MISF: MLLM Guided Iterative Sample Filtering for Data Fault Detection

Guoying Chen, Ruizhuo Zhao*, Zhewei Xu, Bo Yang, Kunlong Wang

Beijing Institute of Computer Technology and Application, Beijing, China
gychen12@126.com, ruizhuozhao@163.com, xzhewei@gmail.com, yangbo9415@qq.com, 15201122897@126.com

Abstract

High quality datasets are critical for training reliable machine learning models, yet data faults caused by insufficient annotation expertise or malicious poisoning attacks remain prevalent. Traditional classifier based methods rely on manually curated subsets for fault detection, but their limited scale frequently leads to model overfitting. While multimodal large language models (MLLMs) based methods offer promising detection capabilities, their few-shot learning limitations hinder generalization in domain specific tasks. To address these challenges, we propose MLLM Guided Iterative Sample Filtering (MISF), a novel framework that combines the strengths of MLLM based initialization and iterative data refinement. Our framework initializes the detection model with MLLM generated synthetic images and a curated clean subset, then iteratively refines it by progressively selecting high certainty clean samples, improving both domain adaptation and detection accuracy. Extensive experiments on RESISC45 and Oxford-IIIT Pets datasets demonstrate that MISF effectively identifies data faults, outperforming existing approaches. MISF provides a robust, scalable solution for improving dataset quality in specialized domains.

Code — https://github.com/ying-cgy/MISF.git

1 Introduction

Recent breakthroughs in deep learning have dramatically advanced artificial intelligence capabilities. These achievement, however, fundamentally depend on high quality training data, which is frequently compromised by pervasive data faults. Such faults include label errors from unqualified annotators and malicious backdoor attacks that poison datasets. In safety-critical domains like autonomous driving, poisoned training samples containing backdoor triggers can induce catastrophic misbehavior during inference, directly endangering human lives (Liu et al. 2018). These challenges underscore the urgent need for effective methods to detect data faults.

Traditional classifier based method typically rely on manually curated subsets to identify data faults. Cola (Lam et al. 2025) iteratively expands a clean subset by training on KNN-selected samples and updating the model until no new clean samples can be found. BHN (Yu, Ma, and Liu 2023a) trains new models exclusively on clean data to estimate each sample mislabeling probabilities. While these methods have shown promise, they often suffer from overfitting due to the limited size of clean subsets. To address this limitation, recent studies have explored leveraging multimodal large language models (MLLMs) for data fault detection. Nahum et al. (Nahum et al. 2024) utilize multiple LLMs to jointly identify and flag potentially mislabeled samples without requiring additional detection models. However, MLLMs still face challenges in generalizing to domain specific datasets, where their few-shot classification performance tends to degrade significantly.

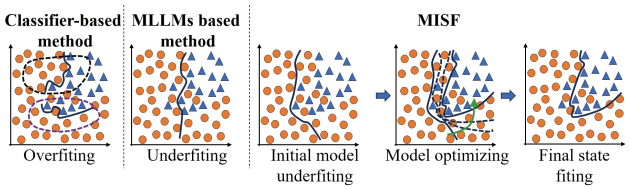


Figure 1: Comparison of MISF with classifier based methods and MLLMs based methods. Orange circles represent clean samples, while blue triangles denote fault samples. Circles denote the subset samples selected by classifier based methods.

We introduce MISF, an MLLM guided iterative framework for data fault detection. By initializing models using MLLM generated data combined with a curated subset and refining through self-optimizing sample selection, MISF achieves precise distributional alignment with target samples, significantly enhancing generalization. Figure 1 shows the differences among classifier based methods, MLLM based methods and MISF. Classifier based methods tend to overfit the small curated subsets, limiting their generalization. In contrast, MLLM based methods often underfit when faced with domain specific tasks. MISF, while achieving solid initial performance through the integration of generated data and curated subsets, shows substantial improvement through iterative training. As the iterations proceed, the model becomes increasingly aligned with the true data distribution. MISF leverages template-generated textual

*Corrsponding author.

prompts to guide MLLM in producing training data, which is subsequently used to train the initial detection model. The framework then iteratively refines the model through autonomous clean sample selection based on prediction-label consistency and low Gini impurity. Iterations terminate automatically when samples exhibiting increasing uncertainty exceed a threshold. This self-optimizing mechanism enhances model performance, achieving domain adaptive fault detection.

The main contribution of this paper is as follows:

- We propose MISF, an MLLM guided iterative framework for data fault detection in image datasets. By leveraging prompt templates, MISF generates diverse images and combines them with a curated subset to train an initial model, then iteratively selects clean samples to refine the model and detect data faults.
- We introduce a mechanism combining Gini impurity and prediction-label consistency for reliable clean sample identification. Clean samples are iteratively selected to refine the model, and the process terminates automatically when the proportion of clean samples with increased Gini impurity exceeds a predefined threshold.
- Experimental results on RESISC45 and Oxford-IIIT Pets datasets validate the effectiveness of MISF in detecting fault samples, with strong performance reflected by favorable TPR and FPR metrics.

2 Related Work

2.1 Large Language Model

In recent years, large language models (LLMs), such as such as ChatGPT (OpenAI 2024) and deepseek (DeepSeek-AI 2025), have remarkable advancements in natural language processing (NLP). Trained through learning on massive text corpora, these models are capable of generating coherent and logically consistent text and excel at tasks such as question answering, instruction execution, and natural language generation.

Recent advances in text-to-image generation have demonstrated the remarkable capabilities of models in producing high-quality and semantically coherent images. Stable Diffusion v1-4 (Rombach et al. 2022) represents a state-of-the-art latent diffusion model that generates images by performing denoising in a compressed latent space.

More recently, GILL (Koh, Fried, and Salakhutdinov 2023) introduces a unified generative framework. By jointly modeling visual and textual modalities, GILL supports not only text-to-image synthesis but also image captioning and vision-language reasoning, demonstrating the potential of MLLM in generative tasks.

2.2 Label Noise Detection

Extensive efforts have been devoted to identifying labeling noise in datasets. A common strategy is to leverage sample relationships through clustering. SimiFeat (Zhu, Dong, and Liu 2022) utilizes feature similarities among samples, where SimiFeat applies local voting based on label consistency among neighbors or scores samples to detect potential label errors. Confidence learning (Northcutt, Jiang, and Chuang 2021) estimates the joint distribution of noisy and true labels to identify mislabeled samples.

Another line of work relies on training auxiliary models. BHN (Yu, Ma, and Liu 2023a) first trains a model using clean data and then applies statistical hypothesis testing with the Benjamini-Hochberg procedure to identify mislabeled samples while controlling false discovery rates. Cola (Lam et al. 2025) introduces a two-stage verification strategy comprising local verification and global verification, using local verification to select a clean subset for training and then performing global verification to detect noisy samples.

Some approaches analyze model dynamics during training. O2U-Net (Huang et al. 2019) ranks instances by average loss, alternating between underfitting and overfitting to reveal unstable samples. TracIn (Pruthi et al. 2020) estimates the influence of each training instance by tracing its impact on the loss function over time.

2.3 Data Poison Detection

Several approaches detect poisoned data by analyzing model behavior or sample characteristics. Neural Cleanse (Wang et al. 2019) detects backdoor samples by comparing the effort needed to misclassify poisoned versus clean samples, as poisoned samples require less perturbation to change predictions. Gao et al. (Gao et al. 2019) found that clean samples' predictions vary significantly under input disturbances, while poisoned samples remain stable due to embedded triggers.

Other methods focus on sample similarity or feature relationships. Deep k-NN (Peri et al. 2020) leverages feature clustering to identify poisoned samples by voting among nearest neighbors. Fabio et al. (De Gaspari, Hitaj, and Mancini 2024) propose using eigenvector analysis to compare data points with class centers, flagging samples whose eigenvector-based assignment conflicts with their labels.

Some approaches rely on global statistical properties of datasets. Tran et al. (Tran, Li, and Madry 2018) introduced spectral feature analysis to identify and remove poisoned samples by outlier detection in feature space. Chen et al. (Chen et al. 2018) proposed a backdoor detection method without relying on trusted data; by clustering neuron activations and using metrics like the silhouette coefficient, poisoned samples are effectively separated from clean ones.

Existing classifier based methods rely on a curated clean subset, making them prone to overfitting. Meanwhile, MLLM based methods often struggle with domain specific datasets and fail to generalize. To address these limitations, we propose MISF, a framework for detecting both label noise and data poison.

3 Method

In this section, we propose MISF, a MLLM guided Iterative Sample Filtering framework detecting data faults in image datasets. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. $\mathcal{X}$ denotes the input space and $\mathcal{Y}$ denote the label space consisting of N classes. Dataset $\mathcal{D}$ may consist of both clean data and fault data. Our goal is to identify the fault data in the dataset. We focus on two types of data faults: label noise and data poisoning.

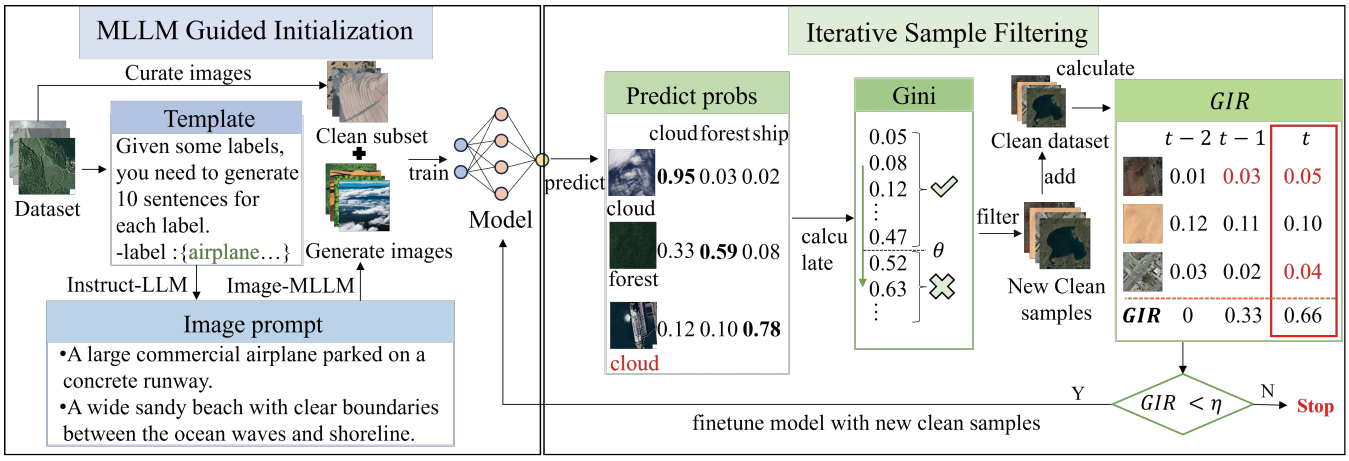


Figure 2: The framework of MISF

- Label noise refers to samples where images is clean while the associated label is incorrect, formally denoted as: $(x_i, y_i) \mapsto (x_i, y_t)$ where $y_t \neq y_i$
- Data poisoning refers to samples where the image is embedded with a backdoor trigger while its label is intentionally changed to a target class, formally expressed as: $(x_i, y_i) \mapsto (x_i \oplus \delta, y_t)$ for trigger δ and target y_t.

As shown in Figure 2, MISF begins with MLLM guided initialization, where prompts derived from templates steer MLLMs to generate synthetic training images $\tilde{\mathcal{D}} = (\tilde{x}_j, \tilde{y}_j)$, which are combined with a curated subset to train an initial detector. After initialization, MISF iteratively refines the model by selecting clean samples. Through iterative selection of samples exhibiting both prediction-label consistency and low Gini impurity ($Gini < \theta$), the model progressively aligns with the target distribution. The iteration proceeds until the proportion of clean samples exhibiting increased Gini impurity surpasses a predefined threshold. Finally, MISF yields a fitted model alongside a reliable partition of clean and fault samples.

3.1 MLLM Guided Initialization

MISF initializes the fault detection model by combining synthesizing training images via MLLMs with a curated subset of clean samples. Let $\mathcal{Y} = \{y_1, y_2, ..., y_k\}$ denote the label set. A naive approach is to manually write some prompts for each label y_k, but this becomes complex when K is large. Moreover, using only the label name as a prompt (e.g., "a photo of y_k") tends to produce overly diverse or stylistically inconsistent images.

To generate high-quality and semantically consistent training data, MISF proposes an initialization mechanism guided by MLLMs. Specifically, we construct a prompt template, which is modified based on the label set of the target dataset and optionally append specific dataset visual cues.

The template is then passed to a instruct-LLM, which generates a variety of natural language descriptions. Let $\mathcal{T} = \{t_k\}_{k=1}^{K}$ denote the set of generated text prompts corresponding to K classes. Each t_k is a syntactically diverse but semantically aligned description of class k.

Next, the generated text prompts are fed into a image-MLLM to generate images. For each prompt t_k, we obtain a set of label-consistent synthetic images. The final synthetic dataset is denoted as Equation 1:

$$\tilde{\mathcal{D}} = \left\{ \left(\tilde{x}_k^{(j)}, y_k \right) \,\middle|\, 1 \leq k \leq K,\ 1 \leq j \leq N_k \right\} \quad (1)$$

To initialize the model, we select a small proportion of clean samples $\mathcal{D}'$ and combine them with the synthetic dataset $\tilde{\mathcal{D}}$ for training. We use the hybrid dataset to train an initial model $\mathcal{M}_0$, which serves as a detector in subsequent sample filtering. By leveraging both clean and synthetic samples, the model achieves an effective initialization.

3.2 Iterative Sample Filtering

MISF then iteratively selects new clean samples and continuously refines the initial model to make the model more fitting and separate the clean set and the fault set. In each iteration, the model is further refined on a set of high confidence clean sets, enhancing its ability to separate clean and fault data. Specifically, We leverage Gini impurity (Quinlan 1986) and prediction-label consistency to select clean samples.

The Gini impurity quantifies the uncertainty of the model's predictions, based on the predicted probability distribution of the given sample. Given a sample x_i, let its predicted probability vector be $P_i = \{p_{i,1}, p_{i,2}, ..., p_{i,k}\}$, where $p_{i,k}$ denoted the probability that sample x_i belongs to class k. The Gini impurity is defined as Equation 2.

$$\text{Gini}(x_i) = 1 - \sum_{k=1}^{K} p_{i,k}^2 \quad (2)$$

A lower Gini impurity indicates higher confidence. By selecting samples with low Gini values and consistent with the prediction and label, we ensure that only clean samples are added to the clean set in each iteration.

Algorithm 1 shows MISF's iterative sample filtering. For the samples (x_i, y_i) in the dataset $\mathcal{D}$, the initial model M_0

predict both the predicted labels $\hat{y}_i$ and probability P_i. Then calculate the Gini(x_i) to evaluate the uncertainty. Clean samples are add to the initial clean sample set $\mathcal{D}'_0$, which is defined as Equation 3.

$$\mathcal{D}'_0 = \{(x_i, y_i) \in \mathcal{D} | \hat{y}_i = y_i \wedge Gini(x_i) < \theta\} \quad (3)$$

After obtaining the initial clean dataset $\mathcal{D}'_0$, we initiate an iterative process to refine the clean dataset. At iteration t, we refine a model $\mathcal{M}_t$ on the clean dataset $\mathcal{D}'_{t-1}$ and use it to predict the entire dataset $\mathcal{D}$. For each sample $x_i \in \mathcal{D}$, we obtain its predicted label $\hat{y}_i^t$ and corresponding Gini impurity $Gini_t(x_i)$. We then sort the samples according to the Gini impurity and construct the update clean dataset $\mathcal{D}'_t$ by adding new clean samples, which are correctly predicted by the $\mathcal{M}_t$, have Gini impurity below the threshold θ and have not been included in the previous clean set $\mathcal{D}'$, which is defined as Equation 4. These new clean samples are used to retrain the model in next iteration. As the quality of the training data improves, the model becomes more accurate, enabling better identification of clean samples in subsequent iterations.

$$\mathcal{D}'_t = \left\{(x_i, y_i) \in \mathcal{D} \setminus \mathcal{D}' \,\middle|\, \hat{y}_i^t = y_i^t \wedge \mathrm{Gini}_t(x_i) < \theta\right\} \quad (4)$$

Based on these predictions, we construct a new clean dataset $\mathcal{D}'$ by selecting samples. A naive stopping criterion is to stop the iteration when no new clean samples are selected. However, this method can be time-intensive and prone to introduce fault samples in the iteration. To address this, we propose a dynamic stopping mechanism that monitors the quality of previously selected clean samples. We define the Gini Increase Rate (GIR) as the proportion of clean samples whose Gini impurity increase from iteration $t-1$ to t, indicating a potential rise in predictive uncertainty, computed as Equation 5.

$$GIR = \frac{num\big(\{x \in \mathcal{D}' \setminus \mathcal{D}'_t \mid Gini_t(x) > Gini_{t-1}(x)\}\big)}{|\mathcal{D}' \setminus \mathcal{D}'_t|} \quad (5)$$

$\mathcal{D}' \setminus \mathcal{D}'_t$ refers to the set of clean samples identified prior to the current iteration. $|\cdot|$ represents the total number of elements it contains. $Gini_t(x)$ and $Gini_{t-1}(x)$ represent the Gini values of sample x at iterations t and $t-1$. The operator $num(\cdot)$ returns the number of elements satisfying the specified condition.

If the ratio GIR exceeds the threshold η, we terminate the iteration, assuming that the model has converged and further refinement offers diminishing returns. The final clean dataset is denoted as $\mathcal{D}'$ and the remaining samples $\mathcal{D} \setminus \mathcal{D}'$ are treated as fault dataset.

Algorithm 1: Gini-guide select

Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, Initial model $\mathcal{M}_0$, Gini threshold θ, ending threshold η
Output: Clean set $\mathcal{D}'$, fault set $\mathcal{D}^*$

Initial clean set $\mathcal{D}'$ and fault set $\mathcal{D}^*$.
$t = 0$
while True **do**
  for $(x_i, y_i) \in \mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ **do**
    $(\hat{y}_{i,t}, P_{i,t}) = f(\mathcal{M}_{t-1}, x_i)$
    $\mathrm{Gini}_t(x_i) = 1 - \sum_{k=1}^K p_{i,k,t}^2, p_{i,k,t} \in P_{i,t}$
    if $\hat{y}_i = y_i$ and $\mathrm{Gini}_t(x_i) < \theta$ and (x_i, y_i) not in $\mathcal{D}'$ **then**
      $\mathcal{D}'_t$.add (x_i, y_i)
    end if
  end for
  $\mathcal{D}'$.add$(\mathcal{D}'_t)$
  Calculate GIR
  if $GIR > \eta$ **then**
    break
  end if
  $\mathcal{M}_t = train(M_{t-1}, \mathcal{D}'_t)$
end while
$\mathcal{D}^* = \mathcal{D} \setminus \mathcal{D}'$
return $\mathcal{D}', \mathcal{D}^*$

4 Experiment

In this section, we introduce our experiment setting and results.

4.1 Dataset

We evaluate our method on two dataset, including RESISC45 (Cheng, Han, and Lu 2017) and Oxford-IIIT Pets (oxford) (Parkhi et al. 2012) for image classification. RESISC45 is a remote sensing imagery dataset comprising 45 scene categories, exhibiting significant geographical diversity. Oxford-IIIT Pets is a fine-grained recognition benchmark containing 37 pet species with challenging variations in scale, pose, and illumination.

To comprehensively evaluate the robustness of our method, we simulate both symmetric noise (Van Rooyen, Menon, and Williamson 2015) and asymmetric noise (Zhu et al. 2022) on the two datasets. Additionally, we consider four data poisoning attacks: BadNets (Gu et al. 2019), Blended (Chen et al. 2017), SIG (Barni, Kallas, and Tondi 2019), and ISSBA (Li et al. 2021). These attack settings cover a broad spectrum of real-world scenarios, including both label corruption and feature-level backdoor injection.

4.2 Baseline

For label noise detection, we adopt confidence learning (CL) (Northcutt, Jiang, and Chuang 2021), SimiFeat (Zhu, Dong, and Liu 2022) and Cola (Lam et al. 2025) as baselines. Regarding data poison detection, we employ the spectral signature (SS) method (Tran, Li, and Madry 2018). Additionally, we leverage a multimodal large language model (MLLM) for zero-shot classification to detect both label noise and data poisoning and a sample is flagged as a fault sample when the MLLM's prediction different from the given label.

- **CL:** Estimate the joint distribution to detect fault. We perform 5-fold cross-validation and train a Resnet50 (He et al. 2016).
- **SimiFeat:** Utilize feature similarities among samples to detect fault. We use Resnet50 (He et al. 2016), configuring k-NN with k=10 and training for 21 epochs according to the paper.

	RESISC45				Oxford				Average			
	Symmetric		Asymmetric		Symmetric		Asymmetric		Symmetric		Asymmetric	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
CL	0.603	0.618	0.769	0.763	0.549	0.543	0.589	0.614	0.576	0.581	0.593	0.591
Simifeat	0.989	0.282	0.940	0.276	0.993	0.251	0.937	0.242	0.991	0.267	0.939	0.259
Cola	0.961	**0.012**	0.902	**0.017**	0.931	**0.038**	0.725	**0.041**	0.945	**0.015**	0.828	**0.034**
MLLM based	0.989	0.501	0.948	0.500	0.985	0.498	0.945	0.502	0.987	0.495	0.947	0.501
MISF	**0.992**	0.071	**0.981**	0.066	**0.994**	0.068	**0.988**	0.062	**0.993**	0.069	**0.985**	0.064

Table 1: Comparison of noisy label detection under 20% noise level

	RESISC45				Oxford				Average			
	Symmetric		Asymmetric		Symmetric		Asymmetric		Symmetric		Asymmetric	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
CL	0.782	0.785	0.668	0.678	0.672	0.675	0.591	0.571	0.727	0.73	0.680	0.688
Simifeat	0.980	0.405	0.585	0.343	0.988	0.431	0.603	0.325	0.984	0.441	0.594	0.334
Cola	0.961	**0.014**	0.548	**0.035**	0.956	0.093	0.580	0.174	0.946	**0.021**	0.582	0.088
MLLM based	0.991	0.502	0.944	0.500	0.983	0.498	0.941	0.500	0.983	0.500	0.943	0.500
MISF	**0.997**	0.040	**0.984**	0.045	**0.996**	**0.046**	**0.981**	**0.050**	**0.997**	0.043	**0.983**	**0.048**

Table 2: Comparison of noisy label detection under 40% noise level

- **Cola:** Use local verification and global verification to detect fault. We follow the settings described in the paper.
- **SS**: Use spectral feature analysis to detect fault. We implement it via the ART toolkit (Nicolae et al. 2018).
- **MLLM based method**: Use BLIP implemented within the LAVIS library (Li et al. 2023) as MLLM.

4.3 Evaluation Metrics

We adopt True Positive Rate (TPR) and False Positive Rate (FPR) as the primary evaluation metrics. TPR evaluates the ability of a method to correctly detect fault samples, while FPR quantifies the proportion of clean samples that are mistakenly identified as fault samples. An ideal detection method should achieve a high TPR while maintaining a low FPR.

In addition, we use classification accuracy (ACC) to assess the model's performance after fault samples are removed. Specifically, we remove the detected fault samples to construct a purified dataset, train a model using purified dataset, and evaluate its accuracy. ACC reflects the effectiveness of the data fault detection method in preserving purified data quality.

4.4 Experiment Setting

We train a ResNet50 model to distinguish between clean and fault samples. We set the prediction threshold at 0.5, which used to determine whether a sample is clean, and set the stopping criterion as 0.5. Considering ChatGPT's excellent text processing capabilities, We employ ChatGPT (OpenAI 2024) to automatically generate textual prompts for image synthesis. For each class, we generate 10 diverse prompts, which are then used as inputs to the Stable Diffusion v1-4 model (Rombach et al. 2022). Stable Diffusion v1-4 is an text-to-image model capable of producing high-quality, photorealistic images from natural language descriptions. We generate 50 images per prompt, yielding 500 images for each class.

4.5 Experiment Result

Evaluation on Label Noise Detection We evaluate our method on label noise detection, where 20% of samples are randomly corrupted. Table 1 summarizes the results of MISF and the baselines. MLLM based method achieves a high TPR across all datasets. However, it also results in a high FPR, suggesting that many clean samples are incorrectly flagged as fault samples. This is partly due to the limited classification accuracy of MLLM based method. SimiFeat achieves a relatively lower FPR and performs better on the Oxford dataset. Nevertheless, its FPR significantly increases when evaluated on different datasets. Cola achieves favorable performance under symmetric noise. However, its detection performance decline when the noise changes to an asymmetric noise. MISF consistently shows strong detection performance across all datasets and noise types, confirming its effectiveness.

We further increase the label noise ratio to 40% to evaluate the robustness of MISF. As shown in Table 2, Cola and SimiFeat performs well under symmetric noise, with results similar to those at a 20%. However, its performance degrades significantly under asymmetric noise, indicating its sensitivity to label distribution. Although MLLM based method has a relatively high TPR, it also results in a substantial number of false positives. MISF achieve marginal improvements in both TPR and FPR. Even as the noise ratio increases, the detection results of MISF remain stable, demonstrating its strong generalizability and adaptability to complex noise conditions.

Dataset	Method	Badnets		Blended		SIG		ISSBA		Average	
		TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
RESISC45	SS	0.409	0.503	0.438	0.500	0.425	0.501	0.417	0.502	0.423	0.501
	MLLM based	**0.998**	0.500	0.996	0.500	**0.998**	0.499	0.997	0.499	**0.997**	0.499
	MISF	**0.998**	**0.033**	**0.997**	**0.037**	0.995	**0.040**	**0.998**	**0.036**	**0.997**	**0.036**
Oxford	SS	0.589	0.484	0.462	0.497	0.490	0.490	0.481	0.495	0.505	0.491
	MLLM based	0.983	0.501	0.976	0.501	0.970	0.500	0.974	0.500	0.975	0.501
	MISF	**0.998**	**0.04**	**0.998**	**0.049**	**0.993**	**0.047**	**0.994**	**0.041**	**0.995**	**0.044**

Table 3: Comparison of poison data detection under 10% poison ratio

Dataset	Method	Badnets		Blended		SIG		ISSBA		Average	
		TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
RESISC45	SS	0.204	0.497	0.261	0.497	0.352	0.496	0.246	0.297	0.265	0.446
	MLLM based	0.998	0.499	0.997	0.499	**0.999**	0.499	0.998	0.499	**0.998**	0.499
	MISF	**0.999**	**0.036**	**0.998**	**0.037**	**0.999**	**0.036**	**0.998**	**0.045**	**0.998**	**0.038**
Oxford	SS	0.471	0.495	0.557	0.494	0.494	0.494	0.471	0.495	0.498	0.494
	MLLM based	0.985	0.501	0.985	0.501	0.971	0.500	0.985	0.502	0.981	0.501
	MISF	**1.000**	**0.044**	**0.998**	**0.044**	**0.999**	**0.048**	**0.999**	**0.045**	**0.999**	**0.045**

Table 4: Comparison of poison data detection under 1% poison ratio

Evaluation on Data Poison We evaluate our method on poisoned data detection with a poison ratio set to 0.1 and table 3 presents the experimental results. SS fails to effectively identify poisoned samples, resulting in low TPR on both datasets. While MLLM based method achieves high TPR, it also yields a considerably high FPR due to insufficient precision. MISF demonstrates strong performance against all poisoning attacks, which achieves an average TPR of 0.997 and FPR of 0.036 on the RESISC45, and an average TPR of 0.955 and FPR of 0.044 on the Oxford , indicating that MISF can accurately detect nearly all poisoned samples.

We further evaluate our method under a lower poison ratio of 0.01. Table 4 presents the results on the RESISC45 and Oxford datasets. Our method maintains strong detection performance, achieving an average TPR of 0.998 on RESISC45 and 0.999 on Oxford. The performance of the MLLM based method remains unchanged, while the detection performance of the SS method degrades on the RESISC45 dataset.

Evaluation on Purified Data After detecting fault samples, We evaluate the effectiveness of data fault detection by training a model on the purified dataset after excluding the detected fault samples. We conduct experiments on datasets with a noise ratio of 0.2, training ResNet50 on the cleaned subsets and measuring accuracy (ACC). Additionally, we evaluate performance on the mixed datasets that contain noise. Table 5 presents the detection performance on the cleaned datasets after filtering out fault samples under a noise ratio of 0.2. MISF achieves the best results in nearly all experiments, demonstrating its effectiveness in identifying and removing noisy labels.

	RESISC45		Oxford	
	Sym	Asym	Sym	Asym
Mixed Dataset	0.863	0.7971	0.8898	0.8059
CL	0.8549	0.8269	0.8993	0.8114
Simifeat	0.9204	0.9158	0.8763	0.7938
Cola	**0.9338**	0.9147	0.9064	0.8891
MLLM based	0.6474	0.6621	0.7174	0.6626
MISF	0.9211	**0.9373**	**0.9263**	**0.9540**

Table 5: Performance on purified datasets measured by ACC

Evaluation of Threshold Settings In our method, a sample is considered clean if it is correctly predicted and its Gini impurity is below the threshold. Therefore, it is important to investigate the effect of different threshold values. Figure 3 presents the TPR and FPR under varying thresholds for both symmetric and asymmetric noise settings. As the threshold increases, both TPR and FPR generally decrease. When the threshold is set too low, the FPR remains high across all datasets and noise types. A high threshold leads to a sharp drop in TPR, suggesting many fault samples are misclassified as clean. Based on this analysis, we set the threshold to 0.5.

Validation of Iterative Model Optimization Effectiveness We validated the effectiveness of iterative model optimization using F1 scores. Figure 4 demonstrates the F1 scores of clean samples selected by the current model during iterative optimization. These scores serve as an evaluation metric for the detection performance of the current model. In the figure, iterative epoch 1 represents the results of clean sample detection using the initial model. As the number of itera-

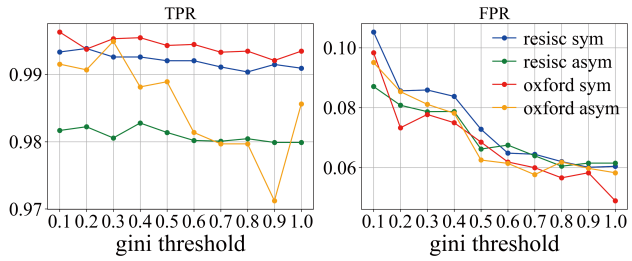
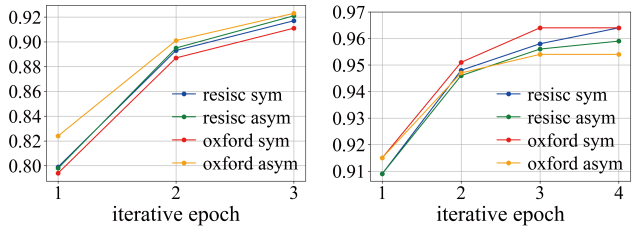


Figure 3: Impact of Gini threshold on TPR and FPR

tive epochs increases, the F1 scores consistently improve, demonstrating the effectiveness of iteratively selecting clean samples for model optimization.



(a) 20% label noise (b) 40% label noise

Figure 4: Comparative F1-score progression over training iterations under different noise ratios

4.6 Ablation Study

Gini vs. Confidence for Sample Selection To investigate the impact of different strategies in iterative clean sample selection, we conduct a comparative analysis between using Gini impurity and confidence to detect data fault. The method using Gini impurity identifies clean samples by selecting those with lower Gini impurity, while the method using confidence directly chooses samples exhibiting higher predicted probability. We set the threshold to 0.5 and conduct experiments on RESISC45 and Oxford with both symmetric and asymmetric noise at a noise ratio of 20%. The results are presented in Table 6. On the RESISC45 dataset, using gini or using confidence to choose samples achieve the same TPR, but method of using gini yields a lower FPR. On the Oxford dataset, gini achieves a higher TPR than Confidence, demonstrating the effectiveness of using the gini value for clean sample selection. Therefore, we adopt the gini value to determine whether a sample is clean.

Effect of Curated and Generated Data on Model Initialization MISF initializes models using generated images and curated clean data. To validate the effectiveness of using both generated images and a curated clean subset for initialization, we conduct experiments comparing three initialization strategies: training solely on generated images (Only G), solely on the curated clean subset (Only C), and using both generated images and clean data (G+C). Table 7 presents detection results across noise ratios (20%: top rows; 40%: bottom rows).

Without supplemental clean data, MISF achieves robust performance (TPR exceeding 0.95 across conditions). When using only a curated clean subset, it is still capable of effectively identifying fault samples. Incorporating clean data further enhances effectiveness, elevating TPR beyond 0.98 while reducing FPR below 0.08.

		Sym		Asym		Avg	
		TPR	FPR	TPR	FPR	TPR	FPR
RESISC	Conf	0.996	**0.039**	0.994	0.041	0.995	0.040
	Gini	**0.997**	0.043	**0.995**	**0.035**	**0.996**	**0.039**
Oxford	Conf	0.994	**0.044**	0.978	0.040	0.986	**0.042**
	Gini	**0.997**	0.055	**0.988**	**0.036**	**0.992**	0.045

Table 6: Comparison of Gini impurity and confidence for clean sample selection under 20% noise level

	RESISC45				Oxford			
	Sym		Asym		Sym		Asym	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Only G	0.981	0.109	0.967	0.107	0.991	**0.050**	0.960	0.066
Only C	0.987	0.089	**0.993**	0.088	0.992	0.078	0.980	**0.051**
G+C	**0.992**	**0.071**	0.981	**0.066**	**0.994**	0.068	**0.988**	0.062
Only G	0.982	0.153	0.955	0.148	0.991	0.065	0.964	0.117
Only C	0.996	0.048	**0.993**	0.050	0.991	0.061	0.973	0.057
G+C	**0.997**	**0.040**	0.984	**0.045**	**0.996**	**0.046**	**0.981**	**0.050**

Table 7: Comparison of different initialization strategies using curated clean data and generated images

5 Conclusion

In this paper, we present MISF (MLLM-Guided Iterative Sample Filtering), a robust and scalable framework for data fault detection that initializes the model with MLLM generated images and a clean subset, and iteratively optimizes it via Gini guidance to effectively detect fault samples. To overcome the limitations of relying on clean data or few-shot MLLM predictions, MISF leverages synthetic data generated by MLLMs combined with curated clean subset to initialize a fault detection model. It then iteratively selects high-confidence clean samples based on label-predict consistency and Gini impurity, enabling progressive model adaptation and choosing clean samples. Furthermore, we propose a novel stopping criterion based on Gini trend analysis, which effectively prevents model degradation and unnecessary computation. Experiments on the RESISC45 and Oxford-IIIT Pets datasets show that MISF consistently outperforms existing baselines in both label noise and data poisoning. Despite the great success, there are still many directions to be explored in the future. MISF focuses primarily on data faults caused by altered labels; extending the framework to detect more subtle faults where labels remain unchanged remains an open and important challenge.

References

Barni, M.; Kallas, K.; and Tondi, B. 2019. A New Backdoor Attack in CNNS by Training Set Corruption Without Label Poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, 101–105.

Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I.; and Srivastava, B. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Cheng, G.; Han, J.; and Lu, X. 2017. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 105(10): 1865–1883.

De Gaspari, F.; Hitaj, D.; and Mancini, L. V. 2024. Have You Poisoned My Data? Defending Neural Networks Against Data Poisoning. In *European Symposium on Research in Computer Security*, 85–104. Springer.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Gao, Y.; Xu, C.; Wang, D.; Chen, S.; Ranasinghe, D. C.; and Nepal, S. 2019. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, 113–125.

Gu, T.; Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7: 47230–47244.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, J.; Qu, L.; Jia, R.; and Zhao, B. 2019. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3326–3334.

Koh, J. Y.; Fried, D.; and Salakhutdinov, R. 2023. Generating Images with Multimodal Language Models. *NeurIPS*.

Lam, P.; Nguyen, H.-L.; Dang, X.-T. D.; Tran, V.-S.; Le, M.-D.; Nguyen, T.-T.; Nguyen, S.; and Vo, H. D. 2025. Leveraging local and global relationships for corrupted label detection. *Future Generation Computer Systems*, 166: 107729.

Li, D.; Li, J.; Le, H.; Wang, G.; Savarese, S.; and Hoi, S. C. 2023. LAVIS: A One-stop Library for Language-Vision Intelligence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, 31–41. Toronto, Canada: Association for Computational Linguistics.

Li, Y.; Li, Y.; Wu, B.; Li, L.; He, R.; and Lyu, S. 2021. Invisible Backdoor Attack with Sample-Specific Triggers. In *IEEE International Conference on Computer Vision (ICCV)*.

Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc.

Nahum, O.; Calderon, N.; Keller, O.; Szpektor, I.; and Reichart, R. 2024. Are llms better than reported? detecting label errors and mitigating their effect on model performance. *arXiv preprint arXiv:2410.18889*.

Nicolae, M.-I.; Sinn, M.; Tran, M. N.; Buesser, B.; Rawat, A.; Wistuba, M.; Zantedeschi, V.; Baracaldo, N.; Chen, B.; Ludwig, H.; Molloy, I.; and Edwards, B. 2018. Adversarial Robustness Toolbox v1.2.0. *CoRR*, 1807.01069.

Northcutt, C.; Jiang, L.; and Chuang, I. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70: 1373–1411.

OpenAI(2024). 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. V. 2012. Cats and Dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Peri, N.; Gupta, N.; Huang, W. R.; Fowl, L.; Zhu, C.; Feizi, S.; Goldstein, T.; and Dickerson, J. P. 2020. Deep k-nn defense against clean-label data poisoning attacks. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 55–70. Springer.

Pruthi, G.; Liu, F.; Kale, S.; and Sundararajan, M. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning*, 1(1): 81–106.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695.

Tran, B.; Li, J.; and Madry, A. 2018. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31.

Van Rooyen, B.; Menon, A.; and Williamson, R. C. 2015. Learning with symmetric label noise: The importance of being unhinged. *Advances in neural information processing systems*, 28.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, 707–723. IEEE.

Yu, C.; Ma, X.; and Liu, W. 2023a. Delving into Noisy Label Detection with Clean Data. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 40290–40305. PMLR.

Yu, C.; Ma, X.; and Liu, W. 2023b. Delving into noisy label detection with clean data. In *International Conference on Machine Learning*, 40290–40305. PMLR.

Zhu, D.; Hedderich, M. A.; Zhai, F.; Adelani, D. I.; and Klakow, D. 2022. Is BERT robust to label noise? A study

on learning with noisy labels in text classification. *arXiv preprint arXiv:2204.09371*.

Zhu, Z.; Dong, Z.; and Liu, Y. 2022. Detecting corrupted labels without training a model to predict. In *International conference on machine learning*, 27412–27427. PMLR.