This parameter allows for applying the noise to a subset of the spatial region (for images and videos), temporal segment (for audios), tokens (for text documents), or attributes (for tables, same effect as $\mathcal{I}$) of each sample.

Unified Intensity. `DataNoiseGenerator` introduces a global intensity parameter, $\mathcal{I}$, to unify the intensities between different modalities and different noise types. This abstract parameter is scaled to a consistent discrete integer range (we use $[0, 10]$ in our experiments), where $\mathcal{I} = 0$ means no noise. The core of this design is a modality-specific mapping layer that translates this abstract intensity $\mathcal{I}$ into a concrete physical parameter for each noise function. For example, $\mathcal{I}$ might be linearly mapped to the standard deviation of Gaussian noise for tabular data, while being exponentially mapped to the attenuation factor for audio echo to better reflect perceptual severity. This parameter can greatly simplify a large fleet of experiments that involve different data modalities and noise types as in this paper.

3.1 Workflow and Usage

`DataNoiseGenerator` is implemented as a Python-based, extensible command-line tool, leveraging popular libraries such as OpenCV for processing images and videos, Librosa for processing audio, and Pandas for processing tabular data. Its modular architecture, depicted in Figure 2, is designed for both ease of use and extensibility.

The system comprises three main components:

- **Command-line interface (CLI).** This is the entry point for the user. It uses a standard argument parser to collect all configurations, including the data modality, the input and output paths, and a list of desired noise functions. Each noise function comes with its specified intensity $\mathcal{I}$ and a global proportion P. This design allows for applying multiple types of noise in a single run.
- **Core orchestration engine.** This is the control plane of the system. It is responsible for (1) parsing and validating user input, (2) discovering and loading the appropriate modality-specific generator based on the user's request, (3) iterating over the input dataset and deciding whether to apply noise to each sample based on the proportion parameter P, and (4) managing data loading and saving efficiently.
- **Modality-specific generators.** These are pluggable modules, one for each data modality. Each generator is a class that inherits from a common base, ensuring a consistent interface. It contains a collection of methods, each implementing a specific noise function (e.g., *apply_blur, add_typos*). This modular design makes `DataNoiseGenerator` highly extensible—adding a new type of noise simply requires adding a new method to the corresponding generator, while supporting a new modality involves creating a new generator class.

The end-to-end use of `DataNoiseGenerator` is streamlined for rapid experimentation. For example, to apply a medium-intensity blur and darkening to 40% of an image dataset, a user can execute the following command via the CLI provided by `DataNoiseGenerator`:

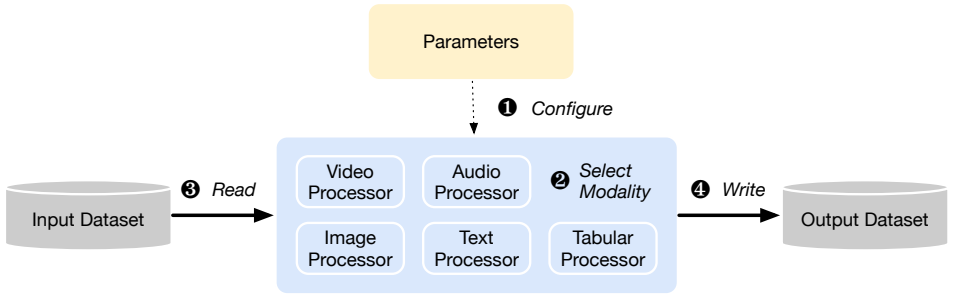


Fig. 2. **Workflow of `DataNoiseGenerator`. The user interacts with `DataNoiseGenerator` via a CLI, specifying the modality, the data paths, and the desired noise types with their intensity ($\mathcal{I}$), proportion (P), and sample coverage. The core engine parses these input parameters and dispatches tasks to the appropriate modality-specific noise generator, which contains a library of noise functions. The engine manages data I/O and ensures that the correct proportion of data is processed.**

```
$ python data_noise_generator.py --image \
  --input_path /path/to/clean_cifar10 \
  --output_path /path/to/noisy_cifar10 \
  --noise blur darken --intensity 4 --portion 0.4
```

Upon execution, `DataNoiseGenerator` reads the data from the input path, shuffles the file list, applies the specified blur function (with parameters mapped from $\mathcal{I} = 4$) to the first 40% of the samples, and saves the complete dataset, with both modified and untouched samples, to the output directory. This ensures that the output is a self-contained, ready-to-use dataset, eliminating the need for manual data merging and simplifying integration into existing ML pipelines. Therefore, `DataNoiseGenerator` can be easily integrated with existing FL benchmarks, e.g., LEAF [6] and FL-Bench [62], which decouple dataset generation and preprocessing from data loading and FL setup. `DataNoiseGenerator` can be applied to add customized noise to the dataset before it is ingested into the FL training pipeline. For example, CIFAR-10, a dataset in `torchvision` [51], can be preprocessed by `DataNoiseGenerator` via the above command and then loaded and split into training and testing sets by the FL-Bench framework.

3.2 Supported Types of Data Noise

To cover a wide range of realistic data quality issues, `DataNoiseGenerator` implements an extensive "zoo" of more than 20 distinct types of data noise across the five data modalities. The selection of these types of data noise was guided by common issues reported in the data collection and processing literature. For example, our image noise includes that from sensor limitations (e.g., darkening), environmental conditions (e.g., overexposure), and post-processing artifacts (e.g., watermarks). Similarly, text noise simulates human mistakes (e.g., typos) and automated processing failures (e.g., noise tokens from ASR). Table 3 provides a representative summary of these supported types of data noise, highlighting their causes in the real world and, more crucially, the explicit mappings of the unified intensity parameter $\mathcal{I}$ to their underlying physical parameters. Table 2 visually demonstrates the tangible effects of this controllable intensity, showcasing how `DataNoiseGenerator` can generate a spectrum of corruptions from "low" to "high."

4 Methodology

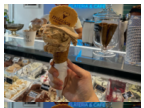
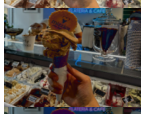
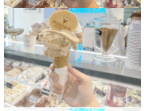
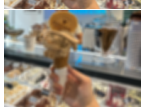
4.1 Experimental Pipeline and Setup

Our evaluation pipeline consists of three stages for each experiment: (1) data partitioning, (2) noise injection, and (3) model training.

Table 2. **Examples of data quality issues generated by `DataNoiseGenerator` for image and text modalities. We showcase three levels of intensity ("low", "medium", and "high") for various noise types, demonstrating the ability of `DataNoiseGenerator` to inject controlled data noise.**

Image Data Quality Issues

Original

	Blur	Darken	Exposure	S&P	Watermark
Low					
Medium					
High					

Text Data Quality Issues

Original	*What's in a name? That which we call a rose by any other name would smell as sweet.*			
	Duplicates	**Missing Words**	**Random Casing**	**Typos**
Low	What's What's in a name? That which we call a rose by by any other name would smell as as sweet.	What's in a name? That which we call a rose any other name would smell sweet.	What's IN a name? That which we call a rose by any other name would smell as SWEET.	Wxat's in a name? That wxich we call a rose bh any other name would smebl is sweet.
Medium	What's What's in a a name name? That which we we call a rose by by any any other other name would smell as as sweet.	What's in a name? THAT which we call a rose by ANY other name would smell AS SWEET.	What's in A name? That which we call a rose by any other NAME WOULD smell as sweet.	What's ia a naml? Tzat which le call a rose bh and other name woubd shell as sxeet.
High	What's What's in a a a name name? That which we we call a rose by by any any other other name would smell as as sweet sweet.	What's in a name? That which call a rose name would smell.	What's in A name? That WHICH we CALL a rose by ANY other name would smell AS SWEET.	What's ia a naml? Tzat mhich wu yall n rose bh and other name woubd shell as sxeet.

Data Partitioning. To simulate a realistic federated environment, we distribute data among $C = 10$ clients using a non-IID partitioning strategy. We use a Dirichlet distribution $Dir(\alpha)$ over the class labels to assign data samples to clients, which has been a standard method in FL to model skewed data distributions. We set the concentration parameter $\alpha = 1.0$, which yields a moderate and realistic heterogeneous data distribution across FL clients.

Table 3. **A selection of supported noise types by `DataNoiseGenerator` and their mappings from the unified intensity parameter $\mathcal{I} \in [0, 10]$. This design enables controlled and comparable cross-modal experiments on the impact of data noise for FL. For tabular data, K is the total number of features and K_{cont} is the number of continuous features.**

Modality	Noise Type	Description	Unified Intensity $\mathcal{I}$ Mapping to Physical Parameter(s)
Image	Blur	Gaussian blur that averages neighboring pixels.	Blur kernel size $(\lfloor \mathcal{I} \rfloor, \lfloor \mathcal{I} \rfloor)$
	Darkening	Linear brightness offset that uniformly darkens the image.	Brightness offset $\beta = -10\,\mathcal{I}$
	Salt-and-Pepper	Random white/black pixel flips across the image.	Number of flipped pixels $\approx 0.02 \times$ (image height $\times$ width) $\times\, \mathcal{I}$
	Exposure	Gamma correction applied to brighten the image.	Gamma correction exponent $\gamma = \frac{10.5-\mathcal{I}}{10}$
	Watermark	Centered watermark overlay using alpha blending.	Alpha blending factor $\alpha = \mathcal{I}/10$
	Rotation	Discrete-angle rotation about image center.	Rotation angle θ from mapping$\{1 \to 36^\circ, 2 \to -36^\circ, \ldots, 10 \to 180^\circ\}$
Video	Blur	Per-frame Gaussian blur that averages neighboring pixels.	Blur kernel size $\approx (k, k)$, $k = \text{odd}(\lfloor 30.1\,\mathcal{I} \rfloor)$
	Shake	Random per-frame subpixel shifts and rotations.	Max translation $10\,\mathcal{I}$ pixels; max rotation $18\,\mathcal{I}^\circ$
	Darkening	Linear brightness offset that uniformly darkens each frame.	Brightness offset $\beta = -10\,\mathcal{I}$
	Salt-and-Pepper	Random white/black pixel flips applied independently per frame.	Number of flipped pixels $\approx$ (frame height $\times$ width) $\times\, (\mathcal{I}/10)$
	Exposure	Gamma correction applied per-frame to brighten the video.	Gamma correction exponent $\gamma = 1 - \mathcal{I}/10$
	Watermark	Centered watermark overlay via alpha blending on each frame.	Alpha blending factor $\alpha = \mathcal{I}/10$.
Audio	Gaussian Noise	Additive white Gaussian noise over the waveform.	Standard deviation $\sigma = 0.1\,\mathcal{I}$
	Echo	Adds progressively fading, delayed repetitions of the audio.	Delay $= 2000 - 190\,\mathcal{I}$ ms; gain per repeat $= 0.1\,\mathcal{I}$
	Silence	Mutes the centered segment of the signal.	Silenced fraction $= 0.1\,\mathcal{I}$ of samples
	Lower Volume	Scales waveform amplitude toward zero.	Volume scale factor $= 1 - 0.1\,\mathcal{I}$
	Background Mix	Mixes a background track with volume adjusted by intensity.	Background volume $= 0.1\,\mathcal{I}$ (trim/pad to length)
Text	Typos	Random per-word character substitutions.	Per-word prob $= \mathcal{I}/10$.
	Missing Words	Random deletions of words.	Per-word prob $= \mathcal{I}/10$.
	Random Casing	Randomly toggles words to UPPER/lower case.	Per-word prob $= \mathcal{I}/10$.
	Duplicates	Randomly inserts repeated tokens.	Per-word prob $= \mathcal{I}/10$.
Tabular	Gaussian Noise	Adds zero-mean noise to continuous features.	*Scope:* $\lfloor K_{\text{cont}} \cdot (\mathcal{I}/10) \rfloor$ features; Standard deviation $\sigma = \frac{\text{range}}{5} \cdot (\mathcal{I}/10)$.
	Uniform Noise (Cont.)	Moves values toward random in-range targets.	*Scope:* $\lfloor K \cdot (\mathcal{I}/10) \rfloor$ features; Interpolate with factor $\mathcal{I}/10$.
	Uniform Noise (Cat.)	Replaces categories with randomly sampled alternatives.	*Scope:* $\lfloor K \cdot (\mathcal{I}/10) \rfloor$ features.
	Adversarial (Cont.)	Pushes values toward min/max based on median split.	*Scope:* $\lfloor K \cdot (\mathcal{I}/10) \rfloor$ features; Interpolate to $\{\min, \max\}$ with $\mathcal{I}/10$.
	Adversarial (Cat.)	Replaces each category with the next in a fixed cyclic order.	*Scope:* $\lfloor K \cdot (\mathcal{I}/10) \rfloor$ features.

The numbers of samples between clients are controlled by another Dirichlet distribution $Dir(\beta)$. A smaller β value indicates a higher skew (i.e., a few clients own most of the data), while a larger value indicates a more uniform distribution. By default, all clients own an equal share of the training samples, i.e., β is set to ∞. We evaluate the effect of β in Section 7.2.

Noise Injection. Using `DataNoiseGenerator` (Section 3), we inject noise into the training data of FL clients. This simulates a scenario where data quality issues are systemic throughout the entire data population. The noise is therefore distributed across all clients. We investigate the impact of data noise along two dimensions corresponding to the two control knobs highlighted in Section 3:

- Noise proportion P, which controls the fraction of noisy samples in the training data. Consequently, each client's local data contains approximately this proportion of noisy samples. We test three levels of noise proportion: *low* ($P = 10\%$), *medium* ($P = 40\%$), and *high* ($P = 70\%$).
- Noise intensity $\mathcal{I}$, which applies noise at a specific severity level for each noisy sample (irrespective of which client it belongs to). We investigate three intensity levels: *low* ($\mathcal{I} = 1$), *medium* ($\mathcal{I} = 4$), and *high* ($\mathcal{I} = 7$). As detailed in Table 3, these values are mapped to modality-specific parameters.

We keep the default 100% for the coverage parameter to control the effect of noise at the granularity of samples. This setup creates a grid of nine data noise configurations, which we evaluate and compare with a baseline using clean data ($P = 0$).

Training with FL and CL. By default, we use the classic FedAvg algorithm [41] for global model aggregation in our FL experiments. In each communication round, all 10 clients participate in training, performing $E = 1$ local epochs on their (partially noisy) data before sending updates to the server. We also develop a CL baseline that serves as a reference point for FL performance, where we train the same model on the entire dataset, which is subject to the same overall proportion and intensity of data noise. We use the Adam optimizer and perform a hyperparameter tuning to find the optimal learning rate for both FL and CL. All ML models are trained until their validation performance plateaus.

Table 4. **Summary of the main datasets, tasks, and ML models used in our experimental evaluation.**

Dataset	Modality	Task	Type	# Samples	Model Architecture
UCF101 [61]	Video	Action Recognition	Classification	13k videos	VideoMAE (ViT-B) [64]
Something-Something [18]				221k videos	
CIFAR-10 [31]	Image	Object Recognition	Classification	60k images	PyramidNet [20]
Pascal VOC 2012 [4]		Object Detection	Regression	7k images	Faster R-CNN [53]
Shakespeare [38]	Text	Next-Word Prediction	Classification	~422k lines	Alpaca-7B [63]+ LoRA [21]
AG News [2]				~120k lines	
ESC50 [49]	Audio	Sound Recognition	Classification	2k recordings	Audio Spectrogram Transformer (AST) [16]
UrbanSound8K [56]				8k recordings	
Phishing Websites [43]	Tabular	Phishing Detection	Classification	11k records	ResNet-like [17]
Housing Prices [46]		Price Prediction	Regression	21k records	MLP Regressor [57]

4.2 Datasets, Tasks, and Metrics

To ensure that our findings are generalizable, we select two public datasets for each modality and cover a diverse range of data modalities. Table 4 summarizes the characteristics of the datasets and the corresponding ML model architectures. For performance metrics, we use top-1 accuracy for all classification tasks and R^2 (coefficient of determination) for all regression tasks.

4.3 FL Emulation

We use PyTorch [47] for model implementation and training. Federated learning is emulated by Flower [3], a popular FL framework that provides flexible configuration, scalable execution, and high fidelity. We run all of our experiments with NVIDIA A100 GPUs.

5 Evaluation Results

In this section, we present the empirical results of our large-scale comparative study. The experiments are designed to systematically answer our core research questions regarding the performance of FL (compared to CL) in the presence of data noise. We show that FL exhibits an elevated vulnerability to data noise in all tested modalities, noise types, and configurations.

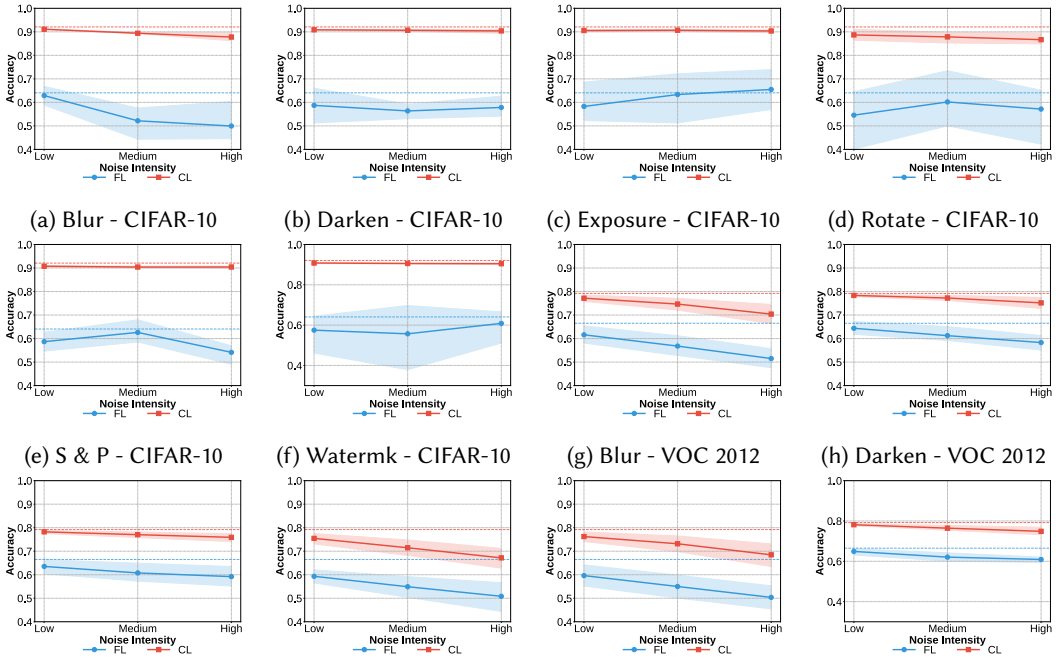
5.1 A Deep Dive into Results from Image Data

We conduct a comprehensive analysis of the image classification task on CIFAR-10 and object detection task on Pascal VOC 2012 to evaluate the resilience of the trained models to noisy data. Our experimental setup is designed to simultaneously probe the effects of both noise intensity $\mathcal{I}$ and the proportion of noisy clients P. For each level of noise intensity displayed on the x-axis, we vary the noise proportion across a range of values $P \in \{10\%, 40\%, 70\%\}$. The resulting performance is aggregated and visualized in Figure 3. The solid line in each plot represents the mean accuracy across the different noise proportions, while the shaded area illustrates the variation of the accuracy, effectively showing the sensitivity of model accuracy to the fraction of corrupted data. For reference, the dashed horizontal line shows the accuracy on clean data.

The results in Figure 3 characterize the different behaviors of CL and FL under noisy conditions.

Exceptional Robustness of Centralized Learning. CL demonstrates remarkable resilience. Its accuracy remains consistently high (near 0.9) in the classification task (Figures 3a–3f) for all noise types and intensities tested and is close to the clean CL baseline (shown as the red dashed line). Furthermore, the almost invisible shaded region around the line indicates that its performance is stable and largely unaffected by the proportion of noisy data in the training set. This establishes CL as a robust baseline that is capable of effective learning even when a significant fraction of its data is corrupted. These findings generally apply to the object detection task (Figures 3g–3l).

Federated Learning's Sensitivity and Instability. In contrast, FL exhibits a profound vulnerability.



(a) Blur - CIFAR-10 (b) Darken - CIFAR-10 (c) Exposure - CIFAR-10 (d) Rotate - CIFAR-10

(e) S & P - CIFAR-10 (f) Watermk - CIFAR-10 (g) Blur - VOC 2012 (h) Darken - VOC 2012

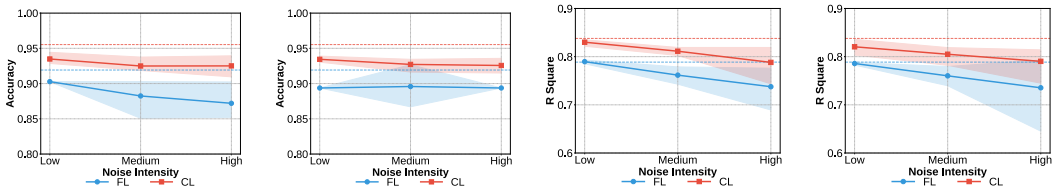
(i) Exposure - VOC 2012 (j) Rotate - VOC 2012 (k) S & P - VOC 2012 (l) Watermk - VOC 2012

Fig. 3. **Test accuracy vs. noise intensity ($\mathcal{I}$) on image datasets for six different noise types. For each intensity level, results are aggregated over varying noise proportions $P \in \{10\%, 40\%, 70\%\}$. The solid line indicates the mean accuracy across these proportions, while the shaded region represents the variation. *Key Observations:* (1) The CL model is exceptionally robust, showing high accuracy and low variation to noise intensity or proportion. (2) The FL model's mean accuracy is substantially lower and, for some noise types (blur, rotate, salt & pepper), degrades with increasing intensity. (3) The vast shaded area for FL reveals extreme sensitivity to the *proportion* of noisy clients, indicating significant performance instability.**

First, the mean accuracy of FL is significantly lower than that of CL across the board, and the gap to the clean FL baseline (shown as the blue dashed line) is also significantly larger. This performance gap is substantial for both classification and regression tasks even with low noise intensity and widens as noise becomes more severe, as seen in the downward trends for the noise types "blur," "rotate," and "salt & pepper." This confirms that FL is more sensitive to the severity of data corruption.

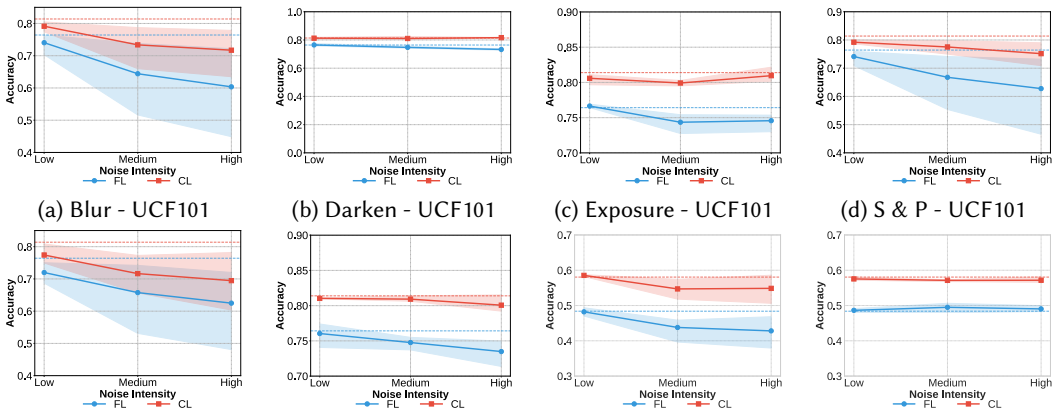
Second, and more critically, the vast shaded area of FL reveals its instability with respect to the proportion of noisy clients. This large variation indicates that the performance of FL is unpredictable—the final accuracy can vary dramatically depending on the amount of corrupted data. For example, with medium intensity, the accuracy of FL can be relatively high or drastically low, completely dependent on the noise proportion P. This high variation is a direct consequence of the federated averaging process, which struggles to aggregate conflicting updates from different clients, making the training outcome unreliable and highly dependent on the specific data distribution across the network (see Section 6).

In summary, our findings on the image tasks clearly show that FL is more vulnerable to data noise than CL. It suffers not only from a lower mean accuracy that degrades with noise intensity but also from a critical lack of stability, where its efficacy is highly dependent on the proportion of



(a) Uniform - PW (b) Adversarial - PW (c) Uniform - HP (d) Adversarial - HP

Fig. 4. **Test accuracy vs. noise intensity on tabular datasets. "PW" refers to the "Phishing Websites" dataset, and "HP" to the "Housing Prices" dataset. Each plot follows the same convention in Figure 3.**



(a) Blur - UCF101 (b) Darken - UCF101 (c) Exposure - UCF101 (d) S & P - UCF101

(e) Shaky - UCF101 (f) Watermark - UCF101 (g) Blur - SS (h) Darken - SS

(i) Exposure - SS (j) S & P - SS (k) Shaky - SS (l) Watermark - SS

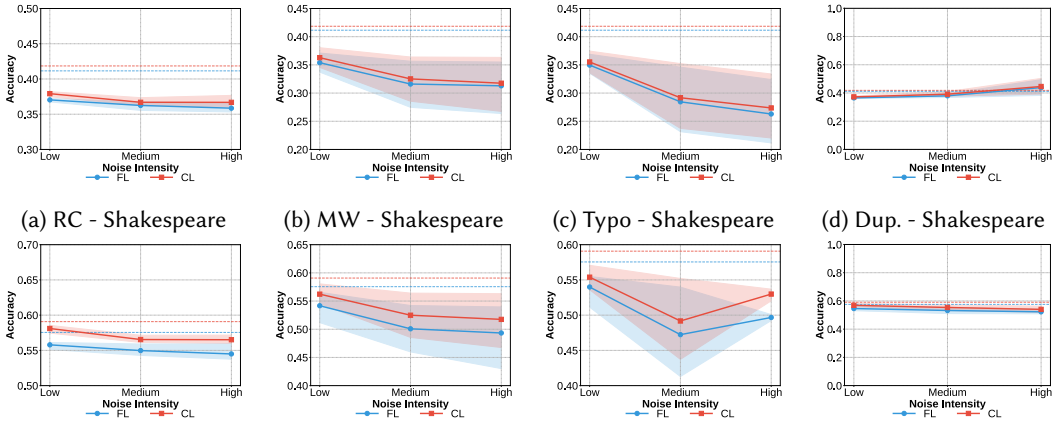
Fig. 5. **Test accuracy vs. noise intensity on video datasets. "SS" refers to the "Something-Something" dataset. Each plot follows the same convention in Figure 3.**

noisy clients. This dual vulnerability highlights a fundamental challenge for the deployment of FL systems in heterogeneous real-world environments where data quality cannot be guaranteed.

5.2 The Vulnerability of FL Across Modalities

To verify that the increased vulnerability of FL observed with image data is a general phenomenon rather than a modality-specific artifact, we extend our comprehensive study to video, text, audio, and tabular data. Figures 4–7 present the results with 16 distinct, modality-appropriate noise types on different datasets. Each plot follows the same convention as our analysis of the image data, illustrating the mean accuracy, its variation with respect to different proportions of noise $P \in \{10\%, 40\%, 70\%\}$ as a function of noise intensity, and its gap to the accuracy on clean data.

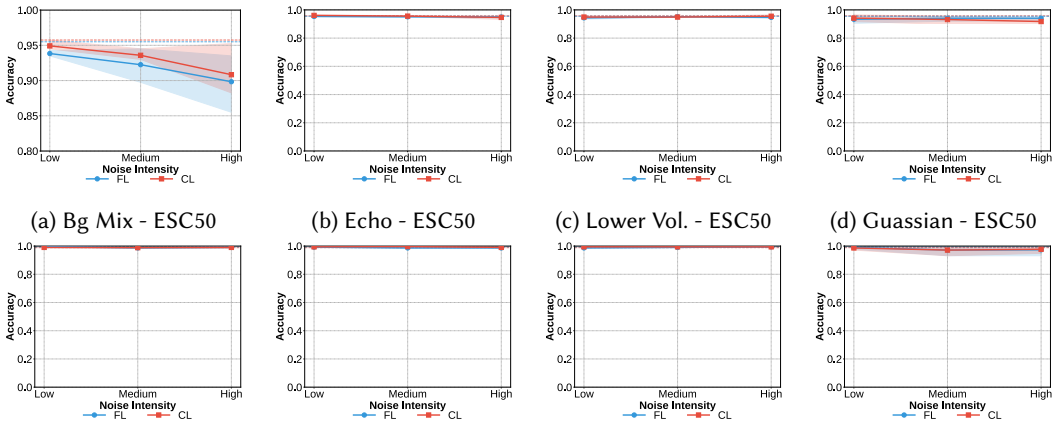
The extensive results provide clear evidence supporting our central hypothesis. The fragility of FL and the resilience of CL are not confined to image tasks but are consistent in all modalities.



(a) RC - Shakespeare (b) MW - Shakespeare (c) Typo - Shakespeare (d) Dup. - Shakespeare

(e) RC - AG News (f) MW - AG News (g) Typo - AG News (h) Dup. - AG News

Fig. 6. **Test accuracy vs. noise intensity on text datasets: "Shakespeare" and "AG News." "RC", "MW", and "Dup." refer to Random Casing, Missing Words, and Duplication, respectively. Each plot follows the same convention in Figure 3.**

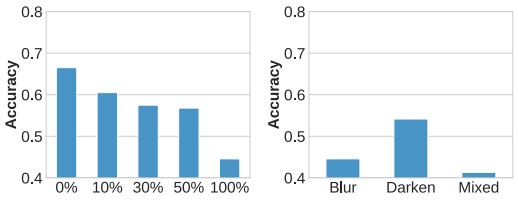


(a) Bg Mix - ESC50 (b) Echo - ESC50 (c) Lower Vol. - ESC50 (d) Guassian - ESC50

(e) Bg Mix - US8K (f) Echo - US8K (g) Lower Vol. - US8K (h) Gaussian - US8K

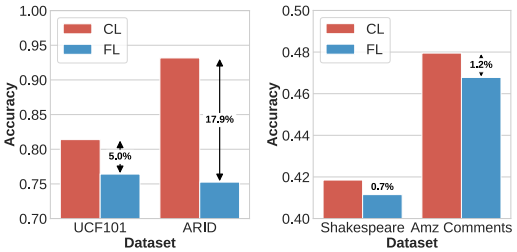
Fig. 7. **Test accuracy vs. noise intensity on audio datasets. "US8K" refers to the UrbanSound8K dataset. Each plot follows the same convention in Figure 3.**

Universal Performance Gap and Sensitivity. From *Tabular* (Figure 4) to *Video* (Figure 5) and *Text* (Figure 6), a significant performance gap between CL and FL is clear, which is also larger than the gap between CL and FL on clean data. CL consistently maintains high and stable accuracy, appearing as a nearly flat line at the top of most plots, demonstrating its profound robustness. In contrast, FL not only starts with a lower baseline accuracy, but also frequently exhibits a clear downward trend as noise intensity increases. This degradation is particularly visible in, for example, *Tabular-Classification-Uniform* (Figure 4a), *Tabular-Regression-Adversarial* (Figure 4d), *Video-Blur* (Figures 5a and 5g), and *Video-Shaky* (Figures 5e and 5k), confirming that FL is more severely impeded by the intensity of data corruption, regardless of the data modality and ML task. However, we also observe that for certain types of noise, such as *Audio-Echo* and *Audio-Lower Volume*, both CL and FL are robust.



(a) % of noisy clients. (b) Mixed types of noise.

Fig. 8. **Impact of heterogeneous noise setup.**



(a) Video (b) Text

Fig. 9. **CL and FL on naturally noisy datasets.**

Pervasive Instability of Federated Learning. More critically, the defining characteristic of FL's vulnerability, its instability, persists across all modalities. The wide blue shaded areas indicate high variation, revealing that the outcome of FL is highly dependent on the proportion of noisy data. For example, the large variation in *Video-Blur* (Figures 5a and 5g) and *Text-Random Casing* (Figures 6a and 6e) highlights this unpredictability. This is in contrast with the stable performance of CL. This instability of FL suggests that the federated averaging mechanism is fundamentally challenged when aggregating model updates derived from corrupted, multi-modal data (see Section 6 for an in-depth analysis).

In summary, our cross-modality, cross-task, and cross-dataset investigation confirms that the elevated sensitivity and instability of FL in the presence of data noise is a fundamental characteristic of the learning paradigm. This vulnerability is independent of the modality of the data, the specific learning task, and the type of noise. It underscores a systemic challenge that must be addressed for the reliable deployment of FL in real-world applications where diverse data sources of varying quality are the norm.

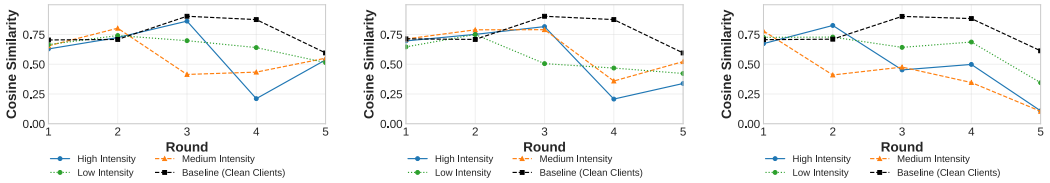
5.3 Heterogeneous Noise

We further vary the noise setup to introduce more heterogeneity. Specifically, we first vary the fraction of FL clients that have noisy data (*Image-Blur* on CIFAR10 with $\mathcal{I} = 7$ and $P = 70\%$) and observe how the final model accuracy changes. Figure 8a shows a clear trend: the accuracy of FL decreases as more clients have noisy data, which confirms the propagation of the negative impact of data noise from individual clients to the global model.

We also introduce heterogeneity to noise types between clients: instead of having the *Blur* noise on all clients, we change the noise type for half of the clients to *Darken* such that different clients can have different types of noise. Figure 8b compares the accuracy of FL with *All-Blur*, *All-Darken*, and *Mixed-Blur-Darken*: while *Darken* alone has less severe impact than *Blur*, mixing them together causes the highest accuracy degradation. This result shows that heterogeneity of noise types can bring additional challenges to FL.

5.4 Real-world Data Noise

To investigate the vulnerability of FL to data noise in the wild, we further include two real-world datasets that are not cleaned and are thus naturally noisy: **ARID** [71], a video dataset collected for action recognition in dark environments, e.g., surveillance or driving vehicle camera footage at night, and **Amazon Comments** [33], a text dataset that consists of real customers' comments on Amazon products and reflects multiple types of text noise in Table 3. We perform the video and text tasks in Table 4 with CL and FL on the two datasets, respectively, and measure how model accuracy degrades from CL to FL. For reference, we also show CL and FL accuracy on the clean UCF101 and Shakespeare datasets.



(a) $P = 10\%$ (b) $P = 40\%$ (c) $P = 70\%$

Fig. 10. ***Average cosine similarity between client gradients.*** **The plots show the average cosine similarity versus communication rounds under different proportions P of noisy data. The baseline (black, clean clients) maintains high similarity. As noise intensity increases (from "low" to "high"), the similarity between client gradients drops significantly, indicating that noise causes clients to learn in conflicting directions. The trend becomes clearer as P increases. The experiments are conducted with the tabular classification task on the "Phishing Websites" dataset with the *Uniform* noise.**

Figure 9 reports the converged test accuracy. For the video task (Figure 9a), the natural noise present in ARID has more significant impact on FL: there is a 17.9% accuracy degradation from CL to FL, while the gap is much smaller on UCF101 (5%). The same observation can also be made in the text task (Figure 16b). While a model's behavior on specific datasets is the result of a conflux of various factors, these observations align with the findings of the controlled experiments in the previous sections: the vulnerability to data noise is magnified when the model is trained by FL.

6 Analysis of FL's Vulnerability to Data Noise

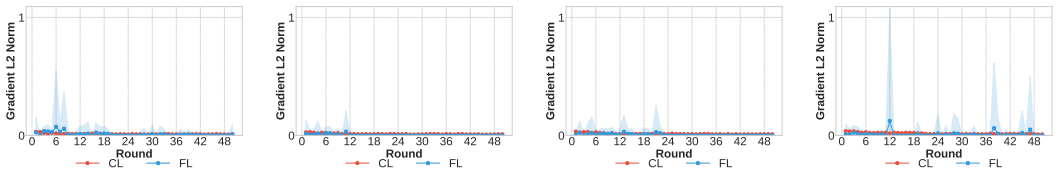
So far, we have empirically established that FL is more vulnerable to data noise than CL. We now perform an in-depth analysis to pinpoint the root causes of this elevated vulnerability. Our analysis identifies two key phenomena that collectively explain this sensitivity: (1) noise induces **divergence** among local client updates, causing them to learn in conflicting directions, and (2) noise generates updates with excessively large magnitudes, leading to **instability** upon model aggregation.

6.1 Divergence of Local Model Updates

In an ideal FL scenario with clean data, clients' local gradients should be generally aligned, pointing towards a common direction for model update. However, when local models are trained with noisy data, they begin to overfit to the local data corruptions, causing their update directions to diverge from each other.

To quantify this disagreement among clients, we measure the average *cosine similarity* of the gradients used for local model updates. This metric specifically captures the angular difference between the directions taken by local model updates, isolating the effect of direction from magnitude. A low cosine similarity indicates a strong disagreement on the direction of the model update.

Figure 10 illustrates this phenomenon. In the scenario with only clean clients (black dashed line), the average similarity remains high as training progresses, showing a strong consensus. However, the introduction of noise causes a severe drop in agreement. We observe that the average cosine similarity on clean data is consistently higher than that on noisy data across different proportions $P = 10\%, 40\%, 70\%$. The trend becomes more apparent when P increases (e.g., 70%): as the noise intensity increases from zero (clean) to "low" and to "medium" and "high," the average cosine similarity drops. This directional conflict in training forces the global model to average out conflicting signals, hindering convergence and degrading performance. This provides strong evidence that noise-induced directional divergence is a primary cause of FL's vulnerability.



(a) Clean ($P = 0\%$) (b) $\mathcal{I} = 4, P = 10\%$ (c) $\mathcal{I} = 4, P = 40\%$ (d) $\mathcal{I} = 4, P = 70\%$

Fig. 11. ***Distribution of L2 norms of gradients*. The plots compare the L2 norm of the gradient for CL (red line) with the min-max range of norms for FL (blue shaded area) under medium noise intensity $\mathcal{I} = 4$. CL exhibits a stable, low norm. In contrast, the gradient norms of FL clients show a wider distribution with significant intermittent spikes (outliers), indicating instability of the training process. The experiments are conducted on tabular data with uniform noise.**

6.2 Unstable Gradients and Training

Beyond the directional divergence, noise also destabilizes the FL training process by affecting the magnitude of the gradients. In gradient-based optimization, updates with excessively large norms can cause drastic and unstable jumps in the parameter space, disrupting the learning process.

We investigate this by comparing the *L2 norm* of the aggregated single gradient in CL with the distribution of the norms of local client gradients in FL. As shown in Figure 11, the norm of the CL gradient (red line) remains stable and relatively small. In contrast, the norms of the local FL client gradients (blue shaded area representing the min-max range) exhibit significant volatility, especially when the proportion of noisy data is large.

The most striking observation is the appearance of sharp, intermittent spikes in the maximum gradient norm as the proportion of noisy data P increases from 10% to 70% (Fig. 11b-d). In FedAvg, these large-magnitude updates can dominate the aggregation step. This explains why FL's training process is less stable than CL's.

7 Impact of FL Configurations

Having established the general vulnerability of FL to data noise in comparison to CL, we now turn our attention to how this vulnerability is modulated by different choices of FL configurations. The decentralized nature of FL introduces several key hyperparameters that govern the training process, such as the degree of data heterogeneity, the number of participating clients, and the client sampling strategy. Understanding their interplay with data noise is critical for deploying robust FL systems in practice.

In this section, we conduct a series of controlled experiments to dissect the impact of these configurations. We use *relative accuracy*, defined as the ratio between the accuracy of the model for noisy data and that for clean data, as our primary metric. A lower relative accuracy indicates a higher sensitivity to noise. All experiments in this section are performed with image classification on CIFAR-10 using the noise type "**blur**," unless otherwise specified.

7.1 Data Heterogeneity (Non-IID vs. IID)

We first investigate the role of data heterogeneity, which is ubiquitous in practical FL applications. We control the degree of non-IID data distribution by varying the `alpha` of the Dirichlet distribution.

As shown in Figure 12, the degree of non-IID distribution has a pronounced effect on the robustness of FL. Specifically, a small `alpha` (e.g., 0.1), corresponding to a more skewed non-IID setting, leads to a lower relative accuracy. This indicates that **the heterogeneity of the data exacerbates the vulnerability of FL to noise**. In a highly non-IID setting, each client's local model is already specialized to a narrow data distribution. When local data is also noisy, the model

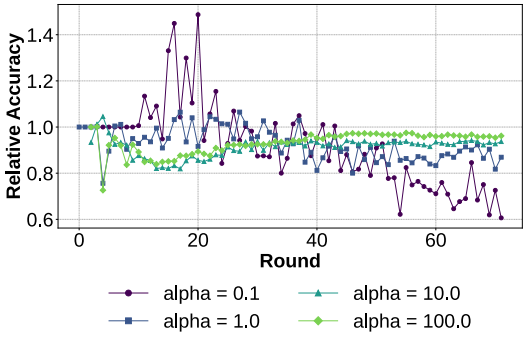


Fig. 12. **Impact of class skew.**

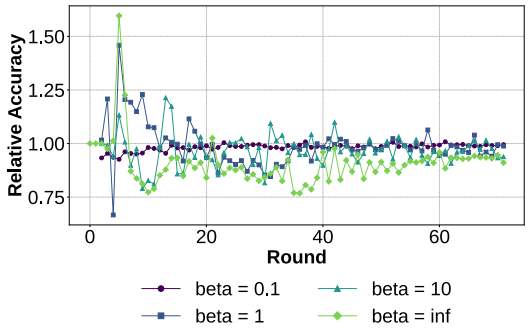


Fig. 13. **Impact of volume skew.**

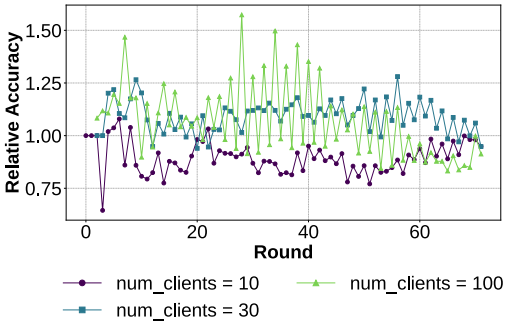


Fig. 14. **Impact of #clients.**

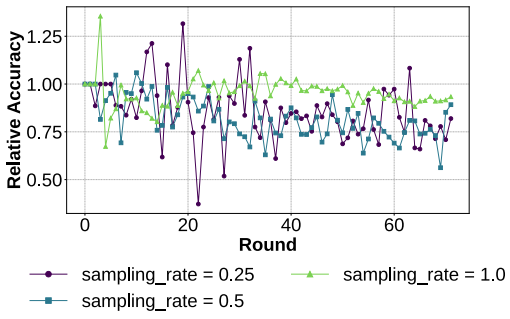
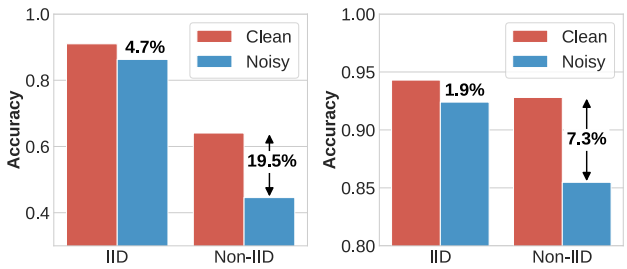


Fig. 15. **Impact of sampling rate.**



(a) Image (b) Tabular

Fig. 16. **IID vs. non-IID distributions.**

is more likely to generate divergent updates. The server, when aggregating these conflicting local model updates, struggles to find a good direction for global model updates, thus amplifying the negative impact of data noise.

We further investigate the impact of data noise on FL with IID distribution, compared to that with non-IID distribution (default `alpha` 1.0). Note that IID distribution can be viewed as a special case of non-IID distribution by setting `alpha` to ∞. Specifically, we measure the accuracy of FL for the image task on clean and noisy (*Blur*) CIFAR-10 and the tabular task on clean and noisy (*Uniform*) Phishing Websites. The noise is configured with $\mathcal{I} = 7, P = 70\%$. Figure 16 reports the results. For both modalities, the accuracy degradation from clean to noisy data is significantly larger with non-IID distribution: 4.7% (IID) vs. 19.5% (non-IID) for image and 1.9% (IID) vs. 7.3% (non-IID) for tabular. This experiment confirms the above finding: the vulnerability of FL to data noise is exacerbated with non-IID distribution.

7.2 Data Volume Skew

Next, we examine the effect of data volume skew by varying the parameter β (see Section 4.1) such that different clients can hold different numbers of data samples.

The results, presented in Figure 13, reveal a somewhat counter-intuitive trend. A higher volume skew with `beta` = 0.1 results in a higher relative accuracy and a very stable training curve, suggesting that the system is *less* sensitive to noise in this configuration. As data volume becomes more balanced across clients (i.e., `beta` increases), the relative accuracy drops. This phenomenon can be explained by the mechanics of the FedAvg algorithm. When data volume is highly skewed, clients with a large number of samples dominate the weighted averaging process during global model aggregation. The training process becomes closer to a centralized setup on top of the data owned by these few large clients. Consequently, noisy updates from smaller clients have a negligible impact on the global model, making the overall system appear more robust to data noise.

7.3 Number of FL Clients

The number of FL clients is another critical configuration parameter in FL. We evaluated the sensitivity of FL to data noise with 10, 30, and 100 clients, while keeping the total data volume and data distribution constant in all experiments.

Figure 14 presents the results. As the number of clients increases, both the average relative accuracy and its volatility increase. With 10 clients, the relative accuracy remains consistently below 1.0, indicating a stable but negative impact from data noise. In contrast, with 30 and 100 clients, the relative accuracy remains unstable throughout the lifetime of FL training, resulting in a lower relative accuracy at the end of training. This phenomenon can be attributed to the amount of data available per client. With a smaller number of clients (e.g. 10), each client holds a larger dataset. Their local updates are more stable and less prone to noise, but noise can still degrade the overall accuracy of the trained model. Conversely, when the number of clients is large (e.g. 100), each client owns a much smaller slice of the data. This makes local model updates much more sensitive to the noise present in the local data. This instability leads to a high variation in the accuracy of the model, which causes the wild fluctuations observed in the training curve.

7.4 Client Sampling Rate

In many practical FL scenarios, only a fraction of clients are sampled to participate in each round of training to save communication and computation costs. We investigate how this client sampling rate affects the sensitivity of FL to data noise. We tested client sampling rates of 0.25, 0.5, and 1.0 (that is, full participation).

The results in Figure 15 demonstrate a clear relationship: a lower client sampling rate increases the vulnerability of FL to data noise. When the sampling rate is 1.0, the relative accuracy is the highest and most stable. As the rate drops to 0.25, the relative accuracy decreases and exhibits more volatility. With a smaller pool of participating clients in each round of training, the model aggregation step is more susceptible to the noisy data. A higher sampling rate provides a more robust aggregated model after averaging local model updates, based on the law of large numbers, where the impact of noisy updates is mitigated.

8 Takeaways and Future Work

8.1 Key Findings

Our study in this paper shows that FL is often significantly more vulnerable to data noise compared to CL. This vulnerability or sensitivity is due to intrinsic characteristics of the FL workflow or training pipeline itself and is therefore ubiquitous regardless of the data modality, ML task, or noise

type. An in-depth analysis reveals a cascading process that amplifies the impact of data noise in both the local and global model updates of FL training: (1) the local model updates performed by clients are misled by data noise to proceed in conflicting directions; (2) these conflicting local model updates are further aggregated by the server to produce a global model update that leads to both slow convergence and high instability.

8.2 Future Directions

Our key findings opened promising avenues for future research to improve the robustness of FL.

Federated Data Cleaning. Although data cleaning has been extensively studied for decades in the database community, its application in FL remains underexplored. The privacy-preserving requirement of FL precludes data cleaning techniques that assume centralized data governance or raw-data exchange. The cascading amplification effect revealed by our study further emphasizes the importance of developing federated data cleaning technologies, which can address data quality problems at their sources before entering the FL training pipeline.

Noise-aware Local Model Update. Given that local model updates can be significantly distracted by data noise, a direction for future exploration is the development of noise-aware local model update mechanisms. For example, an FL client can adopt data filtering algorithms to exclude highly-noisy data samples.

Noise-robust Global Model Aggregation. Moreover, the FL server can adopt noise-robust global model aggregation mechanisms. Our study shows that both the standard FedAvg and the more recent FL algorithms are faced with challenges to learn from noisy clients. If the server could know the data quality issues of the clients in a privacy-preserving way, it could utilize this information when performing global model aggregation. For example, the weight of a client can be determined by how likely and to what extent its parameters are learned from noisy samples. This motivates the design of new client sampling strategies that take into account data quality issues, as we have seen the impact of this step on the FL training pipeline (Section 7.4).

9 Related Work

Data noise in machine learning. Recent work on mitigating data quality issues in federated learning has focused on *label noise*. FPIN [67] presents an incentive mechanism with flexible pricing to sample clients and motivate cleaning. Comm-FedBiO [35] proposes a communication-efficient, learning-based reweighting approach to handling label noise in FL clients. FedELC [23] adopts a two-stage algorithm to detect and correct label noise in FL clients. FedDiv [34] proposes global noise filtering to avoid selecting data with noisy labels from FL clients. Tuor et al. [66] presented a selection algorithm to minimize the effect of label noise by evaluating the relevance of each data sample in each client. RHFL [14] handles label noise with a robust noise-tolerant loss function. Rather than focusing on label noise for specific ML tasks, our work provides a systematic investigation on the impact of *data noise* in training samples, which is more common in practical FL systems deployed on edge devices.

Investigating and mitigating label noise has also been extensively studied in centralized learning [24, 60]. For example, confident learning [45] identifies label noise in datasets by estimating the joint distribution between given noisy labels and unknown labels; approaches based on Shapley value (e.g., Data Shapley [69] and Beta Shapley [32]) detect and remove training samples with noisy labels for more efficient and more effective training.

Improving data quality. Empowering data-intensive applications with high-quality data has been a canonical task in the database community. Extensive techniques have been developed to detect

and address data noise in tabular data [5, 7–12, 22, 30, 48, 52, 59, 68]. This line of work on data cleaning is orthogonal to this study but can inspire new solutions to address data quality issues for FL over tabular/relational data [15, 39]. Besides relational/tabular data, our work additionally investigates the impact of data noise on the accuracy of FL models for other data modalities such as text, audio, image, and video. Developing data cleaning techniques for these modalities beyond relational/tabular data is a promising next step.

Data preparation and cleaning in MLOps systems. The importance of data quality in improving the performance of ML systems has been further underscored by recent research from both the database and the ML communities, coined with the term "data-centric ML/AI" in the literature [70, 74]. In particular, a number of proposals have been devoted to the design and implementation of MLOps systems [29, 42, 50, 73] that include data preparation and cleaning as an important step/component of the overall ML training pipeline [19, 25–27, 36, 37, 44, 54, 55]. To the best of our knowledge, MLOps systems have not yet extended their scope to support FL. As we pointed out in Section 8, such an extension is not trivial, given the decentralized nature of FL in terms of its data distribution. An MLOps system for FL therefore has to take on the challenge of addressing data quality issues in a federated privacy-preserving manner, which is itself an underexplored area that requires further research effort. Recent work in this direction [1] introduces a secure communication protocol that allows FL clients to exchange information about their local data. However, this proposal is not supported in existing FL systems, which only allow communication between FL clients and the server.

10 Conclusion

We present the first large-scale systematic study on the impact of data noise in federated learning (FL), a critical but underexplored challenge for the real-world deployment of FL systems. To facilitate a rigorous and reproducible evaluation, we design `DataNoiseGenerator`, an open-source modular toolkit for the injection of a wide range of controlled noise covering five different data modalities. Our experiments show an unequivocal primary finding: **FL is significantly more vulnerable to data noise than centralized learning (CL)**. We observe this vulnerability consistently across many tested data modalities, noise types, and ML tasks and models. Our in-depth analysis further pinpoints the root causes.

The key takeaway of our work is that data quality is a first-order challenge that cannot be ignored in the design of practical FL systems. Although the data privacy benefits of FL are compelling, they introduce an inherent robustness trade-off in the face of data imperfections. Our findings call for a paradigm shift, urging the community to move beyond idealized assumptions and focus on developing noise-aware mechanisms to ensure that FL models are reliable and trustworthy in the real world. Meanwhile, it opens up new directions for the community to develop privacy-preserving federated data cleaning technologies.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. Benson Chou, Jenny Lin, and Haotian Yang contributed to the early development of the `DataNoiseGenerator` tool. This work was funded in part by Natural Sciences and Engineering Research Council of Canada ALLRP 588144 - 23 and National Science Foundation 2523997, 2534286, and 2315612. Qizhen Zhang is the corresponding author of this research.

References

[1] Aydin Abadi, Vishnu Asutosh Dasu, and Sumanta Sarkar. 2025. Privacy-Preserving Data Deduplication for Enhancing Federated Learning of Language Models. In *32nd Annual Network and Distributed System Security Symposium, NDSS 2025, San Diego, California, USA, February 24-28, 2025*. The Internet Society.

[2] Aman Anand. 2020. AG News Dataset. https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset.

[3] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque De Gusmão, et al. 2020. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390* (2020).

[4] Gopal Bhattrai. 2021. PASCAL VOC 2012 DATASET. https://www.kaggle.com/datasets/gopalbhattrai/pascal-voc-2012-dataset.

[5] Philip Bohannon, Michael Flaster, Wenfei Fan, and Rajeev Rastogi. 2005. A Cost-Based Model and Effective Heuristic for Repairing Constraints by Value Modification. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005*, Fatma Özcan (Ed.). ACM, 143–154. doi:10.1145/1066157.1066175

[6] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. LEAF: A Benchmark for Federated Settings. *CoRR* abs/1812.01097 (2018). arXiv:1812.01097 http://arxiv.org/abs/1812.01097

[7] Anup Chalamalla, Ihab F. Ilyas, Mourad Ouzzani, and Paolo Papotti. 2014. Descriptive and prescriptive data cleaning. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, Curtis E. Dyreson, Feifei Li, and M. Tamer Özsu (Eds.). ACM, 445–456. doi:10.1145/2588555.2610520

[8] Xu Chu and Ihab F. Ilyas. 2016. Qualitative Data Cleaning. *Proc. VLDB Endow.* 9, 13 (2016), 1605–1608. doi:10.14778/3007263.3007320

[9] Xu Chu, Ihab F. Ilyas, Sanjay Krishnan, and Jiannan Wang. 2016. Data Cleaning: Overview and Emerging Challenges. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Fatma Özcan, Georgia Koutrika, and Sam Madden (Eds.). ACM, 2201–2206. doi:10.1145/2882903.2912574

[10] Xu Chu, John Morcos, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Nan Tang, and Yin Ye. 2015. KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, Timos K. Sellis, Susan B. Davidson, and Zachary G. Ives (Eds.). ACM, 1247–1261. doi:10.1145/2723372.2749431

[11] Michele Dallachiesa, Amr Ebaid, Ahmed Eldawy, Ahmed K. Elmagarmid, Ihab F. Ilyas, Mourad Ouzzani, and Nan Tang. 2013. NADEEF: a commodity data cleaning system. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, Kenneth A. Ross, Divesh Srivastava, and Dimitris Papadias (Eds.). ACM, 541–552. doi:10.1145/2463676.2465327

[12] Mohamad Dolatshah, Mathew Teoh, Jiannan Wang, and Jian Pei. 2018. Cleaning Crowdsourced Labels Using Oracles For Statistical Classification. *Proc. VLDB Endow.* 12, 4 (2018), 376–389. doi:10.14778/3297753.3297758

[13] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*. 1605–1622.

[14] Xiuwen Fang and Mang Ye. 2022. Robust federated learning with noisy and heterogeneous clients. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10072–10081.

[15] Xiuwen Fang, Mang Ye, and Xiyuan Yang. 2023. Robust Heterogeneous Federated Learning under Data Corruption. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 4997–5007. doi:10.1109/ICCV51070.2023.00463

[16] Yuan Gong, Yu-An Chung, and James Glass. 2021. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778* (2021).

[17] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in neural information processing systems* 34 (2021), 18932–18943.

[18] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*. 5842–5850.

[19] Stefan Grafberger, Paul Groth, and Sebastian Schelter. 2023. Automating and Optimizing Data-Centric What-If Analyses on Native Machine Learning Pipelines. *Proc. ACM Manag. Data* 1, 2 (2023), 128:1–128:26.

[20] Dongyoon Han, Jiwhan Kim, and Junmo Kim. 2017. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5927–5935.

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[22] Yuchuan Huang and Mohamed F. Mokbel. 2024. Sparcle: Boosting the Accuracy of Data Cleaning Systems through Spatial Awareness. *Proc. VLDB Endow.* 17, 9 (2024), 2349–2362. doi:10.14778/3665844.3665862

[23] Xuefeng Jiang, Sheng Sun, Jia Li, Jingjing Xue, Runhan Li, Zhiyuan Wu, Gang Xu, Yuwei Wang, and Min Liu. 2024. Tackling Noisy Clients in Federated Learning with End-to-end Label Correction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, Edoardo Serra and Francesca Spezzano (Eds.). ACM, 1015–1026. doi:10.1145/3627673.3679550

[24] Davood Karimi, Haoran Dou, Simon K. Warfield, and Ali Gholipour. 2020. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Anal.* 65 (2020), 101759. doi:10.1016/J.MEDIA.2020.101759

[25] Bojan Karlas, David Dao, Matteo Interlandi, Sebastian Schelter, Wentao Wu, and Ce Zhang. 2024. Data Debugging with Shapley Importance over Machine Learning Pipelines. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

[26] Bojan Karlas, Peng Li, Renzhi Wu, Nezihe Merve Gürel, Xu Chu, Wentao Wu, and Ce Zhang. 2020. Nearest Neighbor Classifiers over Incomplete Information: From Certain Answers to Certain Predictions. *Proc. VLDB Endow.* 14, 3 (2020), 255–267.

[27] Bojan Karlas, Babak Salimi, and Sebastian Schelter. 2025. Navigating Data Errors in Machine Learning Pipelines: Identify, Debug, and Learn. In *Companion of the 2025 International Conference on Management of Data, SIGMOD/PODS 2025, Berlin, Germany, June 22-27, 2025*. ACM, 813–820.

[28] Fereshte Khani and Percy Liang. 2020. Feature noise induces loss discrepancy across groups. In *International conference on machine learning*. PMLR, 5209–5219.

[29] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. 2023. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access* 11 (2023), 31866–31879.

[30] Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J. Franklin, and Ken Goldberg. 2016. ActiveClean: Interactive Data Cleaning For Statistical Modeling. *Proc. VLDB Endow.* 9, 12 (2016), 948–959. doi:10.14778/2994509.2994514

[31] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[32] Yongchan Kwon and James Zou. 2022. Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 151)*, Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). PMLR, 8780–8802. https://proceedings.mlr.press/v151/kwon22a.html

[33] Jure Leskovec. 2026. Web data: Amazon reviews. https://snap.stanford.edu/data/web-Amazon-links.html.

[34] Jichang Li, Guanbin Li, Hui Cheng, Zicheng Liao, and Yizhou Yu. 2024. FedDiv: Collaborative noise filtering for federated learning with noisy labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 3118–3126.

[35] Junyi Li, Jian Pei, and Heng Huang. 2022. Communication-Efficient Robust Federated Learning with Noisy Labels. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 914–924. doi:10.1145/3534678.3539328

[36] Peng Li, Zhiyi Chen, Xu Chu, and Kexin Rong. 2023. DiffPrep: Differentiable Data Preprocessing Pipeline Search for Learning over Tabular Data. *Proc. ACM Manag. Data* 1, 2 (2023), 183:1–183:26.

[37] Peng Li, Xi Rao, Jennifer Blase, Yue Zhang, Xu Chu, and Ce Zhang. 2021. Cleanml: A study for evaluating the impact of data cleaning on ml classification tasks. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 13–24.

[38] LIAMLARSEN. 2017. Shakespeare plays. https://www.kaggle.com/datasets/kingburrito666/shakespeare-plays.

[39] Lichuan Ma, Qingqi Pei, Lu Zhou, Haojin Zhu, Licheng Wang, and Yusheng Ji. 2021. Federated Data Cleaning: Collaborative and Privacy-Preserving Data Cleaning for Edge Intelligence. *IEEE Internet Things J.* 8, 8 (2021), 6757–6770. doi:10.1109/JIOT.2020.3027980

[40] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. 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 (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Xiaojin (Jerry) Zhu (Eds.). PMLR, 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html

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

[42] Leonel Aguilar Melgar, David Dao, Shaoduo Gan, Nezihe Merve Gürel, Nora Hollenstein, Jiawei Jiang, Bojan Karlas, Thomas Lemmin, Tian Li, Yang Li, Susie Xi Rao, Johannes Rausch, Cédric Renggli, Luka Rimanic, Maurice Weber, Shuai Zhang, Zhikuan Zhao, Kevin Schawinski, Wentao Wu, and Ce Zhang. 2021. Ease.ML: A Lifecycle Management System for Machine Learning. In *11th Conference on Innovative Data Systems Research, CIDR 2021, Virtual Event, January 11-15,*

2021, Online Proceedings.
[43] Rami Mohammad, Fadi Thabtah, and TL McCluskey. 2015. Phishing websites dataset. (2015).
[44] Felix Neutatz, Binger Chen, Ziawasch Abedjan, and Eugene Wu. 2021. From Cleaning before ML to Cleaning for ML. *IEEE Data Eng. Bull.* 44, 1 (2021), 24–41.
[45] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2021. Confident Learning: Estimating Uncertainty in Dataset Labels. *J. Artif. Intell. Res.* 70 (2021), 1373–1411. doi:10.1613/JAIR.1.12125
[46] Cam Nugent. 2018. California Housing Prices. https://www.kaggle.com/datasets/camnugent/california-housing-prices.
[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
[48] Jinfeng Peng, Derong Shen, Nan Tang, Tieying Liu, Yue Kou, Tiezheng Nie, Hang Cui, and Ge Yu. 2022. Self-supervised and Interpretable Data Cleaning with Sequence Generative Adversarial Networks. *Proc. VLDB Endow.* 16, 3 (2022), 433–446. doi:10.14778/3570690.3570694
[49] Karol J. Piczak. 2015. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia* (Brisbane, Australia, 2015-10-13). ACM Press, 1015–1018. doi:10.1145/2733373.2806390
[50] Fotis Psallidas, Yiwen Zhu, Bojan Karlas, Jordan Henkel, Matteo Interlandi, Subru Krishnan, Brian Kroth, K. Venkatesh Emani, Wentao Wu, Ce Zhang, Markus Weimer, Avrilia Floratou, Carlo Curino, and Konstantinos Karanasos. 2022. Data Science Through the Looking Glass: Analysis of Millions of GitHub Notebooks and ML.NET Pipelines. *SIGMOD Rec.* 51, 2 (2022), 30–37.
[51] PyTorch. 2026. torchvision. https://docs.pytorch.org/vision/stable/index.html.
[52] Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. 2017. HoloClean: Holistic Data Repairs with Probabilistic Inference. *Proc. VLDB Endow.* 10, 11 (2017), 1190–1201. doi:10.14778/3137628.3137631
[53] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 91–99. https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html
[54] Cédric Renggli, Luka Rimanic, Nezihe Merve Gürel, Bojan Karlas, Wentao Wu, and Ce Zhang. 2021. A Data Quality-Driven View of MLOps. *IEEE Data Eng. Bull.* 44, 1 (2021), 11–23.
[55] Cédric Renggli, Luka Rimanic, Luka Kolar, Wentao Wu, and Ce Zhang. 2023. Automatic Feasibility Study via Data Quality Analysis for ML: A Case-Study on Label Noise. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*. IEEE, 218–231.
[56] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. 2014. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*. 1041–1044.
[57] scikit learn. 2026. MLPRegressor. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html.
[58] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.
[59] Shafaq Siddiqi, Roman Kern, and Matthias Boehm. 2023. SAGA: A Scalable Framework for Optimizing Data Cleaning Pipelines for Machine Learning Applications. *Proc. ACM Manag. Data* 1, 3 (2023), 218:1–218:26. doi:10.1145/3617338
[60] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2023. Learning From Noisy Labels With Deep Neural Networks: A Survey. *IEEE Trans. Neural Networks Learn. Syst.* 34, 11 (2023), 8135–8153. doi:10.1109/TNNLS.2022.3152527
[61] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).
[62] Jiahao Tan and Xinpeng Wang. [n. d.]. *FL-bench: A federated learning benchmark for solving image classification tasks*. https://github.com/KarhouTam/FL-bench
[63] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html* 3, 6 (2023), 7.
[64] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems* 35 (2022), 10078–10093.
[65] Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K. Leung. 2020. Overcoming Noisy and Irrelevant Data in Federated Learning. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*. IEEE, 5020–5027. doi:10.1109/ICPR48806.2021.9412599
[66] Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K Leung. 2021. Overcoming noisy and irrelevant data in federated learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 5020–5027.

[67] Hengzhi Wang, Haoran Chen, Minghe Ma, and Laizhong Cui. 2025. Dealing with Noisy Data in Federated Learning: An Incentive Mechanism with Flexible Pricing. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (Eds.). ACM, 4432–4441. doi:10.1145/3696410.3714961

[68] Jiannan Wang, Sanjay Krishnan, Michael J. Franklin, Ken Goldberg, Tim Kraska, and Tova Milo. 2014. A sample-and-clean framework for fast and accurate query processing on dirty data. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, Curtis E. Dyreson, Feifei Li, and M. Tamer Özsu (Eds.). ACM, 469–480. doi:10.1145/2588555.2610505

[69] Jiachen T. Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. 2025. Data Shapley in One Training Run. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. https://openreview.net/forum?id=HD6bWcj87Y

[70] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2023. Data collection and quality challenges in deep learning: a data-centric AI perspective. *VLDB J.* 32, 4 (2023), 791–813.

[71] Yuecong Xu, Jianfei Yang, Haozhi Cao, Kezhi Mao, Jianxiong Yin, and Simon See. 2020. ARID: A New Dataset for Recognizing Action in the Dark. *CoRR* abs/2006.03876 (2020). arXiv:2006.03876 https://arxiv.org/abs/2006.03876

[72] Haotian Yang, Zhuoran Wang, Benson Chou, Sophie Xu, Hao Wang, Jingxian Wang, and Qizhen Zhang. 2025. An Empirical Study of the Impact of Federated Learning on Machine Learning Model Accuracy. *CoRR* abs/2503.20768 (2025). arXiv:2503.20768 doi:10.48550/ARXIV.2503.20768

[73] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Fen Xie, and Corey Zumar. 2018. Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.* 41, 4 (2018), 39–45.

[74] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Ben Hu. 2025. Data-centric Artificial Intelligence: A Survey. *ACM Comput. Surv.* 57, 5 (2025), 129:1–129:42.

Received October 2025; revised January 2026; accepted February 2026