

## Federated Learning for Defect Detection in Additive Manufacturing: Mitigating Label-Flipping Attacks in Distributed Factories

Md Sazol Ahmmed <sup>1\*</sup>, Sriram Praneeth Isanaka <sup>2</sup>, Sung-Heng Wu <sup>1</sup>, Atiqur Rahman<sup>1</sup>, Muhammad  
Arif Mahmood <sup>2</sup>, Frank Liou <sup>1</sup>

<sup>1</sup> Department of Mechanical and Aerospace Engineering, Missouri University of Science and  
Technology, Rolla, MO 65409, USA.

<sup>2</sup> Intelligent Systems Center, Missouri University of Science and Technology, Rolla, MO 65409,  
USA.

\* Corresponding author – Email: ma8gf@mst.edu, Telephone: +15736471355,

Address: Interdisciplinary Engineering Building, Room: M1 A-B,

1215 N Pine St, Rolla, MO 65409 Rolla, MO, 65409, USA.

### Abstract

In modern manufacturing, Distributed Digital Factories (DDFs) is a revolutionary concept that enables collaborative manufacturing across several geographically separate factories but faces challenges in centralized quality control due to data privacy concerns. Federated Learning (FL) offers a privacy-preserving solution by enabling model training without raw data exchange; however, it remains vulnerable to label-flipping (LF) attacks. We propose an FL framework enhanced with MUD-HoG (Malicious and Unreliable Client Detection using History of Gradients) also considered here as Multi-Dimensional History of Gradients and Hierarchical Clustering, which detects malicious clients by analyzing gradient patterns without accessing raw data. To evaluate its effectiveness, Additive manufacturing (AM) data with six input features and three output classes (No Powder, No Laser, Normal) have been used; we simulated a strong LF attack on the weakest client. As a result, the attack reduced global model accuracy to 35.07%, and class-wise F1-scores fell below 0.40. After applying MUD-HoG, the system successfully detected and excluded the compromised client, recovering global accuracy to 98.05%, and improving macro and weighted F1-scores to 0.8006 and 0.8010, respectively. The “Normal” class F1-score improved from 0.35 to 0.97, with over 25% gains in precision and recall for other defect classes. This method ensures secure, scalable defect detection in DDFs, advancing the deployment of reliable, privacy-preserving smart manufacturing networks.

**Keywords:** Distributed Digital Factory; Federated Learning; Label Flipping Attack; Additive Manufacturing; Malicious and Unreliable Client Detection using History of Gradients.

## 1. Introduction

The landscape of modern manufacturing is undergoing a significant transformation driven by globalization, increased customer demands for customization, and rapid technological advancements. Under the strains of aggressive competition and changing market needs, traditional manufacturing systems characterized by centralized production at distinct plant sites are progressively challenged. Limited technical integration, a lack of real-time visibility, ineffective resource use, and delayed response times to system faults [1], [2] are common shortcomings of conventional systems. Human involvement is still very necessary, especially in the wake of system failures that cause significant production downtime and financial losses [3], [4].

Emerging as a potential solution for these constraints is the Distributed Digital Factory (DDF) paradigm. A DDF is a distributed system that links geographically scattered manufacturing sites therefore facilitating dynamic product and production reconfigurability and resource sharing. The DDF provides higher agility, lower lead times, and better supply chain resilience by tying factories together via digital technology [5], [6], [7]. Crucially, DDFs combine additive manufacturing (AM) and subtractive manufacturing (SM) technologies to maximize the capabilities of both paradigms. While AM provides fast prototyping and even production, by adRequest to review of our paperdressing geometric complexity, and material efficiency, SM offers exceptional accuracy and finishing capability [8], [9], [10], [11].

By allowing real-time monitoring, process optimization, and predictive maintenance in digital twins, 3D printing, and smart manufacturing [12], [13] DDF viability has been significantly enhanced. Particularly in AM, methods such Laser Powder Bed Fusion (LPBF) and Direct Energy Deposition (DED) have become well-known for their capacity to create intricate, high-performance components from a broad spectrum of materials [14], [15]. Maintaining product quality and defect detection is one of the most important elements of running a dispersed manufacturing network. Due to data privacy issues, high data transfer costs, and possible exposure of sensitive production data, traditional quality control systems mostly depend on centralized data collecting and processing, which might be difficult to implement in a distributed environment [16], [17]. Here, FL has become a viable solution allowing several production locations to jointly train machine learning models without exchanging actual data [18], [19]. Under FL, individual sites

exchange just the model updates with a central server or among peers but train local models on their own unique data ensuring data privacy and security. This distributed technique ensures data privacy while letting the collective model gain insights from the entirety of the distributed network. In the case of defect detection in AM, where extensive and varied datasets from many production contexts can increase model resilience and generalization, FL can provide major benefits. FL isn't without its flaws. An LF attack is a well-known threat whereby malevolent users intentionally alter the labels of training data to deceive the model [20], [21]. Effective LF attacks in defect detection systems can enable damaged components to be misclassified as acceptable, potentially causing life-threatening failures in important sectors, including aerospace, medical devices, and automotive industries. Beyond safety concerns, these strikes can also cause significant financial losses and brand reputation harm [22], [23].

Because of the distributed nature of the system and the absence of access to raw data, identifying LF attacks is exceedingly challenging in federated environments. Since they depend on centralized data aggregation and human inspection procedures that are impracticable at scale, conventional anomaly detection techniques are generally insufficient for this purpose. In this study, we offer a methodology to minimize LF attacks in FL-based defect detection systems for AM in remote digital factories. Our study focuses on improving the security and dependability of FL by means of techniques to identify and resist LF threats without endangering system efficiency or data privacy.

## **2. Literature Review**

The shift from centralized manufacturing systems to DDFs has been fueled by the demand for greater flexibility, customization, and resilience under global competition and volatile market conditions. By use of sophisticated digital technology and integration of SM and AM processes, DDFs link geographically scattered manufacturing sites [24], [25], [26], [27]. With AM methods like DED and Laser LPBF allowing the capacity to produce complicated, high-performance components, the development of technologies such digital twins, and real-time monitoring, has helped to enable the use of DDFs [28], [29].

Due to data privacy needs, IP protection and security and the associated high transmission costs, quality assurance and defect detection across scattered manufacturing locations provide significant hurdles. FL has been popular as a distributed method for jointly training machine learning models without exchanging private source data. For defect identification in AM, where using different, dispersed datasets might result in more accurate and strong models, FL is particularly exciting [30], [31]. But FL can be challenged, most notably LF Attacks, when malevolent players purposefully alter data labels to compromise the model [31], [32]. Proposed countermeasures for these dangers include cryptographic techniques such as HE and Blockchain, which guarantee safe model aggregation but usually at great computational expense. While defending against poisoning assaults, non-cryptographic countermeasures such robust aggregation (e.g., Krum, Trimmed Mean) and similarity-based filtering (e.g., FoolsGold) have sought to lower computational costs [33], [34].

FoolsGold [35] is a protection technique meant to lower the Sybil attack count by the use of cosine similarity analysis of client update variety. It also assesses FL's flaws. Unlike the techniques now in use, Fools-Gold does not need prior knowledge of the number of attackers, security architecture, or supplementary information. The authors show that FoolsGold is successful against LF and backdoor poisoning attacks by evaluating it on several different datasets, including MNIST, VGGFace2, KDDCup, and Amazon. The data suggests that FoolsGold offers better outcomes without the need for major changes to the FL mechanism. The authors also show that FoolsGold provides a scalable and reasonable approach to shield federated models from sybil-based assaults. N. M. Jebreel et al. and Yanli Li et al. [36], [37] contributed to FL security by identifying and addressing weaknesses in existing methods against LF attacks under Non-Independent and Identically Distributed data (non-IID) settings. The authors carefully evaluate Byzantine-resilient aggregation methods like Krum, Trimmed Mean, and FL Trust and demonstrate that under mon-IID setting, those models failed to differentiate between hostile and benign clients. Experimental data indicates that in attacked scenarios, Hierarchical Secure Client Selection in Federated Learning guarantees strong learning across all classes in non-IID environments and increases accuracy in targeted courses. Client involvement is regulated by the threshold of customer selection ( $p\%$ ).

The P. R. Ovi et al. [38] built a federated framework using the FedAvg algorithm, where clients can train locally on their datasets and subsequently transmit model parameters to the server running the framework remotely. This solution aims to safeguard data confidentiality, separate between data poisoning and model poisoning assaults, and find consumers whose security has been compromised. By use of weight distribution and neuron activity in the last layer, this system effectively detects dangerous consumers as well as the types of attacks they are experiencing. Moreover, Multi-Dimensional History of Gradients and Hierarchical Clustering on Gradients (MUD-HoG) [39] is a noteworthy development as it uses dimensionality reduction and clustering to discriminate between benign and hostile users in non-IID data situations. Outperforming various state-of-the-art defenses, MUD-HoG has shown great detection accuracy even in cases of client compromise.

Although FL is an effective path for defect identification in AM inside DDFs, it is nevertheless prone to LFAs, especially in non-independent and non-IID data setups common in industrial environments. Although strong security assurances are offered by cryptographic methods such HE and Blockchain, their large computational and communication overheads make practical application in resource-limited DDFs difficult. Although non-cryptographic techniques like Krum, Trimmed Mean, and FoolsGold have more light weight protections, they have shown notable shortcomings in managing adversarial assaults in non-IID settings. By using dimensionality reduction and clustering for identifying hostile clients even in non-IID environments, recent developments as MUD-HoG have demonstrated exceptional performance. But, MUD-HoG has not yet been modified or assessed, in the framework of federated defect detection in AM via distributed manufacturing networks. These elements need specific modifications of current FL defensive systems to guarantee scalable, lightweight, and strong operations. Customizing and improving MUD-HoG for safe, privacy-preserving, and effective defect detection in remote AM settings is thereby a critically lacking study. In this context, our work expands on MUD-HoG ideas and adapts and improves on their strengths to meet the particular needs of federated defect detection in AM across distant digital factories.

### 3. Methodology

#### 3.1 Federated Learning with MUD-HoG Framework

Without explicitly sharing their proprietary data, FL lets many factories combine together to create a machine learning model. Unlike Centralized Learning (CL), in which all data is consolidated in a single place, FL guarantees that sensitive and proprietary local datasets stay inside their source factories. Only localized model updates and gradients (the partial derivatives of the loss function with respect to model parameters) are securely sent to a central server for aggregation, therefore maintaining data privacy and supporting group learning by means of gradients, which has been demonstrated in Figure 01.

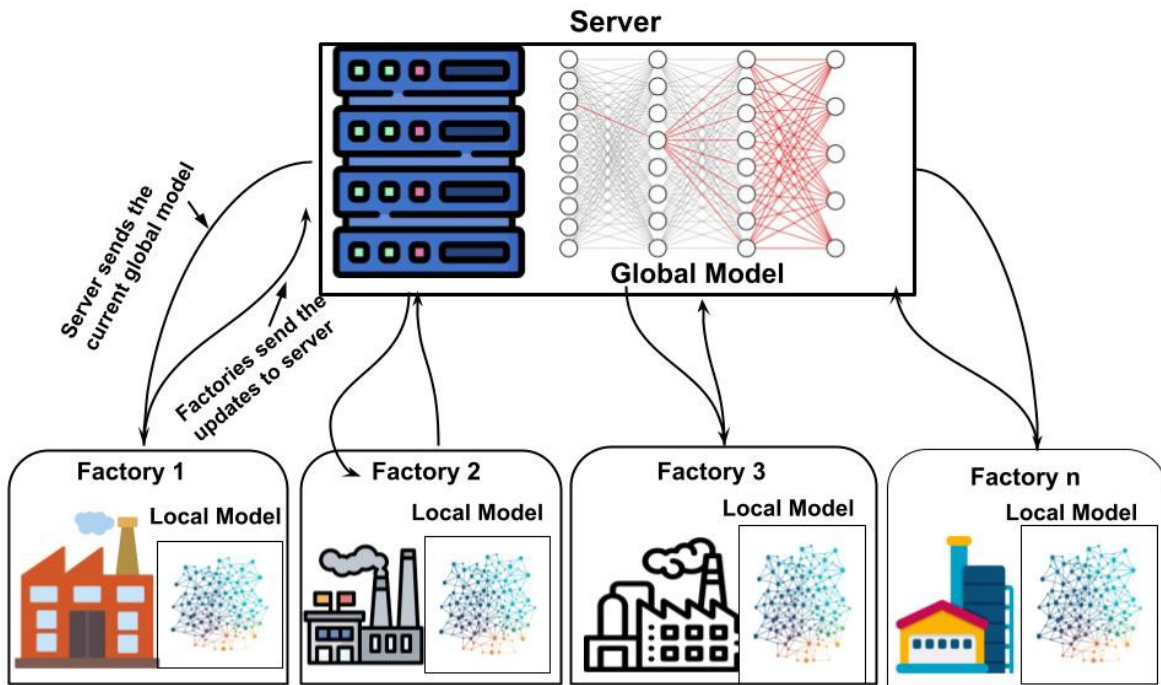


Figure 1: Schematic Diagram of FL.

The MUD-HoG mechanism is included in the proposed framework to improve security against adversarial attacks, especially LF and model-poisoning assaults, by means of the normal FL protocol. The general training procedure for such an FL algorithm consists of these phases:

1. Initialization

The central server initializes the global model  $w^0$  with random weights and transmits these parameters to all participating factories.

## 2. Local Training

Each factory downloads the global model as its own instance, namely  $w^\tau$ , and trains it locally on its private dataset  $D_i$  for a small number of local epochs  $E$  using stochastic gradient descent (SGD) or the Adam optimizer. Local updates  $\nabla_{\tau,i}$  are computed based on minimizing the local loss function.

## 3. Gradient Transmission

Instead of transmitting updated model parameters, each factory securely transmits the computed gradients  $\nabla_{\tau,i}$  to the central server.

## 4. Client Behavior Analysis via MUD-HoG

The central server applies the MUD-HoG framework to analyze the historical gradients from all factories:

- Compute the Short History of Gradients (Short HoG) to detect untargeted attacks.
- Compute the Long History of Gradients (Long HoG) to detect targeted attacks.
- Identify and exclude malicious clients and down-weight unreliable clients before aggregation.

## 5. Robust Aggregation

The server aggregates the gradients using a weighted scheme [39] that excludes these detected malicious clients and down-weights unreliable clients:

$$\nabla_{\tau} = \sum_{i \in C_{norm}} \frac{|D_i|}{|D|} \nabla_{\tau,i} + \alpha \sum_{i \in C_{unrl}} \frac{|D_i|}{|D|} \nabla_{\tau,i}$$

where  $\alpha \in (0,1)$  is a down-weighting factor for unreliable clients.

## 6. Global Model Update

The global model is updated from the weighted data of all clients as [39]:

$$w^{\tau+1} = w^\tau - \eta \nabla_{\tau}$$

where  $\eta$  is the learning rate.

## 7. Iteration

Steps 2 to 6 are repeated over multiple communication rounds until the global model converges according to predefined convergence criteria, such as an improvement in validation loss or achieving a target accuracy.

Using important process metrics like Melt Pool Area (MPA), Mussy Zone region Temperature and Peak Temperature, each factory's local dataset is classified into three categories: No Laser, No Powder, and Normal. Every factory divides its dataset into 25% for testing and 75% for model assessment. After training, the global model is distributed and applied across the domains of all participating factories to enable component certification and accurate defect prediction. Figure 2 illustrates the insights of local factory training and testing procedures and their relationship with the central server.

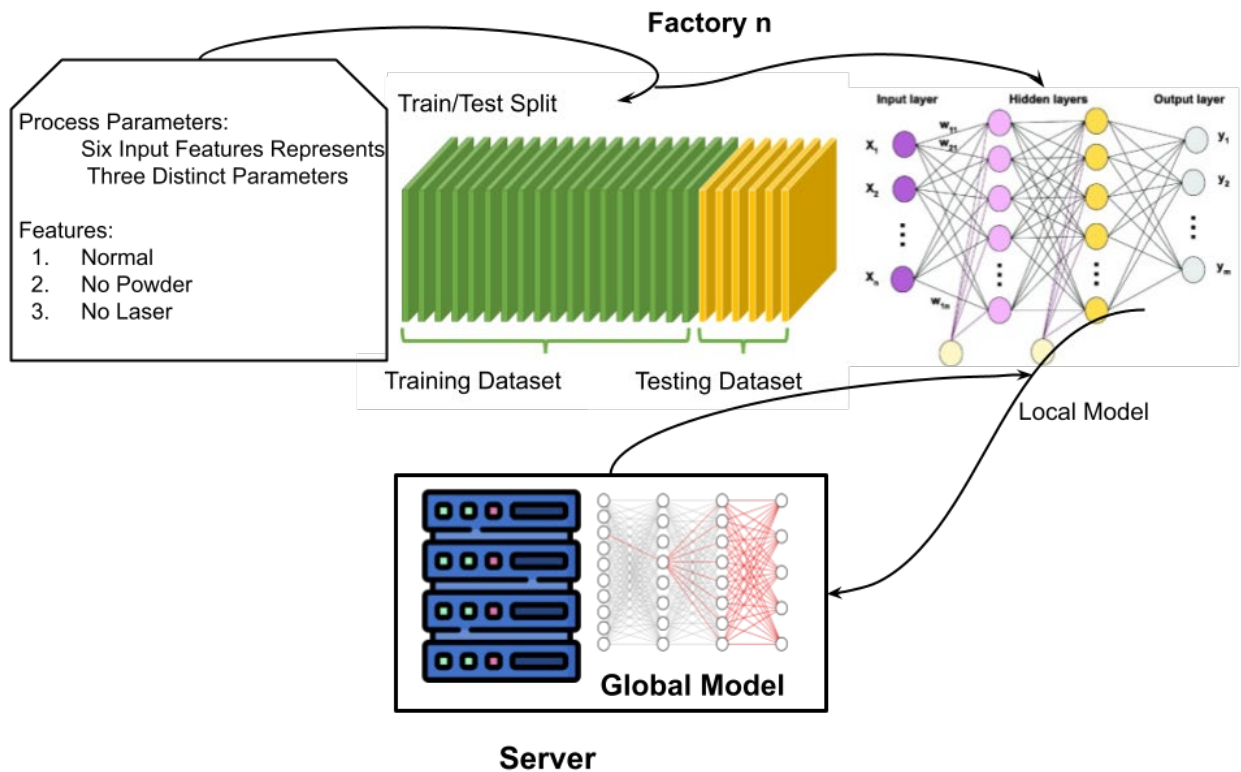


Figure 2: Individual Factory Insights and Relationship with Central Server.

In order to guarantee accurate defect classification in distributed AM environments, we propose through this work a strong and privacy-preserving FL framework enhanced with the MUD-HoG defense system. The aim is to jointly train a model without revealing private local data and hence reduce the effect of hostile customers.

### 3.2 FL Problem Setup

This research collates an FL setup consisting of a central server and  $N$  distributed clients (factories). Each client  $C_i$  possesses a local dataset  $D_i$  containing sensor-based process monitoring

parameters, specifically Melt Pool Area (MPA), Mussy Zone region Temperature and Peak Temperature paired with specific labels  $y \in \{0,1,2\}$ , corresponding to three process conditions: No Laser, No Powder, and Normal. The learning task is formulated as a multi-class classification problem. Each client minimizes the following cross-entropy loss function locally:

$$L(h_w(x), y) = - \sum_{k=0}^2 1[\{y = k\}] \log h_w^k(x)$$

where  $h_w(x)$  is the model output, and  $h_w^k$  denotes the predicted probability of input  $x$  for class  $k$ . At each communication round  $\tau$ , the local gradient update [39] is computed as:

$$\nabla_{\tau,i} = w^\tau - \text{argmin} L(h_w(x), y)$$

and transmitted to the server.

The global gradient aggregation in benign cases is:

$$\nabla_\tau = \sum_{i=1}^N \frac{|D_i|}{|D|} \nabla_{\tau,i}$$

Where,  $\nabla_\tau = \sum_{i=1}^N |D_i|$  is the total number of samples.

The global model update rule is:

$$w^{\tau+1} = w^\tau - \eta \nabla_\tau$$

where  $\eta$  is the learning rate. However, the presence of adversarial clients necessitates a more robust aggregation strategy, motivating the integration of MUD-HoG.

### 3.3 Client Types and Threat Model

We assume the existence of three types of clients:

- Normal Clients: Honest participants with representative, high-quality data.
- Unreliable Clients: Clients with noisy or poor-quality data.
- Malicious Clients: Attackers performing targeted LF poisoning attacks.

Malicious clients while persistent can only manipulate their own data or updates.

### 3.4 MUD-HoG Defense Mechanism

MUD-HoG analyzes the historical gradients from each client to distinguish between normal, unreliable, and malicious clients.

The history of gradients for client  $i$  up to round  $\tau-1$  are defined as:

$$\nabla_i = \{\nabla_{1,i}, \nabla_{2,i}, \dots, \nabla_{\tau-1,i}\}$$

Two constructs [39] are also defined as:

- Short History of Gradients (Short HoG):

$$\nabla_i^{Short} = \frac{1}{l} \sum_{t=\tau-l}^{\tau-1} \nabla_{t,i}$$

where  $l$  is the sliding window size.

- Long History of Gradients (Long HoG):

$$\nabla_i^{Long} = \sum_{t=1}^{\tau-1} \nabla_{t,i}$$

Short HoG smooths recent updates, useful for detecting untargeted attacks. Long HoG captures cumulative influence over time, helpful for detecting targeted attacks.

### 3.5 Model Architecture and Training Details

In the current FL framework, the classification model used by each client is a lightweight linear classifier based on logistic regression, implemented using SGD (stochastic gradient descent) Classifier from Scikit-learn. This choice allows for fast, memory-efficient training across distributed clients. Each client receives input samples with six numerical features, derived from sensor readings. The model architecture consists of:

- Input Layer: 6 input neurons corresponding to the six features
- Output Layer: 3 output neurons corresponding to the three class labels:
  - No Powder (0)
  - No Laser (1)
  - Normal (2)

Utilizing the `partial_fit` approach, the classifier trained by stochastic gradient descent (SGD) utilizing logistic loss (`log_loss`). Appropriate for edge deployment, this replicates one local update step from each communication cycle. Each client's dataset is leveraged 70% of training, while the remaining 30% is utilized for assessment. Following every client's local update, gradient vectors—including model weights and intercept—are retrieved to facilitate anomaly identification

using MUD-HoG. These are then searched across customers using cosine similarity to identify potentially harmful updates.

## 4. Performance Evaluation

### 4.1 Experimental SetUp

The experiments were carried out with an in-house designed Directed Energy Deposition (DED) apparatus housed at Missouri University of Science and Technology, Rolla, MO, USA. Comprising a neodymium-doped yttrium aluminum garnet (Nd:YAG) optical laser system with up to 1 kW of laser power capability, this DED machine is shown in Figure 3.

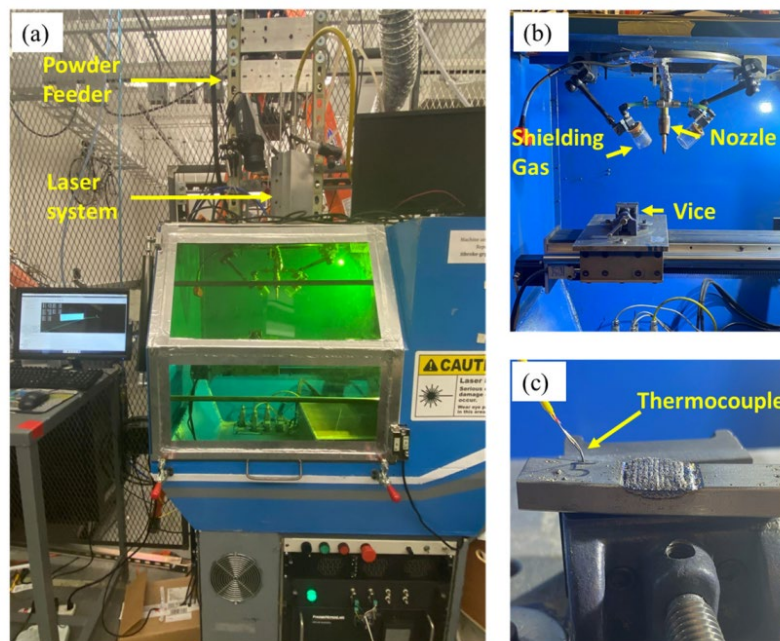


Figure 3: Overview of inhouse DED System [40]; published under open-access license.

The system is capable of three-dimensional movement for exact spatial control during deposition, and includes a shielding gas module coupled with an off-axis powder feeding nozzle (Figure 3b). A multi-sensor configuration was used to track melt pool dynamics and evaluate numerical models. To capture temperature data, K-type thermocouples were attached to the substrate (Figure 3c). Two high-speed cameras were set up focused on recording the melt pool geometry and surrounding regions of temperature gradient during deposition. Camera 1 is calibrated for thermal imaging by tracking melt pool temperature distribution of the top profile of

the deposit region in real-time, and Camera 2 provides side profile thermal gradient information during deposition. Accurate measurement of important process variables was made possible by this coordinated multi-sensor data collection.

#### 4.1.1 Data Acquisition for Federated Learning

The study utilized three real-world datasets representing three part conditions in AM: No Laser, No Powder, and Normal. Moreover, for this experiment Ti-6Al-4V commonly referred to as Ti64, is a high-performance titanium alloy with a composition of approximately 6% aluminum, 4% vanadium, and the remainder is primarily titanium has been used to make the final prints. These datasets were partitioned into five subsets to simulate five independent clients operating under non-IID conditions as shown in Figure 4.



Figure 4: Class Distribution Per Client Dataset Under Non IID Settings.

Every dataset reflects data silos usually present in dispersed industrial systems, therefore representing a different client. Along with a three-class output label showing the component conditions of No Powder, No Laser, and Normal, the datasets include six input attributes derived from machine settings, material specifications, and sensor readings. Every client keeps its dataset locally and engages in federated training by exchanging only model updates rather than raw data,

hence mimicking privacy-preserving industrially decoupled learning. Natural non-identically distributed (non-IID) features among clients are introduced by the different size and class distributions of the datasets, therefore posing a fundamental difficulty in federated optimization. Based on local model accuracy, an LF attack was purposefully included in the weakest-performing client to evaluate the resilience of the FL system. In particular, 90% of the Normal labels were changed to No Powder, therefore representing a corrupted or malevolent client. By use of the MUD-HoG architecture, this adversarial arrangement allowed for the identification and rejection of these poisoned or malicious updates during aggregation. Each dataset was then separated into 70% used for training purposes and 30% for testing the subsets for model assessment and preprocessed to guarantee numerical encoding and label consistency. Thus, closely reflecting real-world deployment situations in industrial AM systems distributed across geography, the data gathering approach allows both traditional FL as well as resilient aggregation under adversarial settings.

#### **4.1.2 Federated Learning Configuration**

The FL simulation was conducted with five clients, each representing a virtual factory with its own locally stored and statistically non-IID dataset. The goal was to collaboratively train a global predictive model for defect classification in AM, while preserving data privacy and security across participating sites. Using a logistic regression classifier implemented via `SGDClassifier` with a `log_loss` objective, each client independently trained a local model. The training dataset at each client was split into 70% training and 30% testing, with no data sharing between clients or the server. Local model updates were generated using a single iteration (`partial_fit`) to simulate lightweight edge computation, and training was conducted in a single communication round for the initial evaluation.

After local training, model gradients were extracted and analyzed using the MUD-HoG (Multi-dimensional Histogram of Gradients) algorithm. This defense mechanism computed pairwise cosine distances to identify anomalous clients between client gradients and applied agglomerative clustering. This clustering process allowed the system to isolate and exclude potentially malicious updates, particularly from clients impacted by a simulated LF attack.

Following anomaly detection, federated aggregation was performed using simple weighted averaging of the parameters (weights and biases) from only the benign clients. The resulting global model was then evaluated on each client's local test data to assess robustness and generalization. This configuration effectively models a realistic but adversarial FL environment in distributed industrial systems.

### **4.1.3 Adversarial Attack Simulation**

One of the participating clients was simulated using LF adversarial attacks in order to assess the resilience of the FL system against insider concerns. The attack was meant to deliberately introduce corrupted labels during local training without changing the input characteristics, therefore compromising the integrity of the global model. This situation substantially resembles actual vulnerabilities where hacked edge devices or malevolent insiders might interfere with cooperative learning operations. Specifically, Client 1 identified as the client with the lowest local model performance during pre-FL training was chosen for attack injection and identified as baseline accuracy of 45.27%. 90 percent of the samples in this client's dataset classified as "Normal," (class 2), were changed to "No Powder," (class 0). The aim was to control the client's model gradients and mislead the global model during aggregation, using this high-volume label manipulation.

The poisoning was done before local model training, therefore guaranteeing that the altered labels directly affected the client's update. Still, the client stayed invisible based just on accuracy, which emphasizes the importance of strong gradient-based anomaly detection methods. After local training for every client, the system examined gradient consistency among them using the MUD-HoG architecture. By effectively identifying the poisoned client as anomalous and excluding its update from the federated aggregation process, our technique reduced the effect of the assault. This adversarial arrangement gave an empirical basis for assessing defensive techniques including MUD-HoG in industrial FL contexts and helped to rigorously evaluate the security-aware architecture of the FL pipeline.

## **4.2 Result Analysis**

Five clients, each with a local dataset reflecting various sites, made up a simulated AM environment used for evaluation of the proposed FL framework. The classification challenge

consisted of identifying Normal, No Laser, and No Powder part quality circumstances. The main objectives were to confirm the efficacy of the MUD-HoG defensive mechanism and evaluate the performance of the system in both normal and hostile settings.

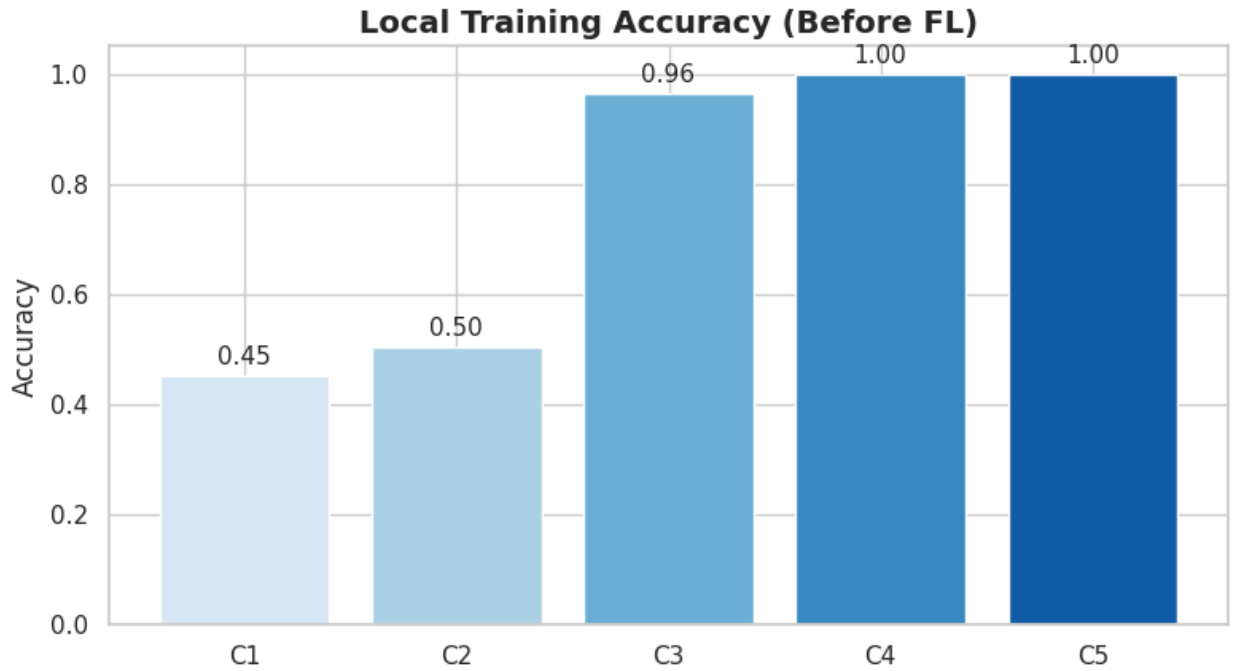


Figure 5: Local Model Accuracy in Each Clients

Using logistic regression, the initial training of local models exposed notable client-wide variance in predicted performance, hence emphasizing the variability of the datasets. While Client 1 and Client 2 exhibited relatively poor performance (45.27% and 50.33%, respectively), Clients 3, 4, and 5 obtained excellent accuracies (96.48%–100%). Figure 5 shows realistic non-IID data distributions in federated industrial systems reflected in this variance.

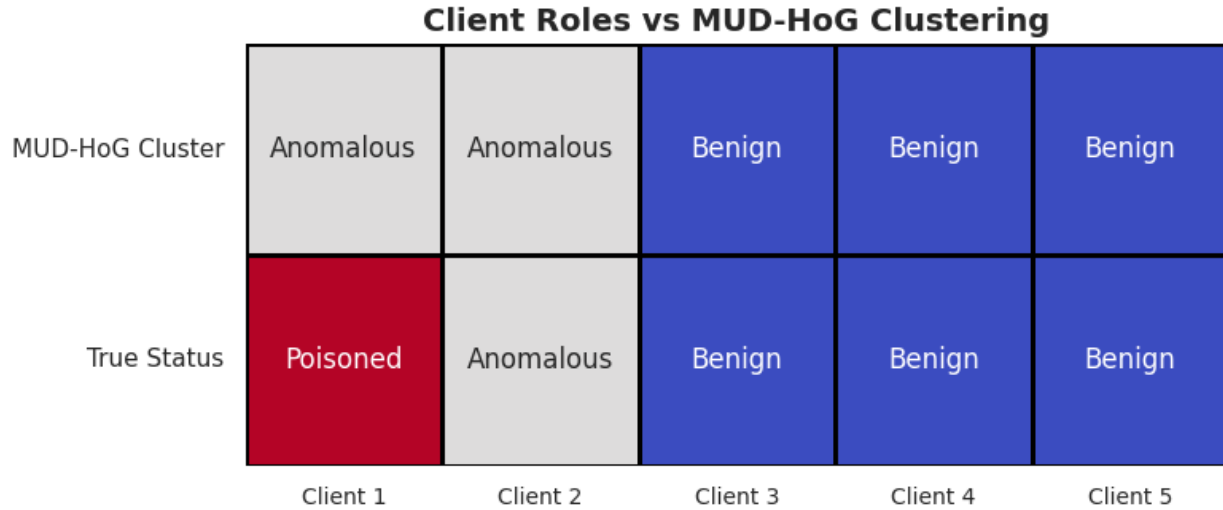


Figure 6: MUD-HoG cluster labels for Comparing Benign clients and Bad Clients.

In Client 1, a focused LF attack was modeled whereby 90% of the Normal labels were turned to No Powder. Using performance-based heuristics by themselves makes it challenging to identify the attack as it seriously compromises the integrity of the local model while preserving somewhat the local accuracy. If this contaminated client were included in the aggregate, the quality of the global model was seriously affected. Following local training, the MUD-HoG anomaly-detecting method was used to examine client gradients. MUD-HoG appropriately detected Client 2 as conservatively unusual and Client 1 as aberrant using pairwise cosine similarity and hierarchical clustering. For federated aggregation, only data from Clients 3, 4, and 5 were retained. In Figure 6, a heatmap comparing genuine client status with MUD-HoG cluster labels helps one visualize these findings. While reducing the possibility of adding compromised clients, the detection accuracy shows the MUD-HoG's sensitivity in spotting hostile gradients.

Table 1: Accuracy of FL Including All Clients vs only Benign clients Aggregation

Round	Accuracy (All Clients)	Accuracy (Malicious Removed)
1	0.3507	0.9805
2	0.3507	0.9805
3	0.3507	0.9805
4	0.3507	0.9805
5	0.3507	0.9805

Table 1 shows when injected with a 90% LF attack, a single poisoned client (Client 1) seriously compromised the accuracy of the global model. Including poisoned updates in aggregate

caused the global accuracy to level to plummet to 35.07% for all five FL rounds. This shows how even one hostile client may control the learning path of the global model and cause convergence to be maligned. Applying the MUD-HoG defensive framework which groups clients based on gradient similarity, the poisoned client was effectively found and removed from the aggregate. The global update was computed using the last benign clients namely Clients 3, 4, and 5. Reflecting both performance stability and adversarial robustness, the federated model routinely attained an average accuracy of 98.05% across all rounds in this scenario.

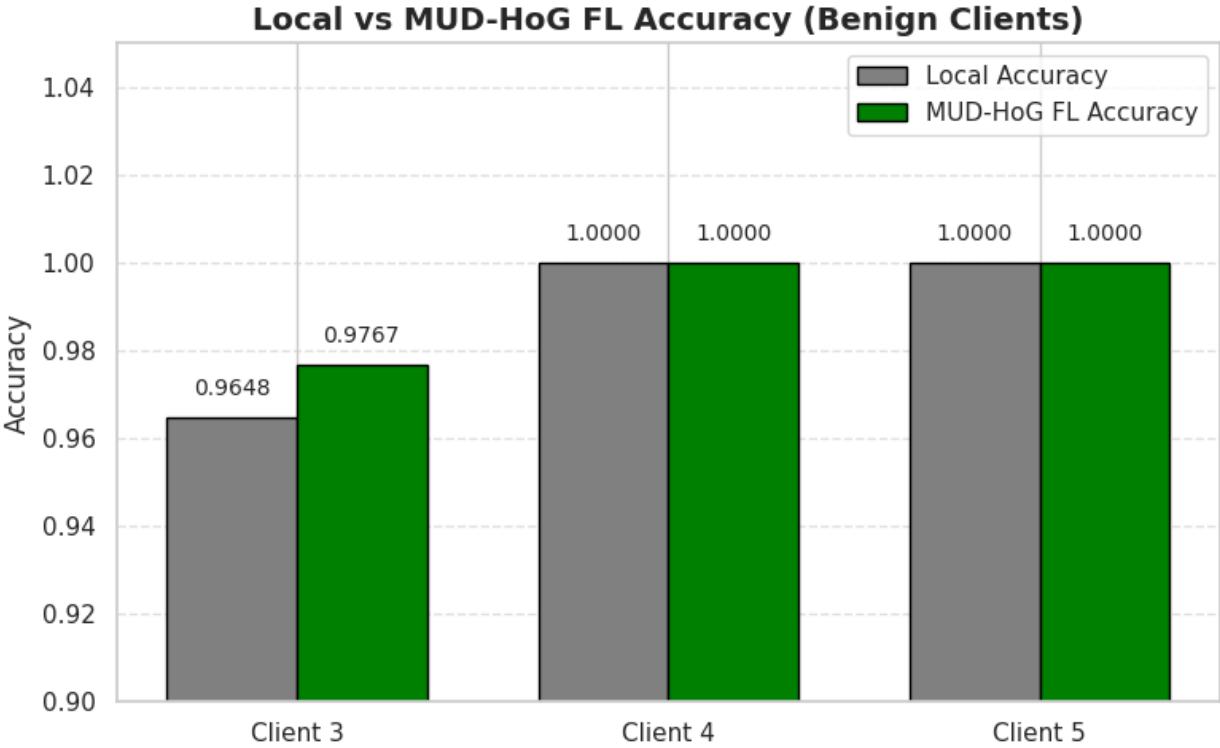


Figure 7: Local Accuracy vs FL Accuracy after MUD HoG Applied

After removing anomalous clients, the federated model was re-trained using only the benign clients. In Figure 7, the final model achieved:

- Client 3 Accuracy: 97.67%
- Client 4 Accuracy: 100.00%
- Client 5 Accuracy: 100.00%

This indicates effective information sharing by federated aggregation as it exhibits an improvement above their independent local models. Furthermore, MUD-HoG filtering kept the

global model strong against hostile impact. We also investigated model performance both before and after the MUD-HoG framework's application into the DED data in order to assess its efficacy in guaranteeing FL. After MUD-HoG was used, Figure 8(a) shows the confusion matrix whereby just benign clients (3, 4, 5) helped to build the model. On the other hand, Figure 8(b) demonstrates the poor performance if every client—including an LF attacker—participated in training. The model recovered dependable classification limits once MUD-HoG eliminated the rogue client. This is shown in the dramatic rise in F1-scores displayed in Figure 8(c), where the "Normal" class F1-score progressed from 0.35 to 0.97 and both "No Powder" and "No Laser" classes improved by more than 20%.

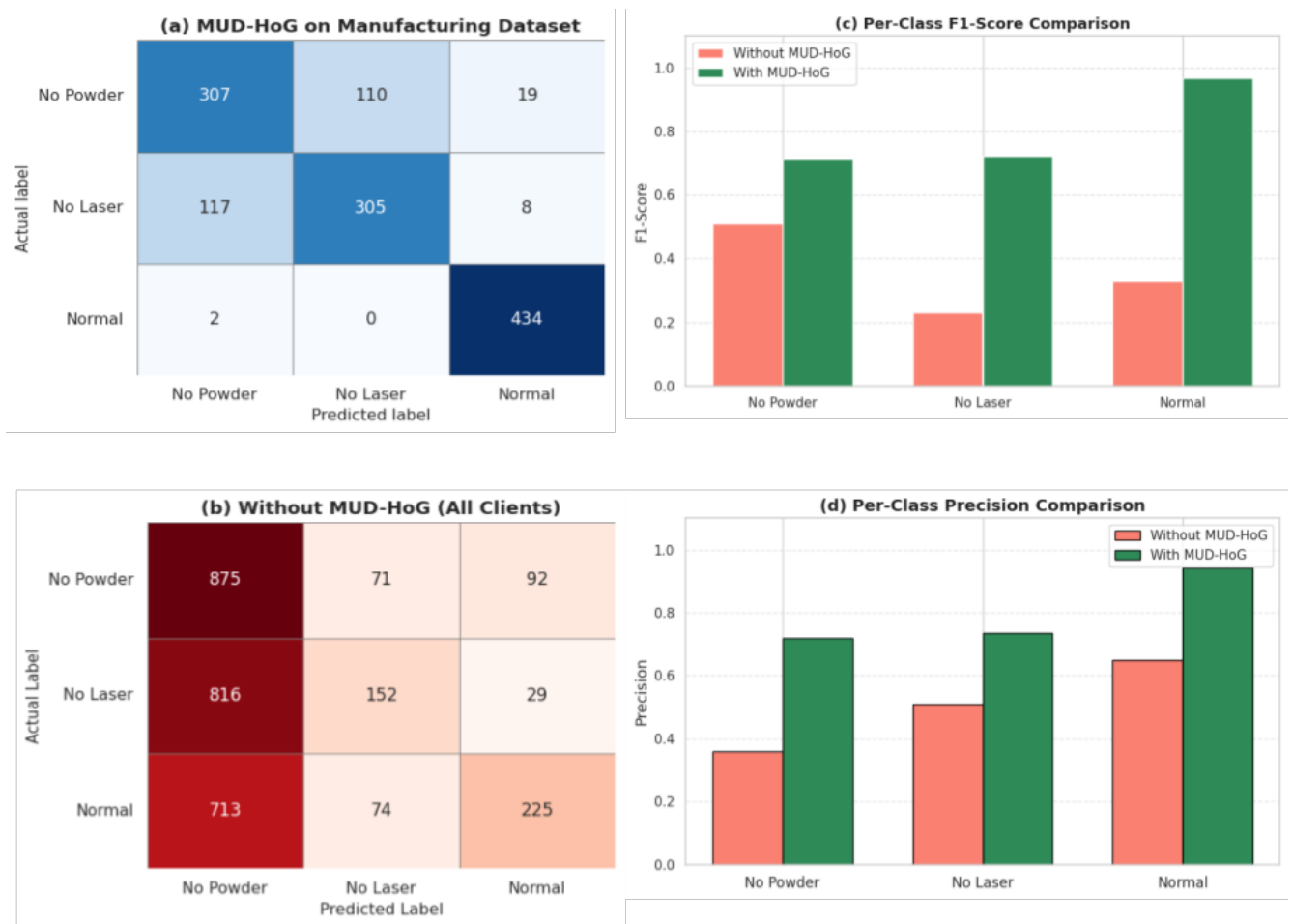


Figure 8: Confusion Matrix, Precision and Accuracy after FL applied between MUD-HoG framework and Without MUD-HoG framework.

Likewise, Figure 8(d) clearly illustrates an increase in per-class accuracy following filtering. These results verify that by reducing the effect of poisoned updates in non-IID federated

systems, MUD-HoG considerably improves classification robustness. Strong performance recovery under adversarial conditions is shown by the model's macro and weighted F1-scores improving to 0.8006 and 0.8010, respectively.

## 5. Conclusion

With a particular focus on adversarial robustness and privacy protection, this work proposes a scalable and efficient FL architecture customized for networked AM settings in DDF. This research demonstrated the susceptibility of collaborative learning systems to LF attacks by modeling real-world non-IID datasets from five AM clients, which, if allowed into training data undetected, can dramatically reduce global model performance. We thus included the MUD-HoG (Multi-dimensional Histogram of Gradients) anomaly detection method into the FL pipeline in order to handle this. In every round, MUD-HoG effectively found and eliminated malicious clients submitting erroneous data for FL training, hence improving global model accuracy from a stationary 35.07% (with all clients) to a stable 98.05% when just benign clients were aggregated. Furthermore, class-wise performance metrics after applying MUD-HoG showed substantial gains:

- F1-Score for “Normal” class improved from 0.35 to 0.97
- Precision and recall for “No Powder” and “No Laser” classes increased by 20–25%
- Macro F1-Score rose to 0.8006, and Weighted F1-Score to 0.8010

The final accuracies of benign clients Client 3 (97.67%), Client 4 (100.00%), and Client 5 (100.00%) also outperformed their local baselines, reflecting the benefits of federated aggregation with MUD-HoG filtering.

This emphasizes how well the system retains strong prediction accuracy even under hostile environments. Furthermore, validating the viability of employing lightweight models and a multi-round gradient sharing technique, the suggested strategy made it possible for real-time, resource-limited AM systems. The framework gets security and scalability by means of gradient-space clustering and local update filtering. Ultimately, this work advances safe, scalable, privacy-aware collaborative learning for smart manufacturing systems. More complicated model structures, adaptive defensive techniques, and scalability to bigger, dynamic federated networks will be the main areas of emphasis in further developments.

## Acknowledgment

We thank the Intelligent Systems Centre (ISC) at the Missouri University of Science and Technology for their support in the pursuance of this research.

## References

- [1] S. Nahavandi, "Industry 5.0—A Human-Centric Solution," *Sustainability*, vol. 11, no. 16, p. 4371, Aug. 2019, doi: 10.3390/su11164371.
- [2] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manuf Lett*, vol. 3, pp. 18–23, Jan. 2015, doi: 10.1016/j.mfglet.2014.12.001.
- [3] P. Muchiri and L. Pintelon, "Performance measurement using overall equipment effectiveness (OEE): literature review and practical application discussion," *Int J Prod Res*, vol. 46, no. 13, pp. 3517–3535, Jul. 2008, doi: 10.1080/00207540601142645.
- [4] B. Bagheri, H. Ahmadi, and R. Labbafi, "Application of data mining and feature extraction on intelligent fault diagnosis by Artificial Neural Network and k-nearest neighbor," in *The XIX International Conference on Electrical Machines - ICEM 2010*, IEEE, Sep. 2010, pp. 1–7. doi: 10.1109/ICELMACH.2010.5607984.
- [5] P. Mehta, P. Rao, Z. (David) Wu, V. Jovanović, O. Wodo, and M. Kuttolamadam, "Smart Manufacturing: State-of-the-Art Review in Context of Conventional and Modern Manufacturing Process Modeling, Monitoring and Control," in *Volume 3: Manufacturing Equipment and Systems*, American Society of Mechanical Engineers, Jun. 2018. doi: 10.1115/MSEC2018-6658.
- [6] M. S. Ahmmed, S. P. Isanaka, A. W. Malik, M. A. Mahmood, and F. Liou, "Feasibility analyses of distributed digital factories over traditional factories: case studies and comparison," *Journal of Electronic Business & Digital Economics*, Apr. 2025, doi: 10.1108/JEBDE-01-2025-0002.
- [7] Md Sazol Ahmmed, Asad Malik, Muhammad Arif Mahmood, Sriram Praneeth Isanaka, and Frank Liou, "Feasibility Analyses of Distributed Digital Factories Integrating Additive and Subtractive Manufacturing: A Case Study," in *Solid Freeform Fabrication 2024: Proceedings of the 35th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference*, Austin, Texas, USA, 2024, pp. 552–575.
- [8] Y. Liu, C. Ma, and Y. Huang, "An Internet of Things-Based Production Scheduling for Distributed Two-Stage Assembly Manufacturing with Mold Sharing," *Machines*, vol. 12, no. 5, p. 310, May 2024, doi: 10.3390/machines12050310.
- [9] D. T. Matt, E. Rauch, and P. Dallasega, "Trends towards Distributed Manufacturing Systems and Modern Forms for their Design," *Procedia CIRP*, vol. 33, pp. 185–190, 2015, doi: 10.1016/j.procir.2015.06.034.
- [10] D. E. P. Klenam *et al.*, "Additive manufacturing: shaping the future of the manufacturing industry – overview of trends, challenges and opportunities," *Applications in Engineering Science*, vol. 22, p. 100224, Jun. 2025, doi: 10.1016/j.apples.2025.100224.

- [11] B. Vayre, F. Vignat, and F. Villeneuve, "Designing for Additive Manufacturing," *Procedia CIRP*, vol. 3, pp. 632–637, 2012, doi: 10.1016/j.procir.2012.07.108.
- [12] L. Jin *et al.*, "Big data, machine learning, and digital twin assisted additive manufacturing: A review," *Mater Des*, vol. 244, p. 113086, Aug. 2024, doi: 10.1016/j.matdes.2024.113086.
- [13] M. H. Zafar, E. F. Langås, and F. Sanfilippo, "Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review," *Robot Comput Integr Manuf*, vol. 89, p. 102769, Oct. 2024, doi: 10.1016/j.rcim.2024.102769.
- [14] W. E. Frazier, "Metal Additive Manufacturing: A Review," *J Mater Eng Perform*, vol. 23, no. 6, pp. 1917–1928, Jun. 2014, doi: 10.1007/s11665-014-0958-z.
- [15] P. Badoniya, M. Srivastava, P. K. Jain, and S. Rathee, "A state-of-the-art review on metal additive manufacturing: milestones, trends, challenges and perspectives," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46, no. 6, p. 339, Jun. 2024, doi: 10.1007/s40430-024-04917-8.
- [16] B. S. Guendouzi, S. Ouchani, H. EL Assaad, and M. EL Zaher, "A systematic review of federated learning: Challenges, aggregation methods, and development tools," *Journal of Network and Computer Applications*, vol. 220, p. 103714, Nov. 2023, doi: 10.1016/j.jnca.2023.103714.
- [17] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 24–29, Mar. 2022, doi: 10.1109/IOTM.004.2100182.
- [18] T. Deng, Y. Li, X. Liu, and L. Wang, "Federated learning-based collaborative manufacturing for complex parts," *J Intell Manuf*, vol. 34, no. 7, pp. 3025–3038, Oct. 2023, doi: 10.1007/s10845-022-01968-3.
- [19] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, "Federated learning review: Fundamentals, enabling technologies, and future applications," *Inf Process Manag*, vol. 59, no. 6, p. 103061, Nov. 2022, doi: 10.1016/j.ipm.2022.103061.
- [20] X. Shen, Y. Liu, F. Li, and C. Li, "Privacy-Preserving Federated Learning Against Label-Flipping Attacks on Non-IID Data," *IEEE Internet Things J*, vol. 11, no. 1, pp. 1241–1255, Jan. 2024, doi: 10.1109/JIOT.2023.3288886.
- [21] Y. Jiang, W. Zhang, and Y. Chen, "Data Quality Detection Mechanism Against Label Flipping Attacks in Federated Learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1625–1637, 2023, doi: 10.1109/TIFS.2023.3249568.
- [22] E. Elmahfoud, S. El Hajla, Y. Maleh, S. Mounir, and K. Ouazzane, "Label flipping attacks in hierarchical federated learning for intrusion detection in IoT," *Information Security Journal: A Global Perspective*, pp. 1–17, Nov. 2024, doi: 10.1080/19393555.2024.2434586.
- [23] L. Lavaur, Y. Busnel, and F. Autrel, "Systematic Analysis of Label-flipping Attacks against Federated Learning in Collaborative Intrusion Detection Systems," in *Proceedings of the 19th International*

- Conference on Availability, Reliability and Security*, New York, NY, USA: ACM, Jul. 2024, pp. 1–12. doi: 10.1145/3664476.3670434.
- [24] N. Anumbe, C. Saidy, and R. Harik, “A Primer on the Factories of the Future,” *Sensors*, vol. 22, no. 15, p. 5834, Aug. 2022, doi: 10.3390/s22155834.
- [25] M. Hovanec, P. Korba, M. Vencel, S. Al-Rabeei, “Simulating a Digital Factory and Improving Production Efficiency by Using Virtual Reality Technology,” *Appl. Sci.* 2023, 13, 5118. <https://doi.org/10.3390/app13085118>
- [26] L. Monostori, “Cyber-physical Production Systems: Roots, Expectations and R&D Challenges,” *Procedia CIRP*, vol. 17, pp. 9–13, 2014, doi: 10.1016/j.procir.2014.03.115.
- [27] Md Bahar Uddin, Md. Hossain, and Suman Das, “Advancing manufacturing sustainability with industry 4.0 technologies,” *International Journal of Science and Research Archive*, vol. 6, no. 1, pp. 358–366, Jun. 2022, doi: 10.30574/ijrsra.2022.6.1.0099.
- [28] S. Dehghan, S. Sattarpanah Karganroudi, S. Echchakoui, and N. Barka, “The Integration of Additive Manufacturing into Industry 4.0 and Industry 5.0: A Bibliometric Analysis (Trends, Opportunities, and Challenges),” *Machines*, vol. 13, no. 1, p. 62, Jan. 2025, doi: 10.3390/machines13010062.
- [29] E. H. D. Ribeiro da Silva, A. C. Shinohara, E. P. de Lima, J. Angelis, and C. G. Machado, “Reviewing Digital Manufacturing concept in the Industry 4.0 paradigm,” *Procedia CIRP*, vol. 81, pp. 240–245, 2019, doi: 10.1016/j.procir.2019.03.042.
- [30] M. Mehta and C. Shao, “Federated learning-based semantic segmentation for pixel-wise defect detection in additive manufacturing,” *J Manuf Syst*, vol. 64, pp. 197–210, Jul. 2022, doi: 10.1016/j.jmsy.2022.06.010.
- [31] M. Mehta, M. V. Bimrose, D. J. McGregor, W. P. King, and C. Shao, “Federated learning enables privacy-preserving and data-efficient dimension prediction and part qualification across additive manufacturing factories,” *J Manuf Syst*, vol. 74, pp. 752–761, Jun. 2024, doi: 10.1016/j.jmsy.2024.04.031.
- [32] H. Li, , Z. Shi, M. Jin et al. “FLGT: label-flipping-robust federated learning via guiding trust.” *Knowl Inf Syst* 67, 3399–3422 (2025). <https://doi.org/10.1007/s10115-024-02323-z>
- [33] F. Colosimo and F. De Rango, “Median-Krum: A Joint Distance-Statistical Based Byzantine-Robust Algorithm in Federated Learning,” in *Proceedings of the Int’l ACM Symposium on Mobility Management and Wireless Access*, New York, NY, USA: ACM, Oct. 2023, pp. 61–68. doi: 10.1145/3616390.3618283.
- [34] D. C. Nguyen *et al.*, “Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges,” *IEEE Internet Things J*, vol. 8, no. 16, pp. 12806–12825, Aug. 2021, doi: 10.1109/JIOT.2021.3072611.
- [35] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh, “The Limitations of Federated Learning in Sybil Settings Authors: ,” in *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, 2020.

- [36] N. M. Jebreel, J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "LFighter: Defending against the label-flipping attack in federated learning," *Neural Networks*, vol. 170, pp. 111–126, Feb. 2024, doi: 10.1016/j.neunet.2023.11.019.
- [37] Yanli Li, Huaming Chen, Wei Bao, Zhengmeng Xu, and Dong Yuan, "HONEST SCORE CLIENT SELECTION SCHEME: PREVENTING FEDERATED LEARNING LABEL FLIPPING ATTACKS IN NON-IID SCENARIOS," *Cryptography and Security*.
- [38] P. R. Ovi, A. Gangopadhyay, R. F. Erbacher, and C. Busart, "Secure Federated Training: Detecting Compromised Nodes and Identifying the Type of Attacks," in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, Dec. 2022, pp. 1115–1120. doi: 10.1109/ICMLA55696.2022.00183.
- [39] A. Gupta, T. Luo, M. V. Ngo, and S. K. Das, "Long-Short History of Gradients Is All You Need: Detecting Malicious and Unreliable Clients in Federated Learning," 2022, pp. 445–465. doi: 10.1007/978-3-031-17143-7\_22.
- [40] U. Tariq, S.-H. Wu, M. A. Mahmood, M. M. Woodworth, and F. Liou, "Effect of Pre-Heating on Residual Stresses and Deformation in Laser-Based Directed Energy Deposition Repair: A Comparative Analysis," *Materials*, vol. 17, no. 10, p. 2179, May 2024, doi: 10.3390/ma17102179.