machines

MDPI

Article

Federated Learning-Enabled Secure Multi-Modal Anomaly Detection for Wire Arc Additive Manufacturing

Mohammad Mahruf Mahdi [1], Md Abdul Goni Raju [2], Kyung-Chang Lee [3,*] and Duck Bong Kim [4,5,*]

[1] Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN 38505, USA; mmahdi42@tntech.edu
[2] Department of Mechanical Engineering, Tennessee Technological University, Cookeville, TN 38505, USA; mraju42@tntech.edu
[3] Department of Intelligent Robot Engineering, Pukyong National University, Pusan 48513, Republic of Korea
[4] Department of Manufacturing and Engineering Technology, Tennessee Technological University, Cookeville, TN 38505, USA
[5] School of Environmental, Civil, Agricultural, and Mechanical Engineering, University of Georgia, Athens, GA 30602, USA
* Correspondence: gclee@pknu.ac.kr (K.-C.L.); dbkim@uga.edu (D.B.K.)

Abstract

This paper presents a federated learning (FL) architecture tailored for anomaly detection in wire arc additive manufacturing (WAAM) that preserves data privacy while enabling secure and distributed model training across heterogeneous process units. WAAM's inherent process complexity, characterized by high-dimensional and asynchronous sensor streams, including current, voltage, travel speed, and visual bead profiles, necessitates a decentralized learning paradigm capable of handling non-identical client distributions without raw data pooling. To this end, the proposed framework integrates reversible data hiding in the encrypted domain (RDHE) for the secure embedding of sensor-derived features into weld images, enabling confidential parameter transmission and tamper-evident federation. Each client node employs a domain-specific long short-term memory (LSTM)-based classifier trained on locally curated time-series or vision-derived features, with model updates embedded and transmitted securely to a central aggregator. Three FL strategies, FedAvg, FedProx, and FedPer, are systematically evaluated against four robust aggregation techniques, including KRUM, Multi KRUM, and Trimmed Mean, across 100 communication rounds using eight non-independent and identically distributed (non-IID) WAAM clients. Experimental results reveal that FedPer coupled with Trimmed Mean delivers the optimal configuration, achieving maximum F1-score (0.912), area under the curve (AUC) (0.939), and client-wise generalization stability under both geometric and temporal noise. The proposed approach demonstrates near-lossless RDHE encoding (PSNR > 90 dB) and robust convergence across adversarial conditions. By embedding encrypted intelligence within weld imagery and tailoring FL to WAAM-specific signal variability, this study introduces a scalable, secure, and generalizable framework for process monitoring. These findings establish a baseline for federated anomaly detection in metal additive manufacturing, with implications for deploying privacy-preserving intelligence across smart manufacturing (SM) networks. The federated pipeline is backbone-agnostic. We instantiate LSTM clients because the sequences are short (five steps) and edge compute is limited in WAAM. The same pipeline can host Transformer/TCN encoders for longer horizons without changing the FL or security flow.

check for updates

Academic Editor: Gianni Campatelli

Received: 27 July 2025
Revised: 7 November 2025
Accepted: 14 November 2025
Published: 18 November 2025

Citation: Mahdi, M.M.; Raju, M.A.G.; Lee, K.-C.; Kim, D.B. Federated Learning-Enabled Secure Multi-Modal Anomaly Detection for Wire Arc Additive Manufacturing. *Machines* **2025**, *13*, 1063. https://doi.org/10.3390/machines13111063

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Keywords: wire arc additive manufacturing; federated learning; process monitoring; reversible data hiding; anomaly detection

1. Introduction

Additive Manufacturing (AM) has emerged as a transformative approach in modern manufacturing due to its ability to fabricate geometrically complex components with reduced material waste and customization across industries [1,2]. Among the various metal AM techniques, wire arc additive manufacturing (WAAM) has gained significant attention for its capability to produce large-scale metallic structures using an arc-welding-based deposition mechanism with comparatively low operational cost [3,4]. WAAM is particularly suited for structural applications in sectors like maritime engineering, defence, and aerospace due to its high deposition rate and compatibility with widely available wire feedstock [5–7]. However, the process involves complex thermal fluid relations, including abrupt thermal gradients, dynamic molten pool behavior, and arc-induced disturbance, which contribute to unpredictable variations in bead geometry and microstructural inhomogeneity [8,9]. These complexities require the deployment of advanced monitoring and control mechanisms to ensure geometric consistency, integrity, and overall process stability [10]. Addressing these challenges requires sensor-driven and data-centric strategies that can interpret high-dimensional and time-dependent process signatures in real-time and across diverse operational rules.

To address the inherent complexity of WAAM, recent developments in process monitoring have increasingly favored data-driven methodologies that utilize real-time sensor streams to infer thermal, geometric, and electrical process states [11]. These monitoring systems integrate high-frequency process signatures, producing asynchronous and high-dimensional datasets that are difficult to interpret through conventional rule-based methods [12]. Machine learning, particularly deep architectures such as convolutional and recurrent neural networks, has demonstrated the ability to learn nonlinear correlations and to detect subtle anomalies that precede physical defects from these heterogeneous signals [13,14]. However, implementing these models through centralized learning paradigms dictates the pooling of raw process data from different WAAM stations, which raises critical concerns around intellectual property exposure, operational confidentiality, and compliance with industrial data governance policies [15]. Moreover, process parameter distributions and sensor modalities differ significantly across WAAM installations, resulting in highly non-independent and identically distributed (non-IID) data that violates the assumptions of uniform training routines [16]. These constraints need a distributed and privacy-preserving learning framework capable of maintaining both local customization and global generalization.

Federated learning (FL) has emerged as a decentralized machine learning paradigm that enables multiple clients to collaboratively train a shared model without exposing their raw local datasets [17]. In a typical FL workflow, each client performs localized training on its process data and transmits model weight updates or gradients to a central server, which then performs global aggregation while maintaining data privacy [18]. This makes FL particularly well-suited for industrial environments, such as WAAM, where different workstations operate under distinct process dynamics but share the same underlying goal of real-time anomaly detection or quality assurance. Furthermore, FL architectures are inherently capable of addressing statistical heterogeneity across clients by supporting personalized models and employing aggregation strategies, which improve generalization across non-IID conditions [19]. Importantly, since FL retains sensitive process information

in local computational nodes, it aligns with the stringent data governance requirements of manufacturing ecosystems [20]. Despite these benefits, the application of FL to WAAM-specific anomaly detection remains largely unaddressed, especially in contexts involving encrypted visual-signal fusion and process-aware client specialization.

While FL has gained attention in Smart Manufacturing (SM) for applications such as predictive maintenance, fault classification, and process health monitoring in domains like laser powder bed fusion or injection molding, its potential in WAAM systems remains largely unexplored [20,21]. In particular, to our knowledge, no existing study has implemented a federated architecture designed explicitly for WAAM, which presents distinct challenges due to its high deposition rates and varied sensor configurations. Moreover, current FL implementations often focus on unimodal inputs and do not accommodate the secure fusion of electrical signatures, positional telemetry, and geometric features extracted from vision systems under a unified learning framework. These approaches typically assume trusted aggregators and overlook the risks posed by adversarial updates or tampered communications, limiting their applicability in sensitive industrial settings. Aforementioned critical gaps motivate the need for an FL system that is both WAAM-specific and resilient, enabling secure, multi-source anomaly detection across distributed process units.

To this extent, this study proposes an FL-based architecture for secure and distributed anomaly detection in WAAM. The architecture introduces client-specific models trained on process signals such as current, voltage, speed, position, and arc-related geometry, which are securely embedded into high dynamic range (HDR) weld images using reversible data hiding. It integrates local model customization with centralized aggregation strategies, including FedAvg, FedProx, and FedPer, to accommodate statistical heterogeneity across WAAM units. The proposed architecture is validated through a multi-client simulation framework that emulates eight distributed cells under replicated non-IID conditions. WAAM clients often have non-IID data due to different sensors, process regimes, and label rates. In our design, FedAvg is an IID reference, FedProx reduces client drift under non-IID data, and FedPer keeps a shared backbone with client-specific heads. Robust aggregators (KRUM, Multi KRUM, Trimmed Mean) reduce the effect of abnormal or skewed client updates. Our evaluation focuses on privacy-preserving learning and robustness to realistic non-IID variation in WAAM. The remainder of the paper is structured as follows: Section 2 presents a review of FL in SM, AM, and secure communication; Section 3 outlines the proposed framework with secure transmission; Section 4 details the system-level deployment and proof of concept; Section 5 analyzes the experimental results and compares aggregation strategies; and Section 6 concludes the study and suggests future research directions.

2. Related Works

In this section, we detail some recent literature concerning FL, where we investigate FL based on additive manufacturing in particular and SM in general. FL is also explored for its security and data encryption, and significant research gaps are outlined.

2.1. Federated Learning in Additive Manufacturing

FL has recently been explored in the context of AM due to its ability to enable collaborative model training across multiple sites without sharing sensitive data. This decentralized approach is particularly beneficial for AM, where diverse datasets are often distributed across various organizations, each with unique process parameters and machine settings. For instance, Mehta et al. implemented an FL-based semantic segmentation approach using a U-Net architecture for defect detection in laser powder bed fusion processes. Their method demonstrates that FL achieves defect detection performance comparable to centralized learning while preserving data privacy and significantly outperforms individual

learning, with improvements seen from data diversity and transfer learning for generalizability [22]. Again, at the enterprise level, Aggour et al. propose a federated multimodal Big Data storage and analytics platform that integrates diverse datasets from the additive manufacturing lifecycle, including material properties, design models, process parameters, sensor data, and inspection results, enabling scalable, unified access for advanced analytics and visualization to optimize manufacturing processes and accelerate technology maturation [23].

Moreover, Shi et al. developed a knowledge distillation-based information sharing (KD-IS) framework that enhances monitoring performance for data-poor units in decentralized manufacturing by leveraging distilled knowledge from data-rich units. Their method achieved comparable accuracy and F-score to models trained with six times more data, while reducing training time by 25% and effectively preserving data privacy [24]. On the other hand, Russell et al. detailed an approach combining self-supervised learning (SSL) with Barlow Twins and FL to improve fault detection in manufacturing. Their results show that integrating FL boosts accuracy from 67.6% to 73.7% for supervised models and from 82.4% to 83.7% for SSL models, demonstrating enhanced generalization and fault discriminability in decentralized settings [25].

The studies have applied FL to additive manufacturing processes, particularly in laser powder bed fusion (LPBF), fused deposition modeling (FDM), and extrusion-based setups. These works typically explore quality prediction, thermal monitoring, or control optimization while preserving data privacy across distributed manufacturing units. However, none of these implementations extend to WAAM, which operates under fundamentally different physical conditions involving high-temperature electric arcs, dynamic melt pool behavior, and continuously evolving deposition geometry. Moreover, the existing frameworks often rely on single-channel data inputs such as force feedback or thermal signatures, whereas WAAM requires the integration of multiple process parameters, including voltage, current, wire feed rate, arc length, speed, and bead geometry. These parameters vary in temporal resolution, demanding a more advanced learning approach. In addition, current federated models assume uniform architectures across clients and do not support local personalization, making them unsuitable for systems where each client observes different signal domains.

2.2. Federated Learning in Smart Manufacturing

FL strengthens predictive maintenance, anomaly detection, and process optimization while protecting data. SM systems with many networked sensors across sites suit decentralized training and analysis [26].

Huong et al. introduced FedeX, an explainable FL anomaly detector for industrial control that combines variational autoencoders, support vector data description, and explainable AI. On SWaT, it reached a Recall 1 and F1 score of 0.9857, beat 14 methods, and ran fast for real-time edge use [27]. Dib et al. used FL with local training and federated averaging for sheet metal forming defects, matching centralized neural network accuracy while preserving privacy [28]. Chen et al. built FMCMC DR for digital twin-based federated analytics, giving better global distribution estimates with accurate 50% and 95% contours and faster convergence than Metropolis–Hastings and random walk MCMC [29].

For IIoT, Kanagavelu et al. proposed a Two-Phase Multi-Party Communication enabled FL with a committee for private aggregation, integrated into an IIoT platform and outperforming peer-to-peer frameworks in accuracy and efficiency [30]. In addition, Gao et al. detailed RaFed, a resource allocation scheme that cut training latency by 29.9% through optimal device and resource choices in static wireless networks [31].

In Industry 4.0, Brik et al. built a federated deep learning monitor using Fog computing to predict disruptions from resource localization errors with low latency, high accuracy, and better QoS, tardiness, and makespan via Tabu search [32]. Kusiak et al. proposed the XRule algorithm and federated explainable AI to improve transparency and insight [33]. Putra et al. created an FL-enabled digital twin for 3D printing with a CNN fault detector that raised accuracy by 8% while keeping training times low and latency near 1026.16 milliseconds between printer and twin [34]. Additionally, Verma et al. offered an FL-based deep intrusion detector using a hybrid CNN + LSTM + MLP, reaching up to 99.447 percent accuracy and protecting privacy with Paillier encryption, surpassing leading methods and addressing FL attack risks [35].

Sun et al. combined FL and blockchain in the SP ED method for energy production and distribution, improving sustainability by 11.48%, flaw detection by 14.65%, and cutting modifications and detection time versus Data-Driven Sustainable Intelligent Manufacturing [36]. Yang et al. selected clients using parameter variation and graph theory, raising accuracy by 0.93% to 2.65% and reducing heterogeneity effects [37]. Zhang et al. proposed DetectPMFL using Cheon Kim Kim Song encryption with unreliable agent detection, and showed stronger robustness and accuracy on F MNIST and CASE WESTERN datasets [38].

Across SM, FL supports decentralized fault detection, health monitoring, and distributed control with privacy. Limits remain. Many systems use generic designs that ignore domain-specific behavior, seldom track round-wise metrics such as global loss or client divergence, and keep static aggregation despite changing conditions, signal quality, or complexity. Without outlier suppression or similarity-based filters, one corrupted client can skew the global model. Heavy reliance on trusted aggregators and the absence of tamper-proof audit trails further reduce assurance.

2.3. Federated Learning in Secure Communication

Security in FL is crucial for AM and SM, where privacy and integrity matter. FL decentralization brings both challenges and opportunities. Ranathunga et al. built a blockchain-based decentralized FL with a hierarchical aggregator network to handle low-quality updates, secured by additive homomorphic encryption and off-chain credibility checks in trusted execution environments, cutting convergence time and latency while raising accuracy and fairness in predictive maintenance and product inspection [39]. Kuo et al. created a privacy-preserving FL with fully homomorphic encryption that computes on encrypted data, protecting privacy in segregated ownership settings in SM and showing strong defense against cyber-attacks while maintaining accuracy in real-world studies [40].

Li et al. proposed a privacy-preserving and Byzantine robust scheme named PBFL for Industry 4.0, using agglomerative hierarchical clustering for robust aggregation and two-party computation to improve security and efficiency. It achieves major runtime reductions with accuracy preserved even with up to 49 percent malicious clients, resisting Byzantine attacks [41]. Zhang et al. presented a joint optimization for FL in IIoT that balances speed and cost by optimizing edge association, resource allocation, and transmit power, decomposed into three subproblems and solved with alternating optimization, improving learning and efficiency while managing the speed–cost tradeoff [42].

Secure FL is used in healthcare diagnostics, industrial IoT, and sensor monitoring, typically relying on encryption, differential privacy, or trusted execution environments for protected transmission and aggregation. Yet most methods do not embed security within the training pipeline through reversible data hiding or image-based parameter encoding, limiting integrity checks and tamper-evident learning in sensitive plants. Many aggregators assume balanced data and are fragile to adversarial updates under non-IID conditions

common in manufacturing. Without outlier suppression or similarity-based filtering, one corrupted client can skew the global model. Current frameworks also depend on trusted aggregators and lack tamper-proof audit trails that verify the source and structure of incoming updates.

3. Proposed Methodology

In order to address the limitation in the existing literature, the proposed methodology is designed to meet the critical need for secure learning across distributed WAAM setups, where each unit operates under unique conditions and generates heterogeneous process data. These datasets, which include time-dependent electrical signals, geometrical bead representations, and high-frame-rate visual sequences, cannot be pooled due to confidentiality and data ownership constraints. Therefore, this architecture adopts an FL framework that allows each WAAM client to independently train local models using its own process data while contributing to a shared anomaly detection model through periodic encrypted updates. The approach ensures that no raw data leaves the local boundary and that every site maintains control over its data assets. As shown in Figure 1, the architecture initiates with individual data acquisition and training at the client nodes, followed by secure transmission of encoded model parameters, iterative model fusion under privacy-preserving protocols, and redistribution of the aggregated model for continued local refinement.

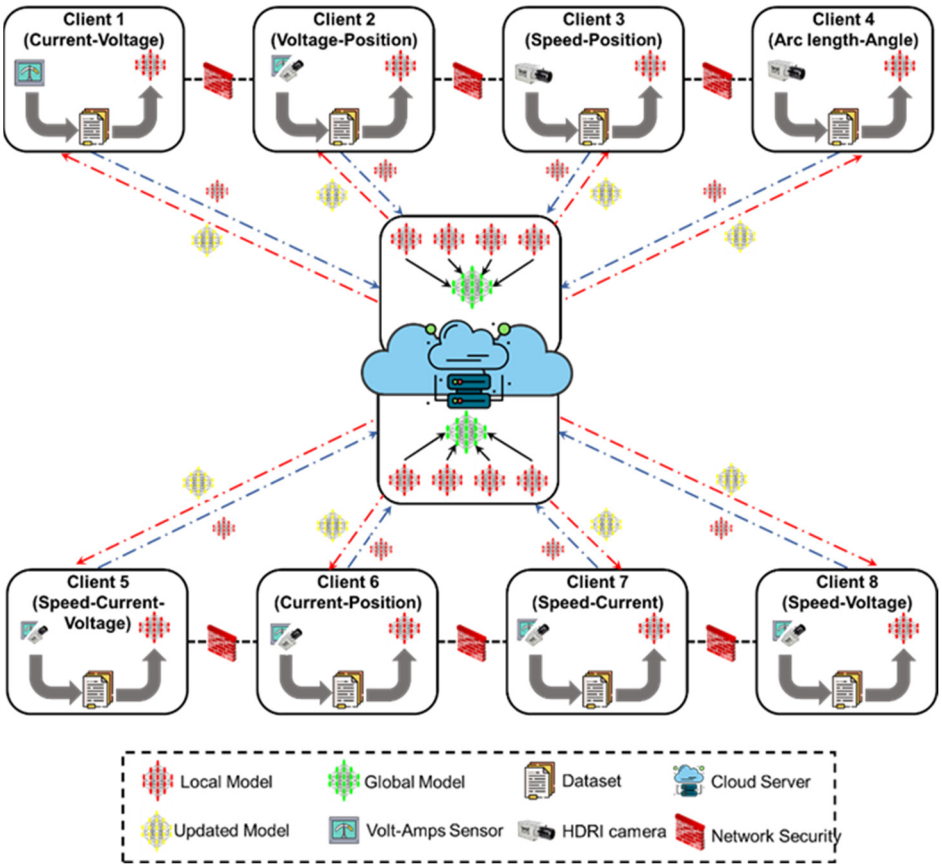


Figure 1. Proposed framework of the FL model.

- **Local Data Collection and Preprocessing**: Each WAAM client acquires structured and unstructured process data, which are filtered, normalized, and aligned temporally to construct feature matrices suitable for model training.
- **Local Model Training**: Clients train sensor-specific models on preprocessed data to learn localized anomaly patterns without transmitting raw information externally.

- **Model Aggregation via FL**: Encrypted local model parameters are securely transmitted and aggregated into a global model using privacy-preserving fusion techniques.
- **Global Model Update**: The aggregated model updates are redistributed to clients, enabling them to enhance their local inference capabilities based on collective knowledge.
- **Iteration and Model Convergence**: The training–aggregation–update cycle repeats multiple times until the model achieves convergence across all validation criteria.
- **Security and Privacy Mechanisms**: A dual-layered protection scheme combining reversible encryption and differential privacy ensures confidentiality and resistance to inference attacks.
- **Model Deployment and Inference**: Once converged, the global model is deployed locally at each client for real-time anomaly detection and process monitoring.

Architecture begins at the client: each WAAM unit is an autonomous, isolated training node with domain-specific sensors. Per physical configuration/instrumentation, clients collect data classes, high-frequency current–voltage and travel–speed signals, plus HDR imaging for bead geometry and arc characterization. Sources differ in sampling, statistics, signal integrity, and failure patterns. Each client runs a tailored curation pipeline: time-based resampling, filtering, and segmentation. For time-series, compute RMS, kurtosis, and spectral-entropy features over sliding windows; image streams undergo contrast enhancement, ROI isolation, and dimensionality reduction via pretrained encoders. Resulting feature matrices feed neural networks tailored to modality, convolutional encoders for visuals, LSTM or GRU blocks for sequential signals, so each client detects domain-specific anomalies while preserving federation heterogeneity.

After local training, each client encodes parameter updates for secure transmission; gradient tensors are differentially excited with calibrated Laplacian noise to hide sensitive training signatures while preserving fidelity, then embedded with reversible data hiding for lossless restoration after decoding. Encoded sets use integrity-preserving channels. At aggregation, encrypted updates are decoded, consistency checked, and fused into a global model via federated averaging weighted by client data contributions. The refined model returns for local updates and repeats until convergence from stable global loss and local validation. Synchronizing nonidentical clients requires a rigorous, sensor-aware client data processing framework described next.

3.1. Multi-Source Data Collection and Processing

WAAM yields high-frequency current/voltage, slower travel speed/wire feed rate, and vision bead/arc imagery, differing in sampling/resolution/statistics. Clients standardize inputs through a custom preprocessing pipeline and a unified schema for federated training that uses time-aligned signal sequences for current/voltage/speed/vibration/acoustic power, plus frame or patch tensors for images and thermals. Porting map sensors to this schema selects per-stream encoders (CNN for images, LSTM or TCN for time series, dual branch when both), and defines labels per local quality criteria. Signals are filtered/normalized/resampled on a common grid, windowed, then featurized with RMS/kurtosis/spectral entropy to capture steady-state and transient anomalies, yielding consistent/representative/time-aligned inputs. The federated protocol, secure updates using RDHE, channel protection, and aggregation, stay unchanged.

For clients with visual data, image frames are processed using grayscale conversion, contrast enhancement, and region-of-interest isolation to highlight weld features, such as bead width and arc boundaries. Pretrained convolutional encoders then extract fixed-length feature vectors, which are synchronized with structured sensor data via timestamp matching. The resulting fused feature matrices are semantically rich and temporally coherent, enabling effective anomaly learning. This dual-channel preparation, combining

signal-based and vision-based approaches, ensures that each client independently constructs a reliable training set tailored to its sensor configuration, laying the groundwork for stable federated learning across non-identical WAAM environments.

3.2. Federated Secure Channel

After local training, each WAAM client sends model updates without exposing sensitive data. Sharing parameters creates integrity risks that can compromise learning, namely model inversion that reconstructs training data from gradients, gradient leakage when sparse current or voltage signals reveal operational patterns, and poisoned updates that degrade the global model or distort anomaly boundaries. Uneven cross-site data further increases identifiability during aggregation. The transmission pipeline must block unauthorized access, preserve update fidelity, and prevent data inference or training interference. The system, therefore, uses a two-layer framework combining reversible encryption with privacy-preserving perturbation, which protects updates in transit and enables detection and correction of tampering before aggregation. These safeguards maintain trust and consistency in federated learning. In this work, transport relies on an encrypted channel and server-side aggregation; we add no noise to gradients or updates, and we do not claim differential privacy. RDHE provides at-rest protection for process parameters in image artifacts and is not applied to federated learning updates. Adversarial evaluations, including poisoning, inversion, and membership inference, and formal privacy budget analyses are outside our scope; see Section 5 for how these can be integrated in future deployments.

Each client secures model updates with a two-stage encoder combining differential privacy and reversible data hiding. First, gradients or weight tensors receive calibrated Laplacian noise that masks individual sample influence, preventing linkage to specific inputs or process states if intercepted. Second, noise-added tensors become image-like payloads, embedding encrypted tensors in the least significant pixel bits. Decoding restores parameters with no precision loss, preserving floating point resolution for aggregation. A watermark signature enables tamper detection and authenticity checks by clients and aggregators. The result is high-capacity, low-distortion embedding that is reversible and secure, blocking passive inference and active manipulation, and allowing only valid, untampered updates into aggregation.

Beyond content protection, communication must be secure and verifiable. Each client runs a network security framework that monitors outgoing and incoming parameter flows for anomalies. Packet inspection flags abnormal payload size, irregular send intervals, or unauthorized access attempts that may signal tampering or injection. Every update carries a cryptographic signature and a timestamp for arrival verification; signatures are validated before aggregation, so only authenticated updates affect the global model. Update histories are recorded in an append-only local registry, providing an audit trail for disputes or rollbacks and enabling provenance and client accountability. This layered scheme, combining noise-based privacy, reversible embedding, and network validation, keeps the federation reliable across distributed WAAM sites. With exchange secured, the next task is to tailor local models that learn from prepared streams while staying compatible with global aggregation. RDHE uses pixel-wise two-bit LSB substitution with table lookup decoding, dominated by inexpensive integer and bit operations suited to edge devices.

3.3. Distributional Assumptions and Shift Regime

Non-IID from differing sensors and features, binary anomaly labels with client-specific prevalence, and approximate stationarity during training. No raw data sharing; only encrypted updates via secure channel with secure aggregation and RDHE. FedAvg IID reference; FedProx proximal term to limit client divergence; FedPer shared backbone with

client-specific heads; KRUM, Multi KRUM, Trimmed Mean resist aberrant updates. The feature adapter and local encoder vary to match sensors, CNN for LPBF melt pool images, LSTM and TCN for CNC vibration; loop, privacy, and aggregation are unchanged.

3.4. Local Client Model Development

WAAM clients face distinct modalities, high-frequency time series such as current, voltage, and speed, and visual features such as bead geometry and arc profiles, so models are tailored. Visual streams use convolutional encoders; sequential streams use LSTM or GRU; dual modality fuses feature vectors, then applies fully connected layers. Apply dropout, batch normalization, and gradient clipping; monitor validation to stop when loss and accuracy stabilize. Models stay lightweight, FL compatible, and sensitive to each operational domain. We use an LSTM at each client because sequences have five steps, per client samples are modest, and devices have limited compute and memory; recurrent models train stably under FL and meet latency and memory limits. The framework is architecture agnostic; a Transformer or a TCN can replace the LSTM with no change to the FL protocol, secure aggregation, or RDHE. The client model is a single-layer LSTM with an input dimension 3 and hidden 256 with a one-neuron classification head. Parameters follow $4 \cdot h \cdot (d + h) + 8 \cdot h = 267,264$ for $d = 3$, $h = 256$, plus 257 for the head, total $267,521$. At FP32, this is about 1.07 MB including the head. With five-step sequences and batch 32, output activations use about 0.16 MB per batch; even with gate activations and optimizer states, training memory stays in single-digit megabytes. Per step compute is dominated by gate matrix multiplies $4 \cdot (d \cdot h + h \cdot h)$; with $d = 3$, $h = 256$ that is $265,216$ multiply adds per step; for five steps and a batch 32 that is about 42.4 million multiply adds, about 85 MFLOPs if one multiply and add equals two FLOPs, feasible on a CPU-only industrial PC. A practical client is an edge-class industrial PC with $\times$86 64 CPU, four to eight cores, eight to sixteen GB RAM, at least 100 megabits per second Ethernet or NIC, and at least 20 GB storage, running Linux or Windows. Analytics and training run asynchronously from robot and power control; actuator loops remain on OEM controllers; the client schedules non-real-time workloads per layer or part or off shift.

Clients map outputs to a shared representation with dimensionality-matching layers for secure federation. Before sending, parameters receive Laplacian noise and are embedded as image-like payloads via reversible data hiding, preserving precision and security. Cryptographic signatures and local logs ensure authenticity and traceability, and secure channels carry updates for consistent, interpretable server aggregation. This localized interoperable design preserves privacy, architecture flexibility, and robustness in WAAM-specific anomaly detection while ensuring each client contributes.

3.5. Global Server Model Aggregation

Aggregating heterogeneous WAAM clients is difficult because data types, sample sizes, and model dynamics differ. Naive averaging can favor data-rich or dominant feature clients and weaken generalization. The server applies selective aggregation, measuring each update alignment with the federation's latent space using similarity metrics such as gradient direction and distribution statistics. Outliers or inconsistent updates are downweighted or removed. Encoded updates are decoded from reversible data hiding, authenticated with cryptographic signatures, and projected into a shared latent space to reconcile architectural differences. Weights reflect historical reliability and consistency, and fusion favors diversity and stability while remaining consistent and interpretable. Personalization layers support local adaptation before the refined global model returns to clients.

During training, the server tracks validation and loss stabilization across rounds. If progress plateaus, aggregation stops to save resources and reduce overfitting. Commu-

nication cadence adapts to client availability and data shifts. Explainability tools such as feature attribution improve transparency and operator trust. Final global models are redistributed securely with privacy-preserving guarantees. Together, these steps create a secure, adaptive, and generalizable federated cycle that enables effective anomaly detection across non-IID WAAM sites.

Temporal operation and detection latency. Inference runs on a five-step sliding window; each new sample or frame advances the window and yields a local anomaly score. Local training is scheduled rather than continuous, either time-driven at fixed intervals or off shift, or batch-driven after N windows or at layer or part boundaries. Training runs at low priority on the client PC and stays asynchronous from machine control. Let Δt be the synchronization period set by the slowest stream, camera frame, or signal sample. With window length $W = 5$, the earliest decision latency $t_{detect} \approx W \times \Delta t + t_{model} + t_{decision}$. The preprocessing time $t_{preproc}$ scales linearly with pixels in a grayscale, then threshold, then contour path. The model time t_{model} per window is about 1.33 MAdds, a few mega FLOPs, and *t_decision* is a light threshold step. Analytic forecasts, not measured, give $W \times \Delta t$ near 0.33 s at 15 fps with Δt about 67 ms, 0.17 s at 30 fps, and 0.08 s at 60 fps. With small CPU overheads from preprocessing and the model, end-to-end latency is about 0.35 s, 0.20 s, and 0.10 s. Detection is local and independent of federated communication.

In the vision path from grayscale to fixed threshold to contour and moments, two-dimensional noise affects binarization only near the threshold τ. With additive pixel noise ε of variance σ^2, only pixels with intensity in a narrow band around τ can flip, and the flip fraction is bounded by the local mass of the gray level histogram near τ. Downstream geometric features such as area, centroid, and moments averaged over N foreground pixels, so perturbations shrink on the order of the $1/\sqrt{N}$. For time series channels such as current, voltage, and speed, a five-step window reduces the variance of zero-mean perturbations from σ^2 to roughly $\frac{\sigma^2}{W}$ with $W = 5$, and the classifier error increase is set by the effective margin relative to the windowed noise. During deployment, standardization and light robust smoothing, such as median or Hampel filtering, can be applied without changing the federated protocol.

4. System Development Architecture

To validate the FL framework in Section 3, we built a scaled implementation as a system-level proof of concept. It demonstrates an end-to-end secure, distributed anomaly detection pipeline for WAAM. All components, from sensor-based acquisition and preprocessing to reversible data embedding, client-specific training, and federated coordination, were built and tested with real process data under controlled deposition. The setup simulates an eight-client federation, each representing distinct process signatures and sensor capabilities found in practice. Recreating the method sequence experimentally validates the interoperability of each module. The following describes the pipeline in order, starting with synchronized data collection and temporal alignment, then reversible data hiding, then per client model configuration and training, and finally the design and operation of the FL framework.

4.1. Data Collection and Preprocessing

Effective federated training needs a well-structured, synchronized dataset for convergence across nodes. We used a GTAW-based WAAM system to record electrical signals and HDR video of deposition, then applied computer vision to extract torch speed, feed angle, and arc length. All streams were timestamped and aligned for LSTM-based temporal modeling. The cell used a six-axis Fanuc ArcMate 120iC robot, with an R 30iA controller, FANUC Corporation, Oshino-mura, Yamanashi, Japan and a Miller Dynasty 400 GTAW power source, Miller Electric Mfg. LLC,

Dynasty 400 GTAW (TIG) Welding Power Source, Appleton, WI, USA for precise parameter control [43]. A Weldvis HDR camera captured the weld pool and feed wire under varied lighting. Low-carbon steel was the feedstock, with consistent wire feeding across trials. Two controlled trials created binary labels, one with stable settings of 60 A, 20 cm/min travel, and 160 cm per minute feed, and one with altered settings of 140 A, 40 cm/min, and 180 cm/min to induce defects. Sensors recorded continuously and synchronously, yielding temporally and spatially rich inputs for federated LSTM anomaly detection. Each participant runs on an edge-class industrial PC near the WAAM cell with an $\times 86\ 64$ CPU, four to eight cores, eight to sixteen gigabytes of RAM, and a CPU only. This matches the client compute profile in Section 3.3 and the per-round communication budget in Section 4.4. The named brands identify sensor streams, while client hardware denotes the computing unit for preprocessing, local training, and secure updates. Actuation and safety-critical control remain on the FANUC and Miller controllers, and the client PC performs analytics and training asynchronously.

Electrical current and voltage were captured with a Miller Insight ArcAgent Auto sensor sampling every 0.10 s, giving sufficient granularity for deposition dynamics. Arc instability at ignition and extinction introduces inconsistent entries, so initial and terminal segments were cleaned by removing near-zero frames with a binary threshold. These operations run in time linear to the number of pixels using simple integer arithmetic, are designed for CPU-only execution at modest frame rates, avoid heavy convolutional inference, and keep latency low on typical edge PCs. Timestamps from the current and voltage sensors served as reference markers to synchronize all other sensors, including video-based speed and geometric parameters. Preprocessing also removed anomalous points from sensor latency and transient disruptions such as rapid parameter changes or signal dropout. Outliers were found with statistical filters and replaced by interpolation to preserve temporal continuity. These steps mitigate noise-driven irregularities that could harm generalization. The final voltage and current time series, denoised and aligned with video timestamps, provide reliable temporally coherent input to each client LSTM classifier.

Normal and abnormal deposition sequences under controlled conditions appear in Figure 2a. Each video stream was split into frames, and the FPS was extracted to model time dependence. After grayscale conversion to reduce computation, binary thresholding isolated high-intensity arc regions. For each frame, contours were found, and the largest contour, typically the arc plasma, was chosen. Spatial moments gave the centroid $(x_c,\ y_c)$ per frame, representing the torch position. Welding speed v came from the Euclidean displacement of consecutive centroids, scaled to physical units by FPS and a spatial calibration factor $\alpha = 0.1$ mm/pixel, as in Equation (1).

$$v = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \times \text{FPS} \times \alpha \tag{1}$$

where $(x_t,\ y_t)$ and $(x_{t+1},\ y_{t+1})$ are centroids at consecutive steps, shown in Algorithm S1. Speeds pair with centroids, timestamped at 0.10 s and mapped to the current voltage timestamp axis for alignment. Algorithm S2 aligns all streams to a common timeline and outputs masks and metadata for missing entries.

Finally, HDR video was processed with computer vision to extract part parameters and weld geometry, Figure 2b, detecting arc length and feed angle. Arc length L_{arc} is the vertical spread of arc plasma: frames convert to grayscale, threshold at intensity 190 isolates arc, then first and last rows with nonzero pixels define extent; their difference gives arc height in pixels. With vertical resolution 100 pixels per millimeter, Equation (2) maps height to arc length.

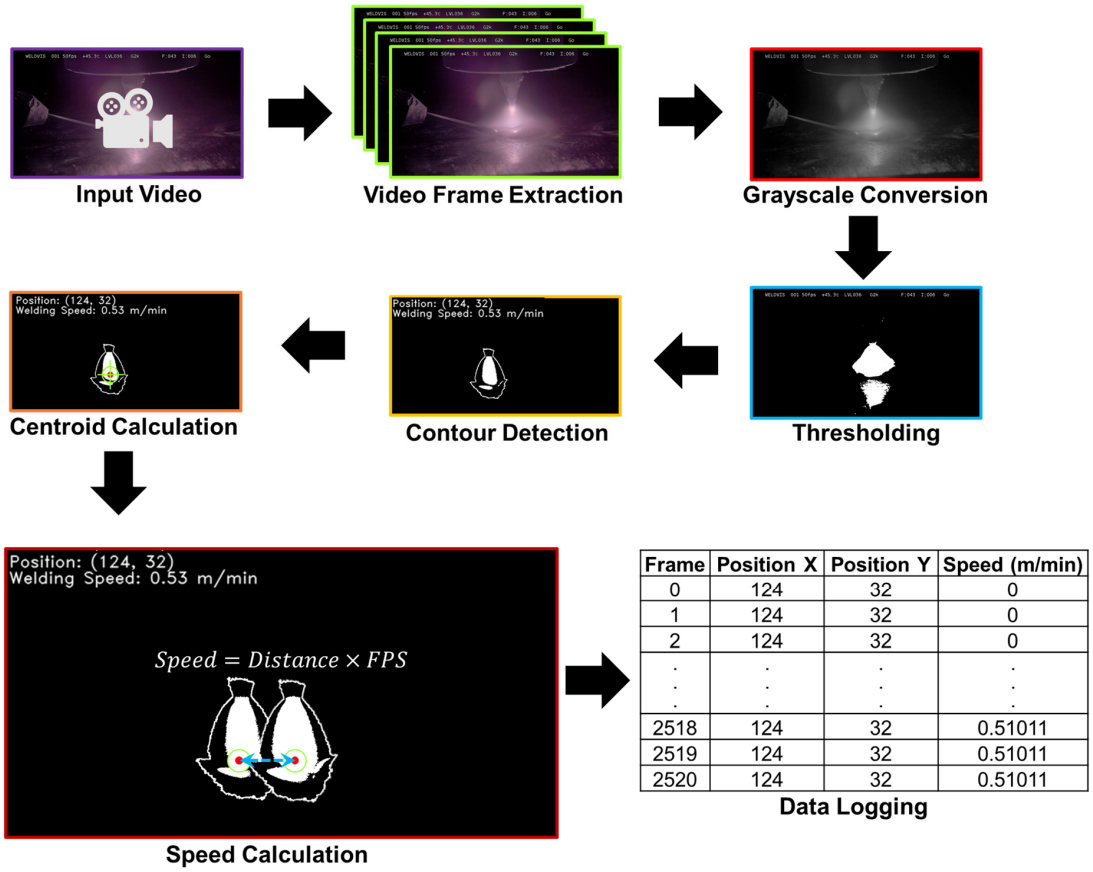
$$L_{arc} = \frac{y_{max} - y_{min}}{100} \tag{2}$$

where y_{max} and y_{min} are the bottommost and topmost arc positions. For the feed angle, crop ROI around the feed wire. Preprocess with grayscale inversion, contrast normalization, and multi-stage Gaussian blur. Run Canny edge detection [44], then a probabilistic Hough line transform [45] to detect straight segments. Compute each segment angle from its slope and average to obtain θ_{feed} in Equation (3).

$$\theta_{feed} = \left| \frac{1}{N} \sum_{i=1}^{N} \tan^{-1} \left(\frac{y_2^i - y_1^i}{x_2^i - x_1^i} \right) \right| \tag{3}$$

Each value of L_{arc} and θ_{feed} was timestamped and interpolated to match the current-voltage time in Section 4.2, ensuring multi-data temporal alignment. Heuristics of determining L_{arc} and θ_{feed} was shown in Algorithm S3.

After synchronizing and preprocessing sensor-derived and vision-based features, we built labeled datasets for supervised training across federated clients. Two WAAM experts independently reviewed each deposition video and identified normal and abnormal sequences based solely on bead morphology. Annotations were aligned to exact frame timestamps and matched with multimodal features. The labeled set combined synchronized current and voltage signals from sensors with centroid position and welding speed, extracted through a frame-wise computer vision pipeline. Every instance received a binary label, zero for normal and one for abnormal. Geometric parameters, arc length, and feed angle were also time-matched and labeled in the same way. These two labeled sources were distributed to eight client models with distinct input combinations as detailed in Section 4.3. To reflect non-uniform industrial availability, clients received different sample counts, from 216 in Client 1 to 1692 in Client 3, per Table 1, and for all clients, a consistent 75:25 train–test partitioning was applied across all clients.



(**a**)

Figure 2. *Cont.*

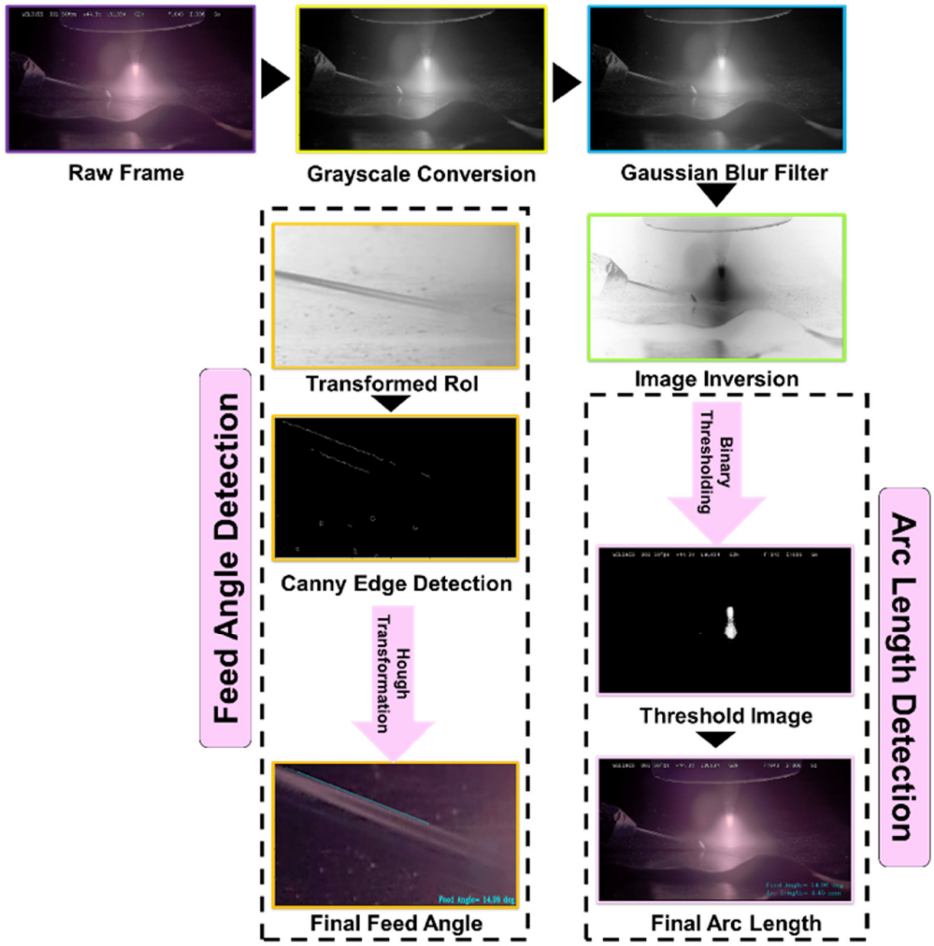


Figure 2. Detection process for (**a**) speed and centroid position and (**b**) feed angle and arc length.

Table 1. Train and test data distribution across the clients.

Client	Train Data	Test Data
1	216	72
2	348	116
3	1692	564
4	300	100
5	348	116
6	348	116
7	348	116
8	348	116

4.2. Reversible Data Hiding

Figure 3 shows the RDHE module, which embeds high-resolution WAAM parameters, current, voltage, travel speed, arc length, and feed angle into visual process images in the encrypted domain without irreversible distortion, following [46]. A weld image acquired during deposition is partitioned into an n × m grid; each sub-block is split into RGB planes $I_R, I_G, I_B \subset \mathbb{z}^{H \times W}$ that serve as embedding substrates. Channels are scanned in pixel-wise raster order, and low gradient regions found via Sobel and Laplacian thresholds mark candidate zones, avoiding perceptual loss at high intensity edges near the weld pool. The payload P serializes these parameters in 8-bit ASCII, then forms a bit stream

$b = b_1, \ldots, b_L$ with the fixed terminator "=====" for stream end. Embedding alters the two least significant bits of selected pixels $I_c(i,j)$ using Equation (4).

$$I_c^*(i,j) = \left\lfloor \frac{I_c(i,j)}{4} \right\rfloor \cdot 4 + \text{bin}^{-1}(b_{2k-1}, b_{2k}) \tag{4}$$

where $c \in \{R, G, B\}$ and bin^{-1} denotes the inverse binary-to-decimal mapping. This substitution yields the encoded image I^*, which visually preserves the structural texture and colorimetry of the original image I while encapsulating process intelligence at a pixel level.

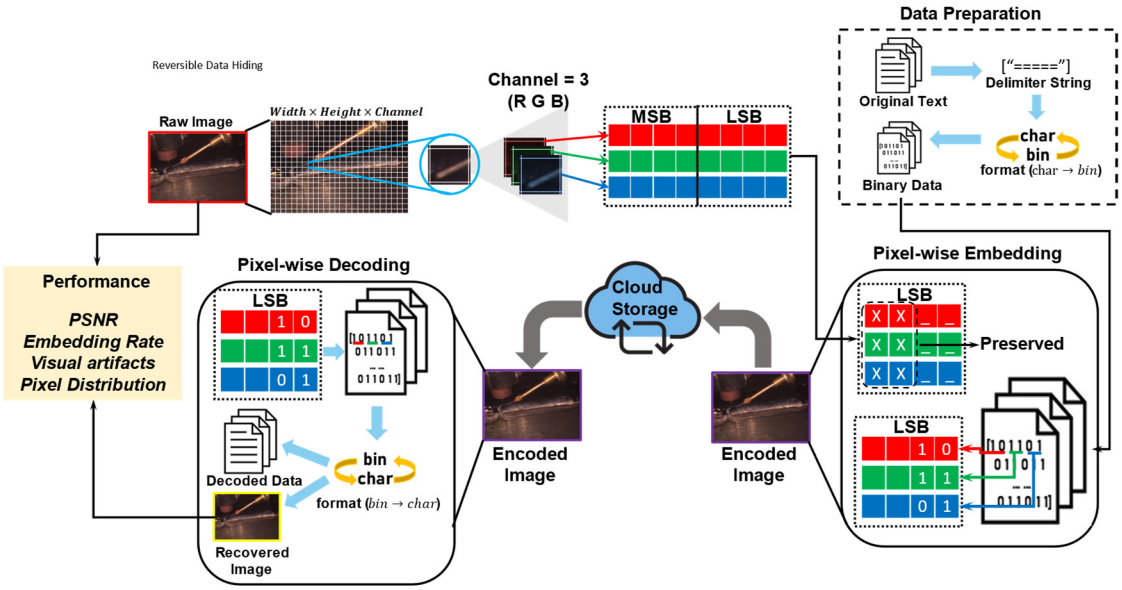


Figure 3. Reversible data hiding in the encrypted domain scheme.

Before storage or optional telemetry, a verification stage checks peak signal to noise ratio PSNR and mean squared error MSE between I and I^* to keep embedding perturbation below a perceptual threshold. At the same time, a digest of the original image is embedded in $\mathcal{P}$ for post hoc integrity checks during decoding. The mechanism supports synchronous and asynchronous client-server retrieval and integrates with edge-deployed WAAM control and cloud-aggregated analytics. Decoding retrieves the stored image with metadata, then rescans using the spatial and channel-wise offset map to extract embedded two-bit sequences from $I_c^*(i,j)$ using Equation (5).

$$(b_{2k-1}, b_{2k}) = \text{lsb}_2(I_c^*(i,j)) \tag{5}$$

We parse aggregated bitstream until terminator, then deserialize to recover the parameter payload. Simultaneously restore the image by reconstructing higher order bits and reinserting LSBs from the reversible buffer per Equation (6).

$$I_c(i,j) = \left\lfloor \frac{I_c^*(i,j)}{4} \right\rfloor \cdot 4 + R_c(i,j) \tag{6}$$

where $R_c(i,j)$, denotes the retained LSB snapshot prior to embedding. This guarantees exact recovery of I, and ensures that all embedded WAAM parameters are faithfully restored.

RDHE in this study is a reversible two-bit LSB scheme for confidentiality and integrity at rest; it was not evaluated against active steganalysis or adversarial tampering, so no robustness is claimed. If needed, RDH variants with histogram shifting and difference

expansion, plus cryptographic authentication of payload and container, can add steganalysis resistance and authenticated provenance without changing the FL pipeline, though outside our scope. The pipeline supports modular deployment and real-time streaming for continuous in situ encoding on edge WAAM systems and decouples encoding from decoding for scalable federation-wide interaction. Encoded images serve as visual objects and secure telemetry, integrate with FL, and ensure WAAM data are embedded, transported, and recovered without loss of confidentiality or signal integrity. RDHE images are not used in federated rounds because the FL uplink carries only the serialized, encrypted update $\Delta\theta$, about 1.1 MB FP32 base LSTM; a lossless 1080p RDHE image is about 0.5 to 2 MB for grayscale or 1 to 6 MB for RGB, so images remain at rest and optional telemetry.

4.3. Client Models Development

Each WAAM federated client trains a local sequence-based classifier with a shared backbone consisting of an LSTM encoder and a linear head. A client is a federated participant; in deployment, it maps to a machine, cell, or site; our experiments emulate eight participants with distinct data silos and feature groupings as in Table 2. The model LSTMClassifier handles fixed-length sequences with input feature dimension 3 and hidden size 256 per Algorithm S4. The recurrent layer captures temporal dependencies in each client sensor stream. The final time step hidden state goes to a fully connected layer with one output neuron, the logit for normal versus abnormal. Inputs have shape $[N, 5, 3]$ with N sequences, length 5 , and three features per client. Chronology is preserved by overlapping windows via a sliding frame that teaches short-term variations and anomalous transitions. Structure is identical across clients, while feature spaces differ. During federated training, only the shared LSTM parameters are uploaded to the server; the final fully connected layer stays local when using personalization such as FedPer. This preserves each client's decision boundary while contributing to a shared temporal encoder.

Table 2. Mapping between federated clients, signals, and sensor origins (emulated participants).

Client	Signals	Sensor/Origin
01	Current, Voltage, Time	Miller (Current, Voltage); System clock (Time)
02	Voltage, PosX, PosY	Miller (Voltage); FANUC controller (PosX, PosY)
03	Speed, PosX, PosY	FANUC controller (Speed, PosX, PosY)
04	Time, Arc Length, Feed Angle	System clock (Time); Vision (Arc Length, Feed Angle)
05	Speed, Current, Voltage	FANUC (Speed); Miller (Current, Voltage)
06	Time, Speed, Current	System clock (Time); FANUC (Speed); Miller (Current)
07	Time, Speed, Voltage	System clock (Time); FANUC (Speed); Miller (Voltage)
08	Current, PosX, PosY	Miller (Current); FANUC (PosX, PosY)

Eight FL clients differ by input features, each giving a distinct sensor view of WAAM to mirror real-world multi-enterprise settings. Clients 01 to 04 form a first diversity cluster. Client 01 uses average voltage, average current, and a timestamp normalized to $[0, 1]$, emphasizing electrical dynamics while avoiding scale dominance. Client 02 uses arc centroids x and y plus arc voltage, linking geometry with electrical behavior to capture spatiotemporal correlations. Client 03 combines welding speed with centroid coordinates, encoding motion-driven changes and spatial drift that signal torch trajectory instability. Client 04 is geometric, using arc length, feed angle, and timestamp from HDR vision to capture bead structure and wire orientation, and applies standard scaling to arc length and feed angle so both influence gradients equally. For all clients, data from five-step overlapping sequences via sliding windows; the i^{th} sequence covers time steps $[i, i+1, \ldots, i+4]$, with the label from the final frame, so models learn temporal evolution and local transitions that indicate emerging defects.

Clients 05 to 08 expand federation diversity by mixing electrical and kinematic features, giving complementary anomaly views across enterprises. Client 05 uses speed, current, and voltage to capture joint motion and electrical load, useful for disruptions from arc instability or inconsistent travel. Client 06 uses timestamp, speed, and current to embed temporal progression with measurements, modeling time drift, and time-aligned degradation. Client 07 is similar but replaces current with voltage, capturing high-frequency power delivery changes relative to speed and timestamp. Client 08 pairs current with $centroid_x$ and $centroid_y$, fusing electrical energy with a localized arc trajectory. All feature sequences are sampled every 0.10 s and interpolated as needed to match reference timestamps from the current and voltage stream, ensuring multimodal synchronization. Sequences have shape $[N, 5, 3]$, preserving order and spacing for LSTM-based temporal learning. Data loaders use a batch size of 32 , keeping mini-batch stochasticity and consistent time structure. Timestamp alignment gives each client a harmonized temporal view despite differing inputs. This deliberate heterogeneity across 05 to 08 increases global robustness, encoding anomaly cues from spatial distortion to electrical jitter, and strengthens generalization of the federated LSTM encoder.

Algorithm S5 details non-IID partitioning to emulate heterogeneous clients across quantity skew, label skew, feature view differences, and optional temporal strata. Low compute clients use partial participation and fewer local epochs, while higher capacity clients run the full schedule; the FL loop tolerates this mix. During local training, each client runs a full optimization cycle on its sequence data with a recurrent classifier as in Algorithm S6. The forward pass feeds each sequence to an LSTM to extract temporal features, then a fully connected head outputs a single logit for binary classification. Loss uses binary cross-entropy that combines sigmoid and cross-entropy in a numerically stable form. All clients use Adam with a learning rate of 0.001, and clip gradient norms to 1.0 to avoid exploding gradients. With FedProx, a proximal term defined as $\frac{\mu}{2}|w - w_{\text{global}}|^2$, where $\mu = 0.01$, keeping updates near the global model and reducing divergence from non-IID data.

After training, only LSTM encoder weights are shared with the server, while the classifier head stays local under FedPer to keep client-specific decision boundaries. Evaluation thresholds sigmoid outputs at 0.5 and computes accuracy, precision, recall, F1 score, and ROC AUC as in Algorithm S7. Each client logs its metrics and confusion matrix every round for detailed local analysis, and these logs feed the aggregated global metrics on the server. Together, this pipeline builds a robust global temporal encoder while preserving architectural consistency and data privacy, essential for secure distributed WAAM anomaly detection.

4.4. Federated Learning Approach

In FL, each WAAM client trains locally on its own time series modalities and sends updated LSTM weights to a central server, which aggregates a global model and returns it for the next round. The eight clients each handle a distinct mix of timestamped current, voltage, centroid position, travel speed, feed angle, and arc length, each with its own temporal dynamics. Before training, signals are segmented into overlapping five-step sequences that feed a client-specific LSTM classifier. This horizon matches short-term WAAM dynamics; when longer horizons or richer cross-sensor interactions are needed, the same pipeline can use attention-based encoders as drop-in replacements. In our experiments, the eight clients run as separate participants; in a plant, they map to machines or cells connected over a secure channel. The LSTM plus a fully connected layer maps sequences to binary labels for normal and anomalous deposition. Training uses binary cross entropy with logits, the

Adam optimizer, and gradient clipping; temporal correlations in the hidden states reveal deviations across consecutive steps.

For other processes, we follow a fixed recipe. Define clients at the cell or machine level; map sensors to the unified schema in Section 3.1; choose a compatible encoder per stream, such as CNN for images and LSTM or TCN for time series; keep the same RDHE path and FL round structure; and select a robust aggregator to match non-IID severity, such as Trimmed Mean or Multi KRUM. RDHE images do not travel in federated rounds; the uplink carries only the serialized encrypted update, while images serve at rest protection and optional telemetry. No changes are required to server orchestration, secure aggregation, or round scheduling. Table 3 gives a portability guide. For the personalization ablation, we hold the aggregator fixed, for example, Trimmed Mean, and compare FedPer with FedAvg and FedProx to isolate personalization under non-IID. Across experiments, FedAvg and FedProx act as industrial baselines and FedPer as a representative personalization strategy, each paired with robust aggregators described below.

Table 3. Portability guide across processes.

Process	Typical Client Unit	Common Signals	Local Encoder	Label Paradigm	Common Non-IID Pattern	Robust-Aggregator Guideline
LPBF	Machine/site	melt-pool images, photodiode, layer temperature	CNN (± LSTM for traces)	defect/no-defect per layer	material lot and geometry	Trimmed Mean
FFF/FDM	Printer/site	extruder temp, flow rate, motor current	LSTM/TCN	under/over-extrusion	filament type and nozzle wear	Trimmed Mean
CNC	Machine	spindle current, vibration	LSTM/TCN	chatter/out-of-tool	tool wear and part type	Multi KRUM
Ultrasonic welding	Station	acoustic amplitude, power	LSTM/TCN	bond failure	stack-up and amplitude set-points	Trimmed Mean

After local training, each client uploads only the base LSTM weights; the fully connected classifier stays local, especially under FedPer, which adapts decision boundaries to local heterogeneity by keeping the final layer private. The server gathers these weights from all clients and builds a global model using a defined federated optimizer. In this setup, the uplink per round is about 1.1 MB at FP32 for LSTM weights only, and the downlink is similar, keeping client-to-server transfer modest for edge deployment.

We evaluate three optimization strategies. FedAvg, the baseline in Algorithm S8, computes a weighted mean of client models using each client's local sample count as weight; it assumes IID data and benchmarks homogeneous cases. FedProx in Algorithm S9 adds a proximal regularizer with $\mu = 0.01$ to the local objective, penalizing drift from the global model and improving stability under non-identically distributed data by discouraging large local changes. FedPer in Algorithm S10 targets fundamentally distinct client distributions by keeping a shared encoder while enabling local adaptation through private classification layers.

Rounds start on a periodic schedule T or after K new batches, with T and K set by the integrator. $t_{round} \approx t_{local_{train}} + t_{uplink} + t_{server_{agg}} + t_{downlink}$. Under personalization, the uplink is about 1.1 MB of FP32 LSTM weights, and the downlink is similar. Transfer times are about 0.9 s at 10 Mbps, 0.09 at 100 Mbps, and 0.009 at 1 Gbps. The protocol supports partial participation and variable local epochs, enabling lower-capacity clients.

Let K be the number of clients scheduled in a round, and let $\rho \in [0, 1)$ denote the per-client packet-loss probability on the uplink. The expected number of received updates is $\mathbb{E}[\widetilde{K}] = K(1 - \rho)$. Under random missingness, the aggregated update remains unbiased, while its variance scales approximately as $\sigma^2/\widetilde{K}$, implying a convergence slowdown of order

$1/\sqrt{\widetilde{K}}$. Robust aggregators (Trimmed Mean, KRUM/Multi KRUM) continue to operate with fewer clients, but require a quorum policy: the trimming/Byzantine budget f should be set relative to the actually received updates in a round. In practice, we recommend adapting f to the observed $\widetilde{K}$ and allowing partial participation so that training proceeds even when some uplinks drop. The entire process is summarized in Table 4.

Table 4. Robustness envelope and mitigation knobs.

Stressor	Primary Effect	Analytic Scaling (Qualitative)	Mitigation Knob
Uplink packet loss ρ	Fewer updates per round; slower convergence	$Variance(\propto 1/\widetilde{K})$, $\left(\widetilde{K} = K(1-\rho)\right)$	Partial participation; adapt trim f to $\widetilde{K}$
Image noise (vision path)	Threshold flips near τ	Flip mass confined to narrow band around τ; feature perturbations $\sim 1/\sqrt{N}$	Threshold selection; standardization; optional median/Hampel
Signal noise (time series)	Higher variance in windows	Windowed variance $\approx \sigma^2/W$	Keep $W = 5$; standardization; optional robust smoothing

To defend against untrustworthy or faulty clients, we trigger resilient aggregation after a fixed training round threshold. We use three strategies: KRUM, Multi KRUM, and Trimmed Mean, each limiting the effect of outliers and poisoned updates. KRUM Algorithm S11 computes pairwise Euclidean distances among client models and chooses the update closest to its neighbors, discarding aberrant ones. Multi KRUM Algorithm S12 selects several near majority models and averages them, improving resilience and representation. Trimmed Mean Algorithm S13 sorts each weight coordinate, removes extremes, then averages, reducing adversarial noise and anomalous gradients. These methods engage after early rounds to preserve unconstrained learning diversity. We evaluate twelve configurations by pairing FedAvg, FedProx, and FedPer with Vanilla, KRUM, Multi KRUM, and Trimmed Mean. The protocol supports heterogeneous clients through partial participation and variable local epochs, so modest devices can contribute without strict uniformity.

We then evaluate a 3 by 4 design with strategies FedAvg, FedProx, and FedPer crossed with aggregators Vanilla, KRUM, Multi KRUM, and Trimmed Mean. This separates routine heterogeneity handling, where FedProx limits drift and FedPer enables local adaptation, from defenses against abnormal or skewed updates provided by Multi KRUM and Trimmed Mean. Gains under Multi KRUM and Trimmed Mean indicate robustness to skewed client updates rather than an IID assumption. The server coordinates rounds, manages updates, and tracks convergence with Algorithms S14–S16. Each round, it initializes or refreshes the global model, sends parameters to selected clients, receives updates, and computes the L2 norm gap between successive global parameters to quantify progress and stability. It aggregates loss, accuracy, precision, recall, F1 score, and ROC AUC using sample weighted averages, saves global metrics for reproducibility, and stores client metrics and confusion matrices for diagnostics. The logs enable visualizations of round-wise accuracy, loss curves, and convergence paths, supporting transparent comparison across strategies and clients. This modular FL framework addresses privacy, heterogeneity, and adversarial risk while enabling benchmarking across configurations.

5. Results and Discussion

We evaluate the WAAM federated framework with controlled experiments to test strategy personalization and aggregation robustness. Three strategies are compared: FedAvg, FedPer, and FedProx, each over one hundred rounds. RDHE lies outside analytics, so training and inference use original frames and signals, and accuracy is unchanged. Its contribution is at rest security by reversible embedding for storage or optional telemetry. With a two-bit LSB scheme, pixel change is within three levels, and only pixels within three of a decision threshold can flip; the flip fraction is $\pi_{(\text{flip})} \leq \sum_{k=-3}^{+3} p(\tau + k)$. Table 5 summarizes

the threat surface; RDHE strengthens at rest confidentiality and integrity, while in transit, privacy of model updates comes from the encrypted channel and server-side aggregation.

Table 5. RDHE ablation (security scope) relative to the FL channel.

Attack Surface/Requirement	With RDHE (This Work)	Without RDHE	Covered by FL Secure Channel/Agg.?
At-rest confidentiality of images + proc. params	Yes (payload embedded; reversible)	No (plain files at rest)	N/A
At-rest integrity/tamper detectability	Yes (recovery check consistency)	Reduced	N/A
In-transit confidentiality of model updates	N/A (RDHE not used on FL link)	N/A	Yes (encrypted channel + aggregation)
Gradient/privacy leakage of updates	Not addressed by RDHE	Not addressed by RDHE	Out of scope here

Holding the aggregator fixed, FedPer versus FedAvg versus FedProx isolates personalization. Trends show personalization stabilizes client decision boundaries and improves robustness under label or feature skew while keeping the same communication footprint by updating only the base LSTM. FedProx limits drift with a proximal term but does not customize heads; FedAvg serves as the non-regularized reference, so FedPer gains arise from per-client heads, not communication or aggregation changes. For every strategy we test Vanilla, KRUM, Multi KRUM, and Trimmed Mean, yielding twelve configurations evaluated with global and client metrics to assess convergence, stability, fairness, and consistency across sources. Gains with FedProx under Trimmed Mean match drift control benefits for non-IID clients, and FedPer captures client-specific effects without altering communication. These trends are driven mainly by strategy and aggregator behavior under non-IID participation rather than WAAM-specific signals. The approach should transfer to other processes by replacing the local encoder and feature adapter while keeping the same FL and aggregation stack; for example, the robustness of Trimmed Mean and Multi KRUM as client skew grows. Our empirical scope is WAAM, with a porting recipe in Section 4.4. The strongest F1 and accuracy with FedPer and Trimmed Mean suggest the LSTM captured key patterns at a five-step horizon while staying stable on edge. Attention models may help with longer or denser sequences, yet results indicate that strategy choice, aggregator choice, and stable client training matter more than the encoder.

Anomaly detection runs on the client PC in short five-step windows and is independent of federated rounds. The cadence T or K affects the rate of global improvement but not per window detection latency, which depends on window length and sampling rate. In practice, local training is scheduled at layer or part boundaries or off shift while the machine operates normally; control remains on OEM controllers, and analytics and training run asynchronously at low priority, preserving safety and quality of service while enabling regular model refresh through FL.

For each strategy and aggregator, global and local performance is measured by Accuracy, F1 score, Precision, Recall, and AUC. Convergence is tracked by the L2 norm between successive global weights. For secure and verifiable transfer, RDHE is evaluated separately by PSNR, MSE, and embedding rate. Clients use LSTMs on time-aligned process-specific data. Results assess limits and feasibility for FL in WAAM. Client results show heterogeneity in difficulty and label balance; under this skew, Krum can be conservative when few clients shape update geometry, limiting gains. Trimmed Mean and Multi KRUM stabilize accuracy and recall by reducing aberrant updates. FedProx reduces drift, and FedPer shares a backbone while adapting heads to local data. These trends support robustness under non-IID conditions.

With random uplink loss, the estimator stays unbiased and degrades gracefully as effective participation $\widetilde{K} = K(1 - \rho)$ decreases. Convergence slows with the $1/\sqrt{\widetilde{K}}$, and robust aggregators keep working when the trimming budget is tuned to receive updates. Under sensor perturbations, a five-step window and moment-based vision features reduce variance and average geometry, so decisions flip only near the vision threshold, and time series variance stays low. These safeguards come from the existing pipeline with no protocol changes and guide deployment, while empirical stress tests remain future work. We did not run adversarial studies such as steganalysis of RDHE images or poisoning or inference attacks on the FL loop, and we did not add noise to updates, so a formal ε and δ privacy budget is out of scope. Both are compatible extensions. RDHE tests would target steganalysis and tamper cases under controlled perturbations, and a differential privacy pipeline would use stochastic gradient descent or update perturbation with per-step gradient clipping C, a Gaussian noise multiplier sigma, and composition accounting across local steps and federated rounds. These additions leave the architecture unchanged and can be layered to match a deployment threat model.

5.1. Global Performance Across Strategies and Aggregators

Aggregation choice strongly affects global performance across all configurations, seen in final round metrics and their trajectories in Figures 4–6. Figure 4 summarizes FedAvg across aggregation methods. Accuracy in Figure 4a shows Trimmed Mean and Multi KRUM are higher and more stable than KRUM, which consistently lags. In Figure 4b, KRUM has a highly fluctuating loss, indicating unstable convergence, while Trimmed Mean reduces loss smoothly and quickly. Precision in Figure 4c peaks with Trimmed Mean and Multi KRUM, whereas KRUM remains suppressed. The F1 score in Figure 4d mirrors this, with Trimmed Mean exceeding 0.91 by round 100 . Recall in Figure 4e shows KRUM never exceeds 0.88 , while others stabilize beyond 0.93 . Finally, ROC AUC in Figure 4f confirms the early underperformance of KRUM and the consistent superiority of Multi KRUM and Trimmed Mean through the rounds.

Figure 5 shows FedPer learning fastest and most robust across metrics. In Figure 5a, accuracy quickly exceeds 0.95 for all aggregators except KRUM. Figure 5b shows the loss under the Trimmed Mean and Multi KRUM dropping sharply and staying under 0.2 , while KRUM remains high. Precision in Figure 5c is similar for Vanilla, Multi KRUM, and Trimmed Mean, with KRUM weakest. Figure 5d confirms F1 score dominance as Trimmed Mean passes 0.91 early and holds it. Recall in Figure 5e starts strong for all, then KRUM degrades. In Figure 5f, ROC AUC hits 0.94 by round 30 with Trimmed Mean, indicating strong early-stage generalization and robustness.

Figure 6 focuses on the FedProx strategy, which achieves stable but slower improvements across all metrics compared to FedAvg and FedPer. Figure 6a shows moderate accuracy growth with Trimmed Mean and Multi Krum outperforming KRUM, although the convergence is slower than FedPer. In Figure 6b, the global loss steadily declines under all methods, but remains above 0.25 for KRUM, indicating ineffective learning. Precision (Figure 6c) remains balanced and less volatile under Trimmed Mean and Vanilla, while KRUM again underperforms. The F1-score in Figure 6d demonstrates smoother progression, particularly for Trimmed Mean, reaching nearly 0.91 by round 100. Recall (Figure 6e) improves gradually across configurations, but the KRUM curve stagnates around 0.86. In Figure 6f, the ROC AUC values confirm that Trimmed Mean ensures robust and steady growth, while KRUM fails to generalize effectively.

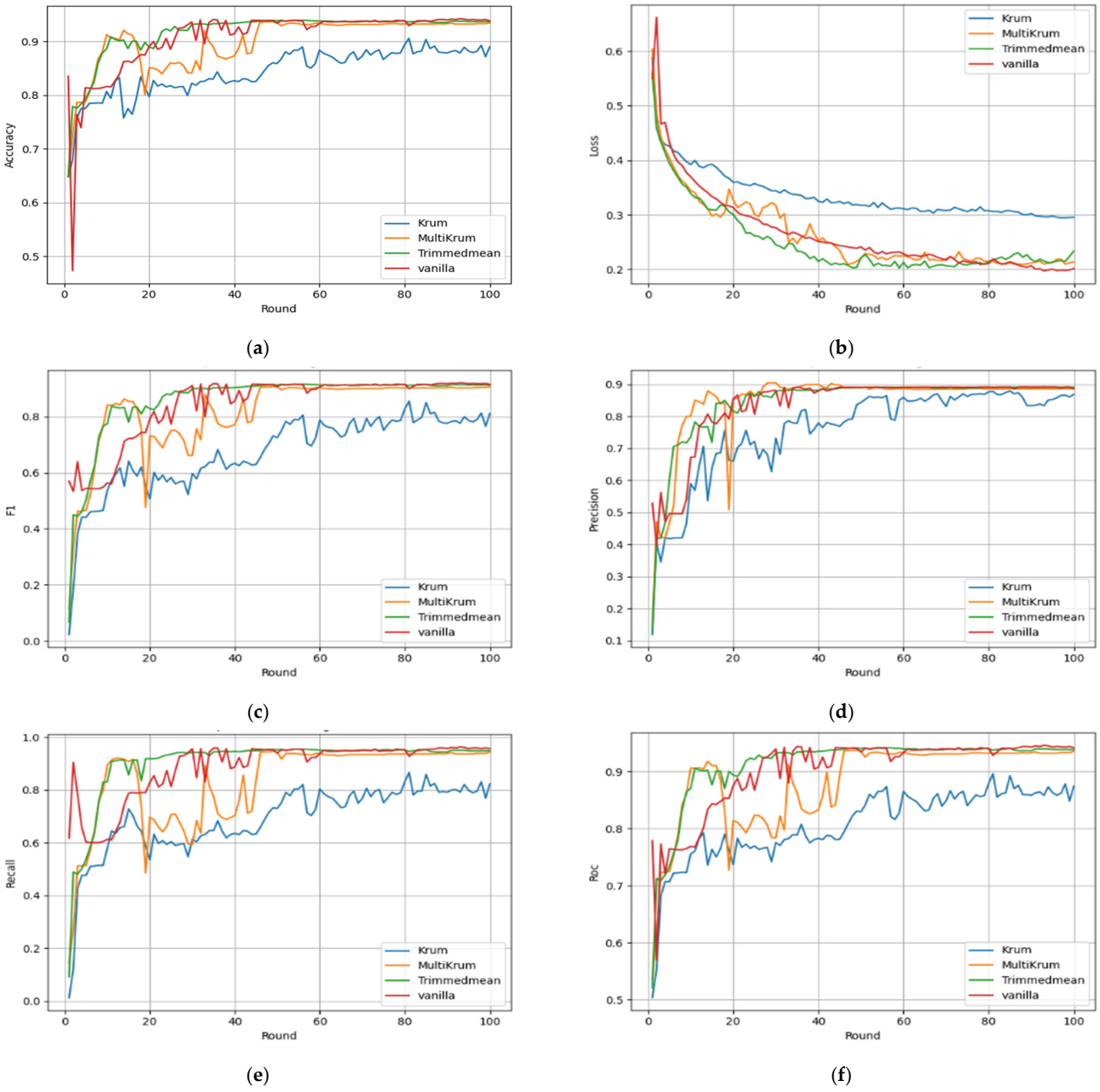


Figure 4. Final metrics vs. rounds for FedAvg: (**a**) accuracy, (**b**) loss, (**c**) precision, (**d**) F1-score, (**e**) recall, and (**f**) area under the curve.

FedPer leads across metrics, especially with Trimmed Mean or Multi KRUM. Personalization adapts to heterogeneity, giving faster convergence and higher accuracy. FedAvg works with robust aggregators but degrades with KRUM under noise and imbalance. FedProx is slower yet converges stably and resists client drift, suiting consistency priority. Overall, FedPer with Trimmed Mean is the most effective and balanced for secure, generalizable WAAM anomaly detection.

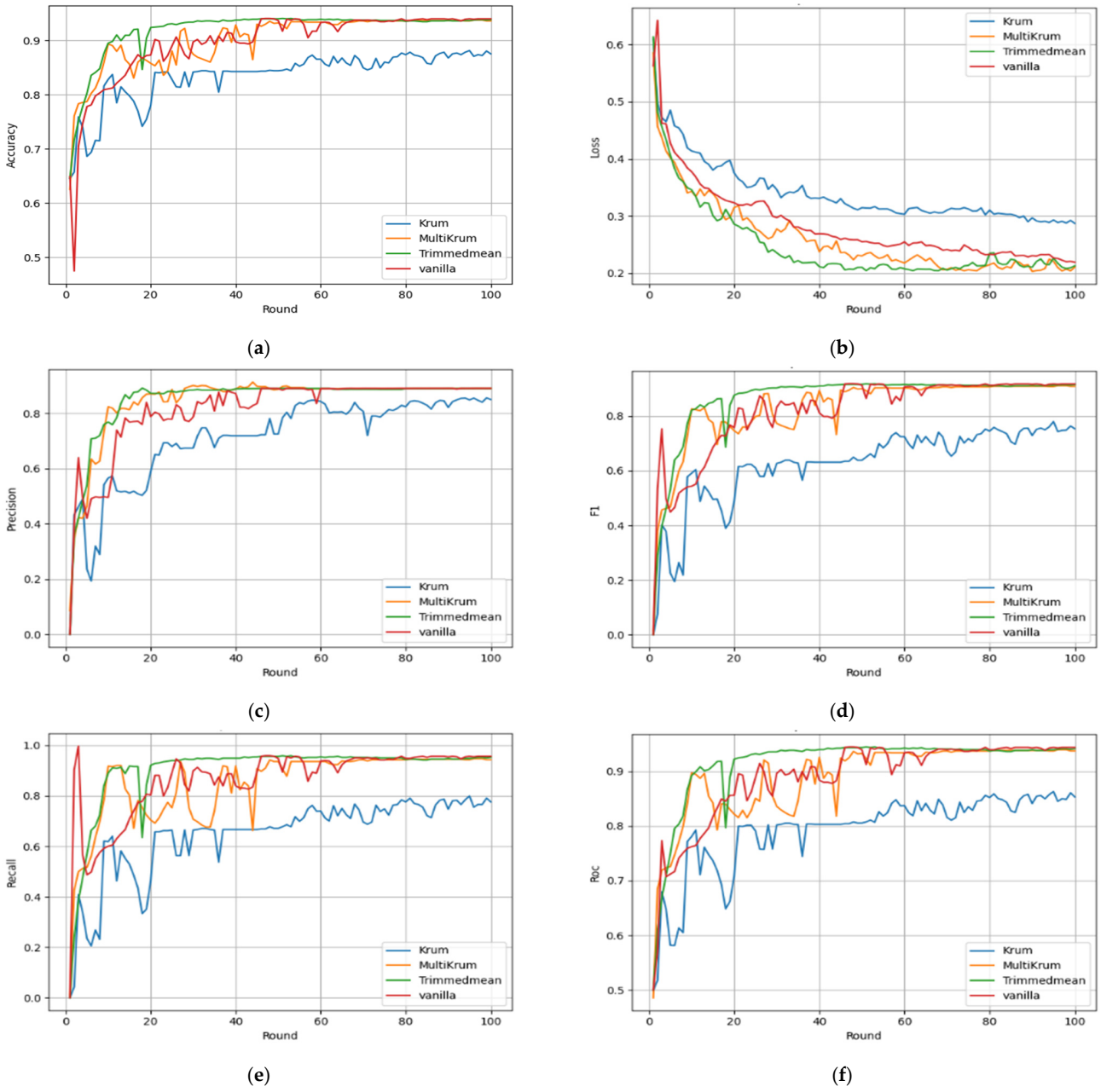


Figure 5. Final metrics vs. rounds for FedPer: (**a**) accuracy, (**b**) loss, (**c**) precision, (**d**) F1-score, (**e**) recall, and (**f**) area under the curve.

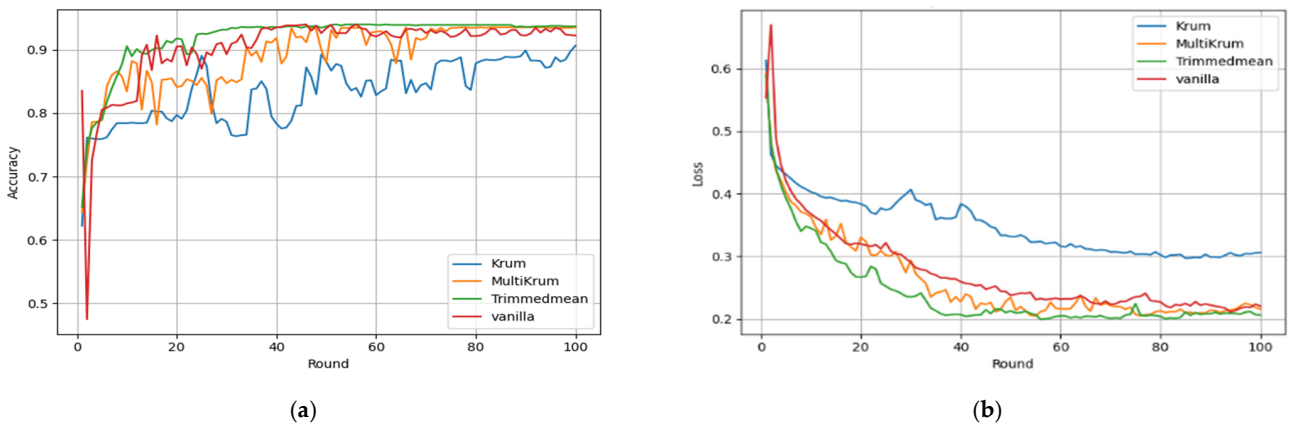


Figure 6. *Cont.*

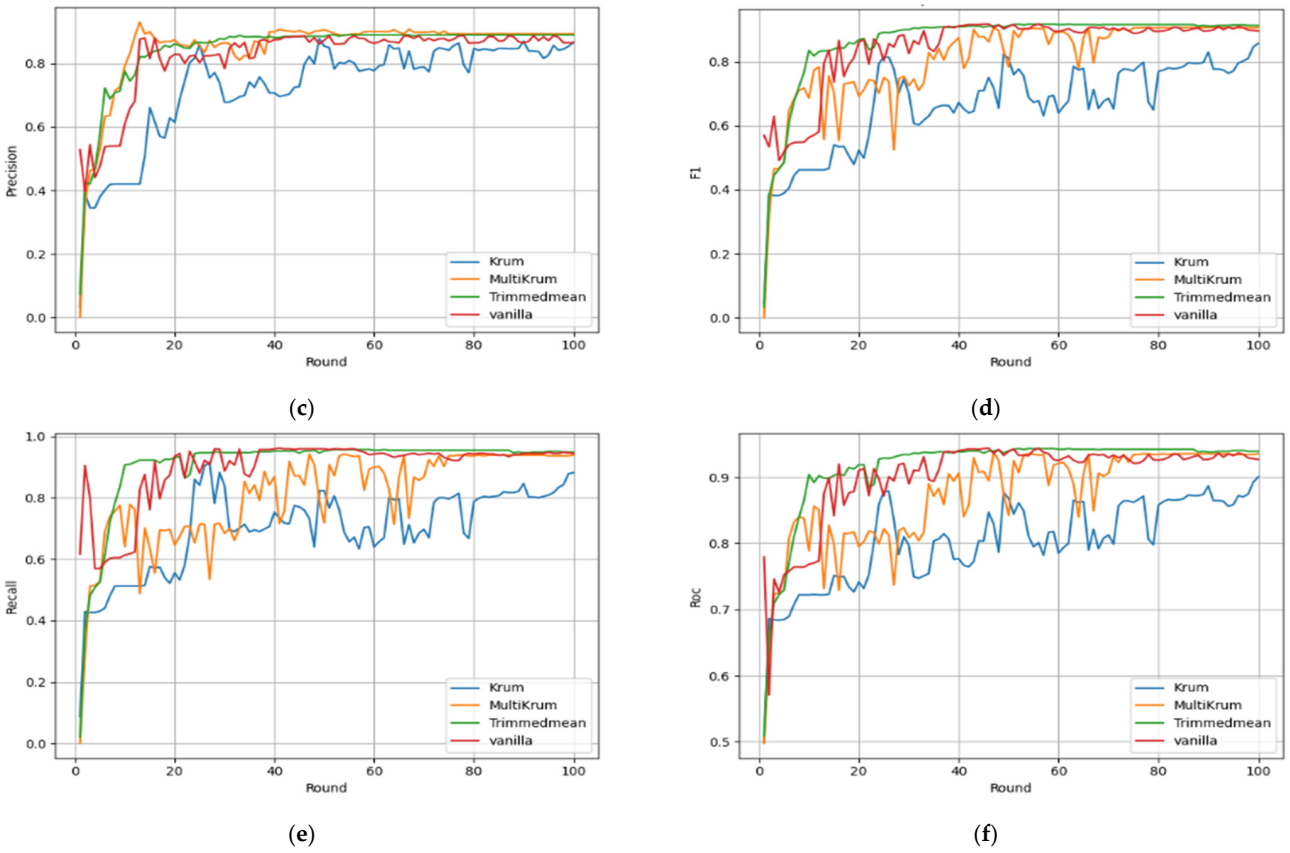


Figure 6. Final metrics vs. rounds for FedProx: (**a**) accuracy, (**b**) loss, (**c**) precision, (**d**) F1-score, (**e**) recall, and (**f**) area under the curve.

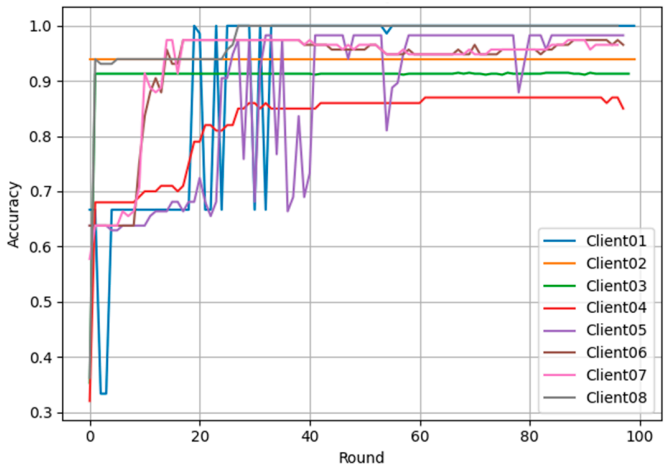
5.2. Client-Level Performance Disaggregation

The client-level performance, Figure 7, shows large client-level differences under non-identical data. Architecture, modality, and process complexity create distinct convergence patterns. Under FedAvg, Figure 7a–d, disparities are sharp. Client 3 and Client 8 stay above 90 percent, while Client 4 stalls near 80 percent with higher loss. KRUM worsens gaps, causing erratic behavior for Clients 1 and 5 and delayed gains for Client 2. Multi KRUM and Trimmed Mean give better consistency and accuracy. With FedPer, Figure 7e–h, trajectories stabilize; most clients exceed 90 percent, especially with Trimmed Mean and Vanilla. Client 2 and Client 6 rise early and converge robustly. Personalization supports diverse inputs and reduces outlier impact. FedProx, Figure 7i–l, is intermediate; slower than FedPer yet helps Client 5 recover from early instability. KRUM again induces oscillations and weak learning. Trimmed Mean remains most balanced, giving smooth fair gains while preserving diversity with stability.

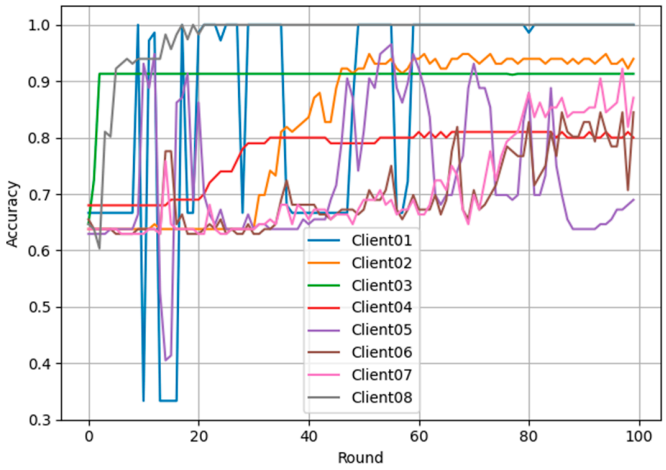
Complementing these accuracy results, Figure 8 shows client-wise loss and how aggregation affects stability. Figure 8a–d shows the losses mirror accuracy. KRUM causes volatility and plateaus for Clients 1 and 5. Multi KRUM and Trimmed Mean reduce loss, while Vanilla shows early variability. Figure 8e–h shows lower losses with FedPer, and Trimmed Mean is the smoothest and consistent. Clients 4 and 7 with geometric or time-sensitive inputs show minor instability, yet losses stay under 0.25. Figure 8i–l shows decay. Trimmed Mean stays consistent, KRUM hinders Clients 6 and 3, and Vanilla suits stable Clients 3 and 8, but can spike under FedProx.

Overall, FedPer with Trimmed Mean or Multi KRUM leads, handles heterogeneity, and converges quickly. FedAvg works with robust aggregators but degrades under KRUM due

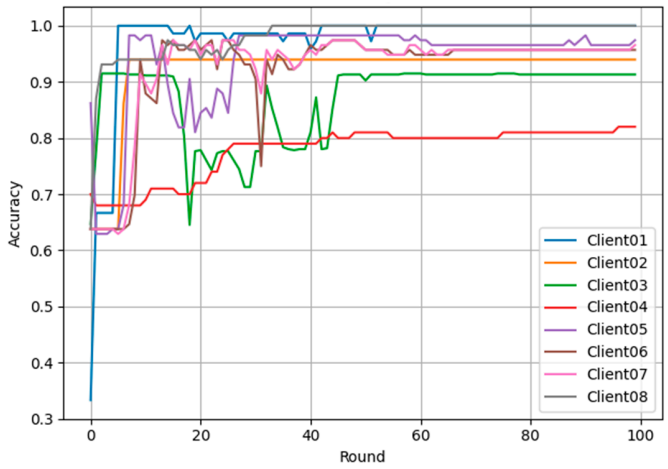
to noise and imbalance. FedProx is slower, stable, and resists drift. FedPer with Trimmed Mean is most effective for secure, generalizable anomaly detection in decentralized WAAM.



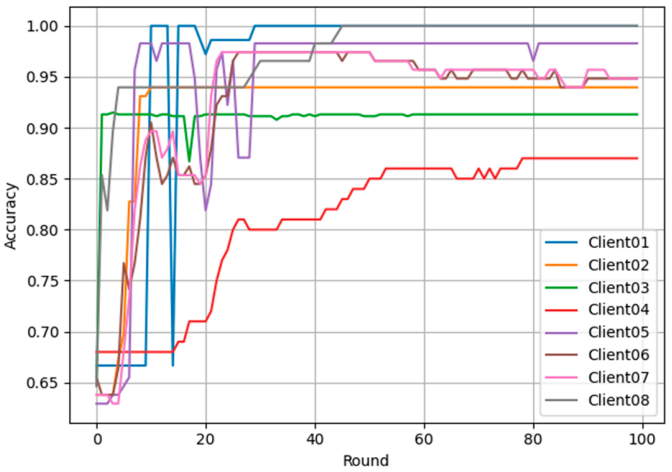
(**a**) Vanilla



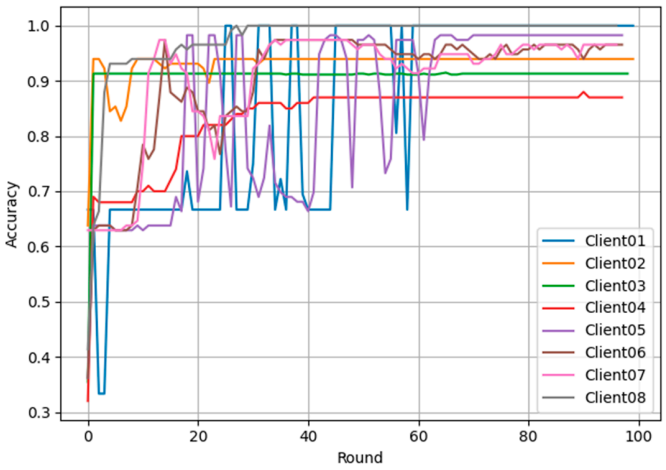
(**b**) KRUM



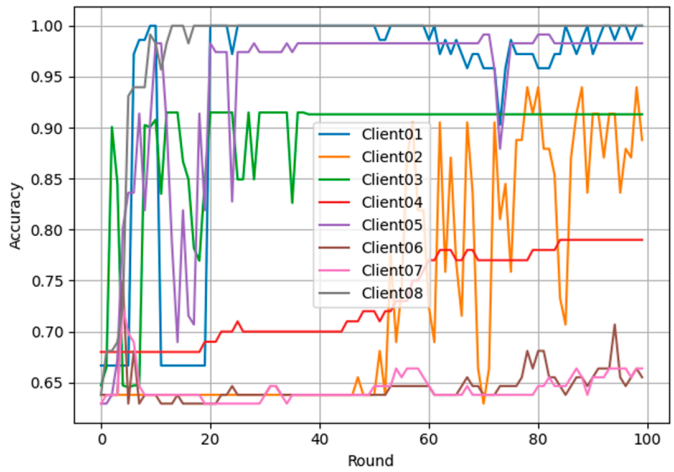
(**c**) Multi KRUM



(**d**) Trimmed Mean

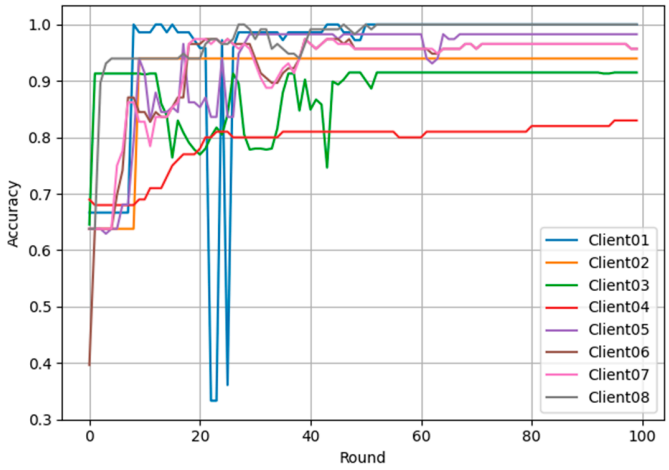


(**e**) Vanilla

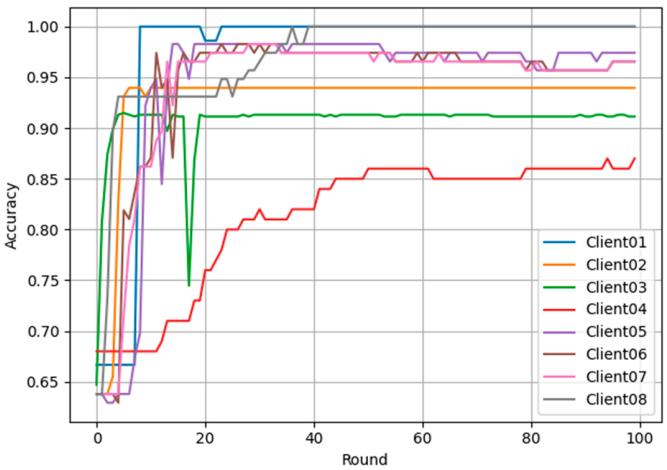


(**f**) KRUM

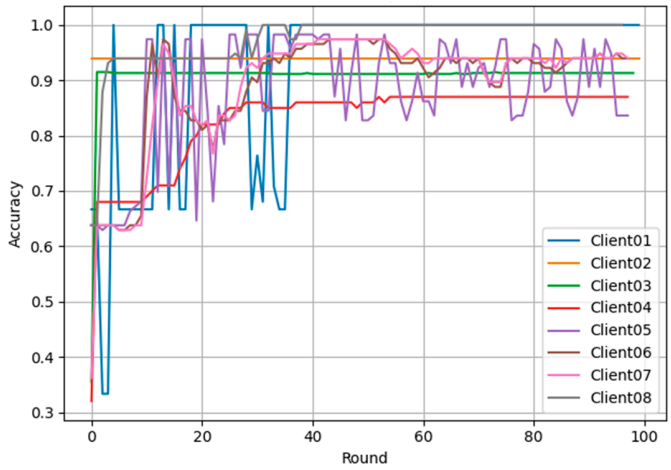
Figure 7. *Cont.*



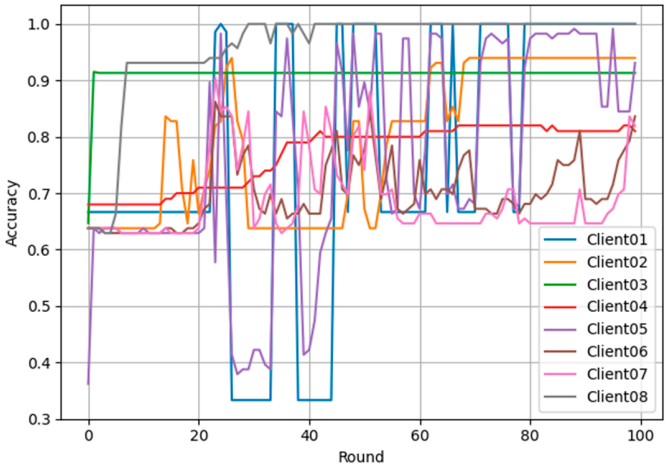
(**g**) Multi KRUM



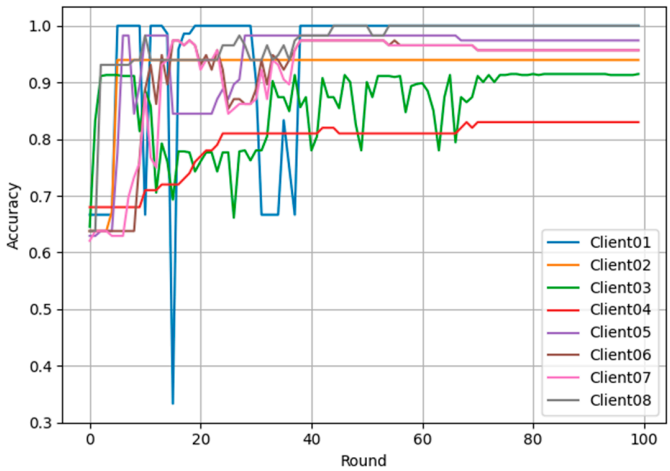
(**h**) Trimmed Mean



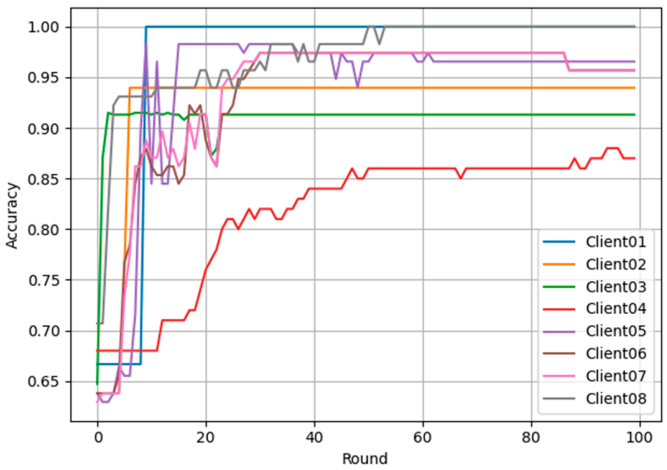
(**i**) Vanilla



(**j**) KRUM

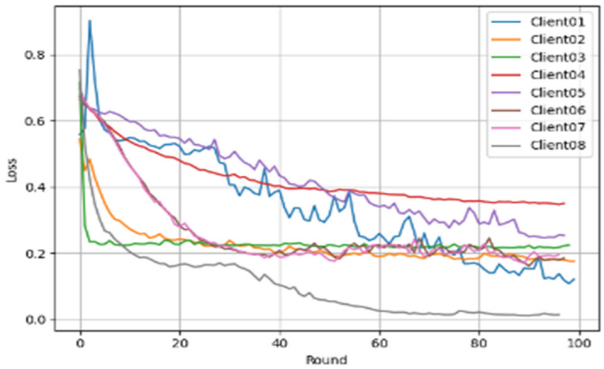


(**k**) Multi KRUM

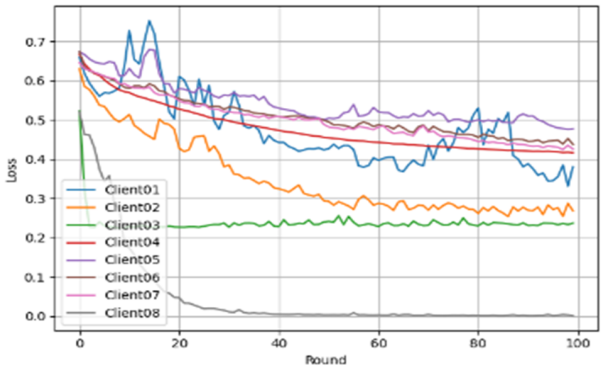


(**l**) Trimmed Mean

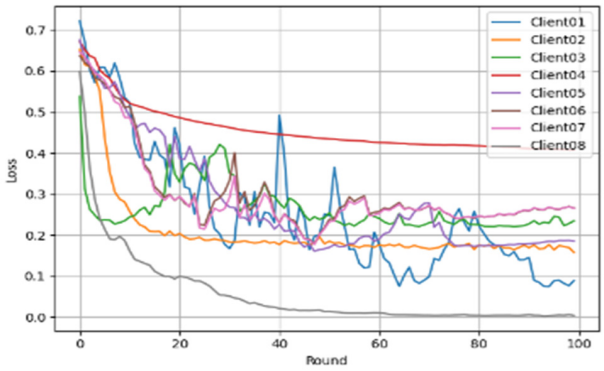
Figure 7. Accuracy vs. rounds for different strategies (Vanilla, KRUM, Multi KRUM, and Trimmed Mean) for (**a**–**d**) FedAvg, (**e**–**h**) FedPer, and (**i**–**l**) FexProx.



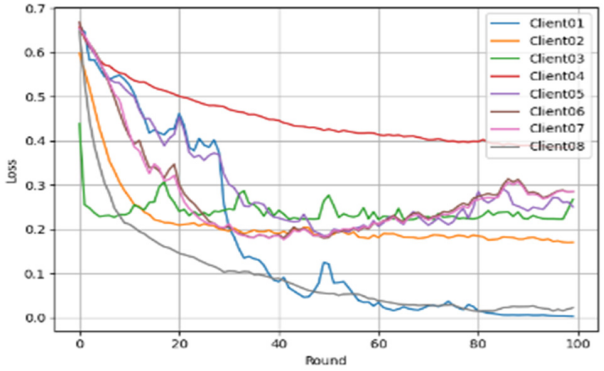
(**a**) Vanilla



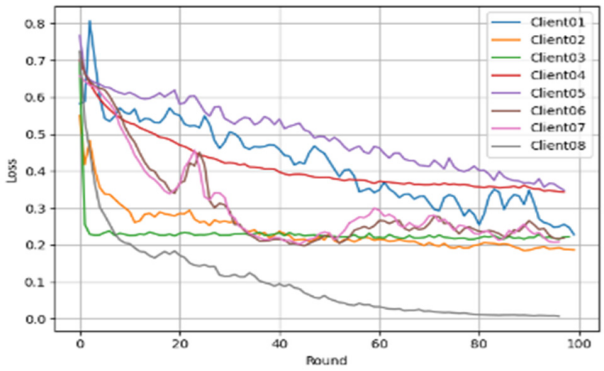
(**b**) KRUM



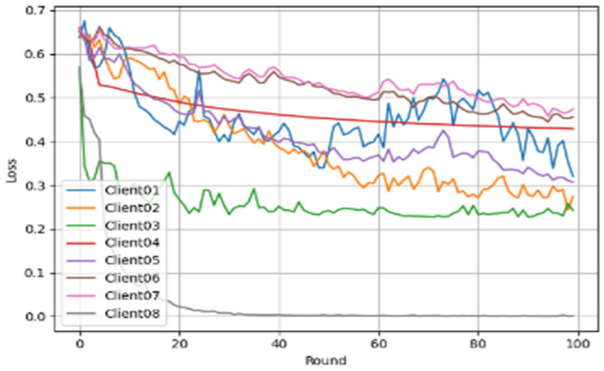
(**c**) Multi KRUM



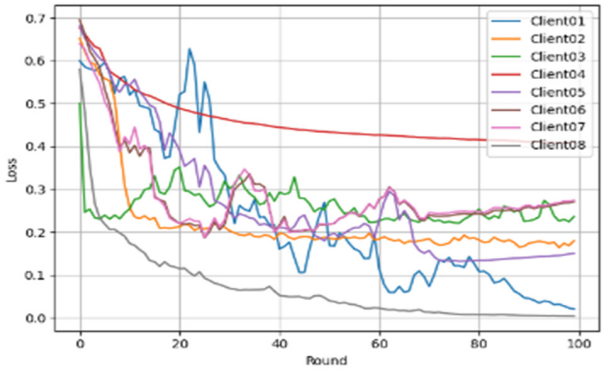
(**d**) Trimmed Mean



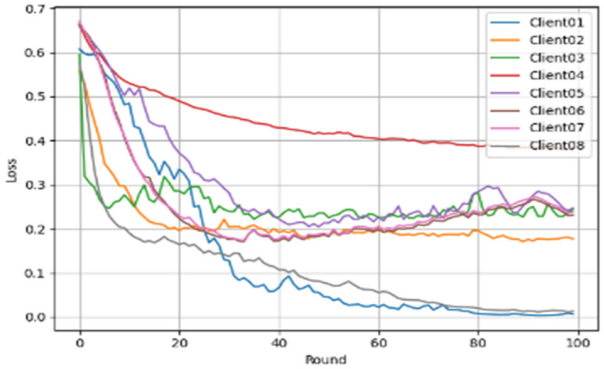
(**e**) Vanilla



(**f**) KRUM

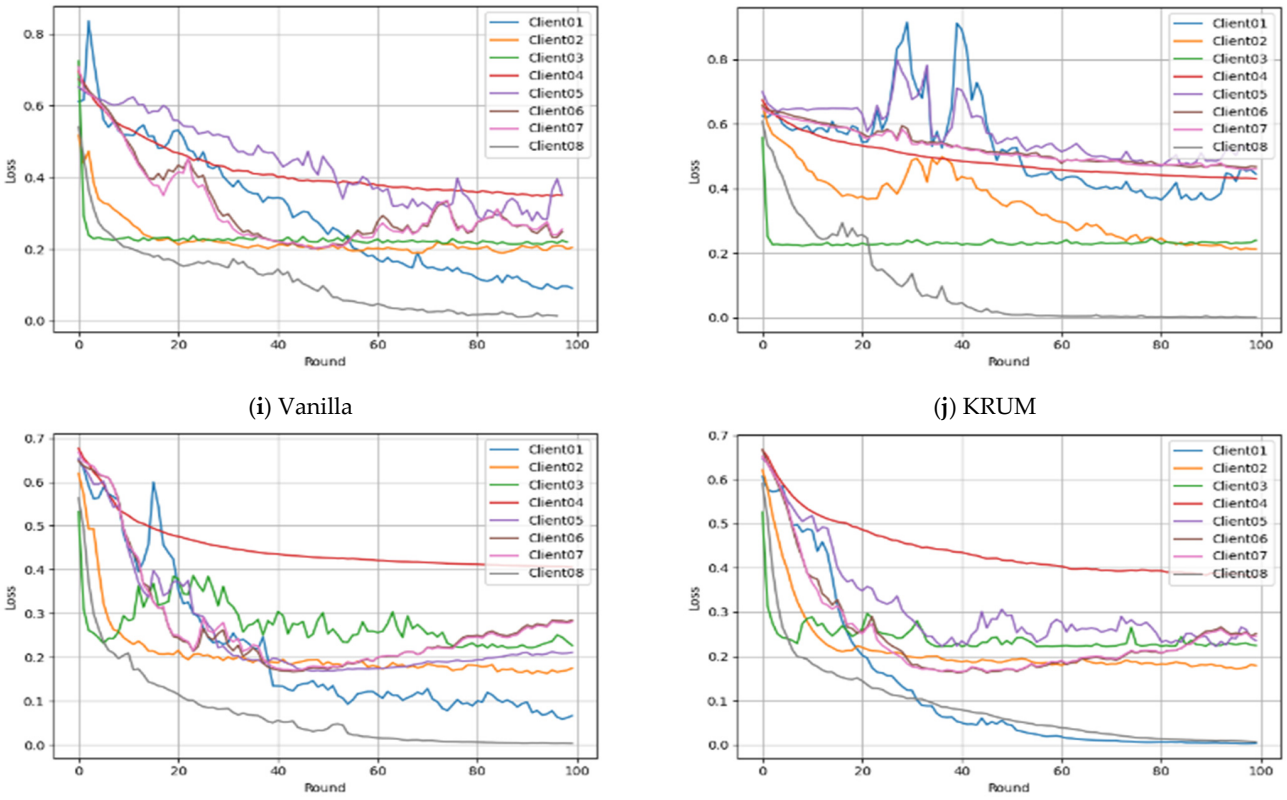


(**g**) Multi KRUM



(**h**) Trimmed Mean

Figure 8. *Cont*.



(**i**) Vanilla (**j**) KRUM

(**k**) Multi KRUM (**l**) Trimmed Mean

Figure 8. Loss vs. rounds for different strategies (Vanilla, KRUM, Multi KRUM, and Trimmed Mean) for (**a**–**d**) FedAvg, (**e**–**h**) FedPer, and (**i**–**l**) FexProx.

5.3. Client-Wise Confusion Matrices Analysis

Client-wise confusion matrices in Tables S1–S3 show that heterogeneity drives outcomes beyond global metrics. Clients sensed different signals, current and voltage profiles, bead geometry, travel speed, and part parameters, creating non-uniform distributions and greater task complexity. Under FedAvg, Multi KRUM, and Trimmed Mean delivered consistent performance, accuracy above 0.90 for all except Client 5, which rose from 0.69 with KRUM to 0.98 with robust aggregation. Client 4, using arc length and feed angle, had high false positives with KRUM FP = 17, Accuracy = 0.80, indicating sensitivity to parameter variability and that conservative aggregation can suppress local discriminability. FedPer was stronger for Clients 5 to 7, where Trimmed Mean and Multi KRUM pushed accuracy above 0.96 and reduced false negatives. Client 6 improved from 0.66 with KRUM to 0.97 with robust methods, showing that personalization reduces misclassification of current and voltage fluctuations from stochastic electrical noise. FedProx stabilized training, especially Clients 4 and 5; Trimmed Mean raised all clients above 0.94, offsetting KRUM underperformance: Client 5 Accuracy = 0.93. Clients 1 and 8 kept low false positives and false negatives across settings, implying linearly separable features like centroid shifts and deposition symmetry.

A closer look at client trade-offs shows that strategy and aggregation effects vary by the nature and entropy of captured signals. Client 1 modeling current voltage time achieved perfect classification across all strategies and aggregators (Accuracy = 1.00), implying separable anomalies and little ambiguity. Client 2, using high dynamic range vision, was aggregation sensitive; under FedPer, KRUM accuracy fell to 0.89 with false positives 6, while FedAvg and FedProx stayed near 0.94 with Trimmed Mean and Multi KRUM, so robust averaging is needed for vision features where illumination and geometry can mimic

anomalies. Client 3, with speed and position, was invariant near 0.91 across strategies, indicating LSTM captures temporal patterns and aggregation matters little when evolution is strongly time dependent.

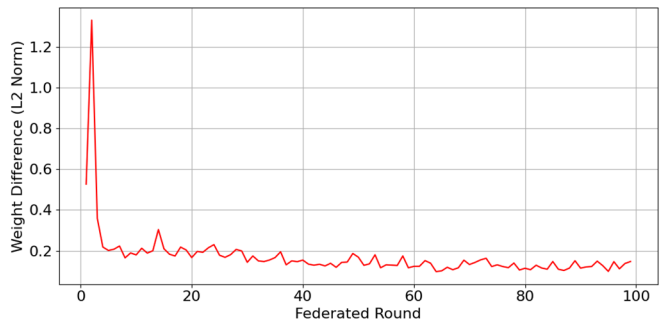
Client 4, with arc length and feed angle, was the hardest to stabilize. KRUM produced high false positives, 17 to 19, across strategies due to irregular morphology and erratic gradients; FedPer and FedProx nudged accuracy to 0.87, and Multi KRUM was steadier than Vanilla. Client 5, combining current speed and voltage, showed large gaps, 0.69 in FedAvg KRUM to 0.98 in FedPer Trimmed Mean, proving complex couplings benefit from personalization and robust outlier suppression. Clients 6 and 7 with overlapping domains, such as voltage speed and voltage time reached Accuracy greater than 0.96 only under Multi KRUM and Trimmed Mean, underscoring variance-aware aggregation in non-IID edge settings. Client 8 tracking centroid shifts held 100 percent accuracy in all configurations, serving as a stability baseline. These behaviors support strategy adaptive allocation in WAAM, assigning aggregation by sensing modality to improve resilience and fault localization. From Section 3.3, the runtime fits a CPU-only industrial PC x86 64 with four to eight cores and eight to sixteen GB RAM, and with an uplink of nearly 1.1 MB per round. Section 4.4, plus CPU-friendly preprocessing, Section 4.1, and RDHE, Section 3.2, the end-to-end loop is practical near the machine.

5.4. Comparative Convergence Trends

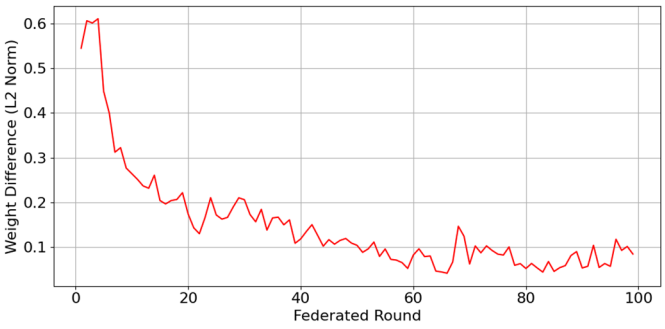
FedAvg convergence in Figure 9 shows clear differences across aggregators using L2 norm trends. FedAvg Vanilla drops steeply, then oscillates before stabilizing near round 60, Figure 9a, due to aggressive averaging of heterogeneous updates without correction or outlier rejection, which can cause overfitting or oscillatory behavior. FedAvg KRUM Figure 9b and FedAvg Multi KRUM Figure 9c are steadier. Multi KRUM declines smoothly by excluding extreme updates and is robust to Byzantine and stochastic noise. KRUM fluctuates early, but volatility dampens after round 30. FedAvg Trimmed Mean Figure 9d is the slowest, plateauing near round 50 from underfitting when non-IID data leads to the removal of useful signals. Thus, FedAvg is fast initially, but generalization and consistency hinge on the aggregator under heterogeneity.

FedPer behaves differently because personalization decouples local and global parts. FedPer Vanilla Figure 9e shows a smooth L2 norm decline with minor oscillations as shared layers synchronize while local heads diverge. FedPer KRUM Figure 9f has sharp early volatility until about round 30 since selective filtering conflicts with already personalized updates. FedPer Multi KRUM Figure 9g better suppresses noise with a mostly monotonic decrease and rare spikes. FedPer Trimmed Mean Figure 9h slows after round 40 and underfits as boundary updates are dropped with reduced shared capacity. Personalization aids stability, while the aggregator sets the balance between adaptability and synchronization.

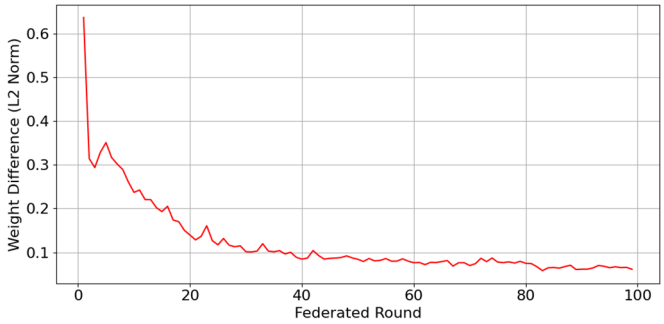
FedProx adds proximal regularization that controls gradient drift. FedProx Vanilla Figure 9i shows a steady L2 norm decrease with small oscillations as the proximal term pulls local updates toward the global direction, more stable than early FedAvg. FedProx KRUM Figure 9j converges later and fluctuates early since selection favors already proximal updates and adds little when variance is suppressed. FedProx Multi KRUM Figure 9k keeps a stable, steep slope, combining outlier filtering with proximity anchoring. FedProx Trimmed Mean Figure 9l is slower and plateaus early because excessive exclusion compounds conservative local training. Overall, FedProx with moderate aggregators gives fast, stable convergence, but aggressive selection is suboptimal, especially with structurally aligned clients.



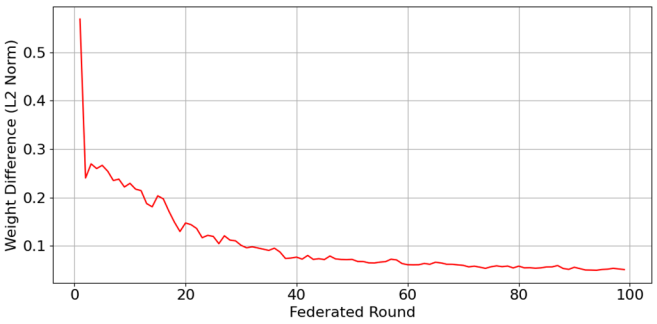
(**a**) Vanilla



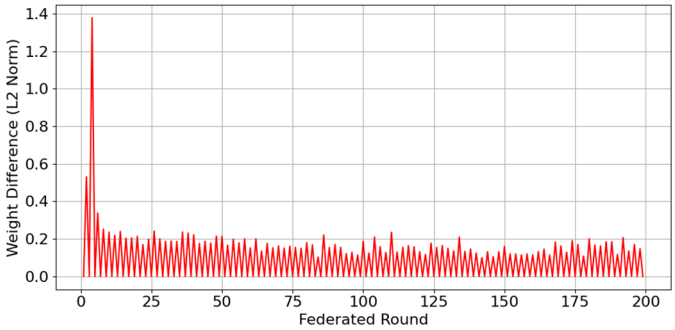
(**b**) KRUM



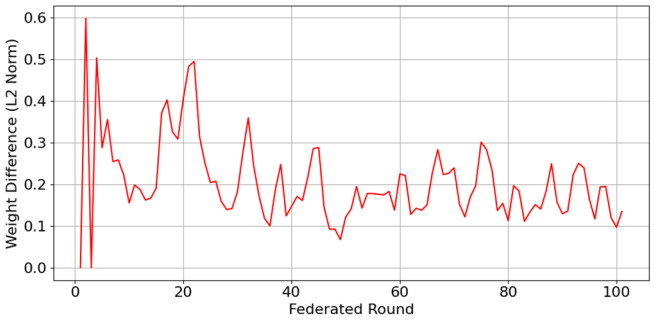
(**c**) Multi KRUM



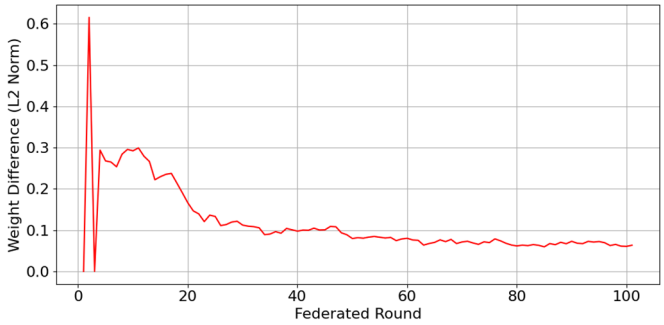
(**d**) Trimmed Mean



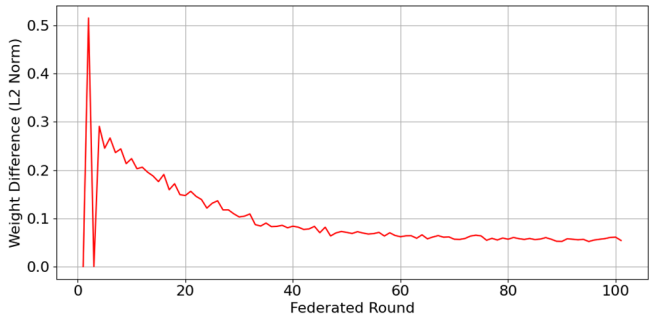
(**e**) Vanilla



(**f**) KRUM

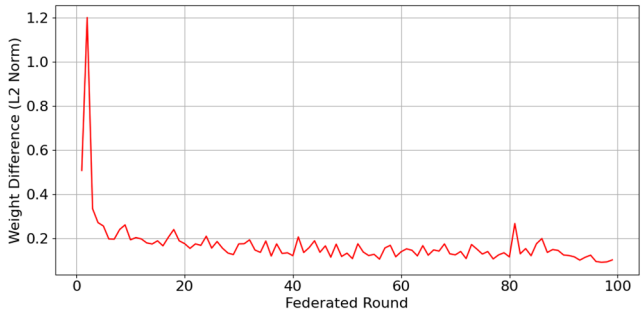


(**g**) Multi KRUM

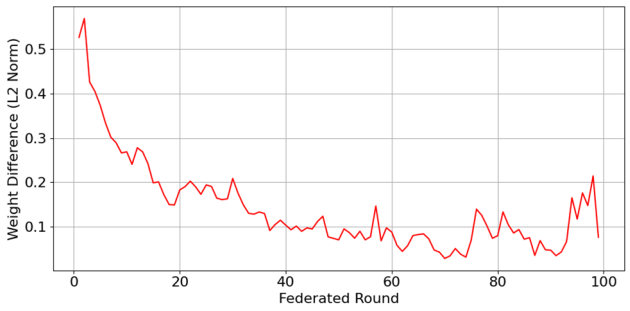


(**h**) Trimmed Mean

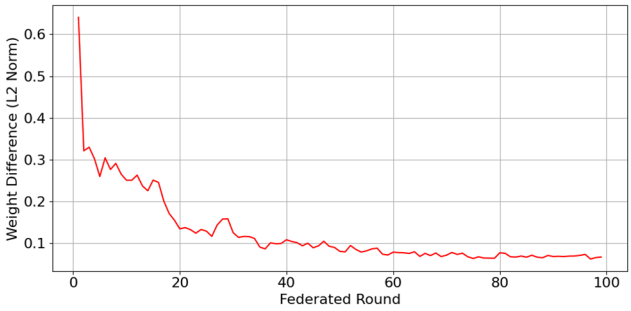
Figure 9. *Cont*.



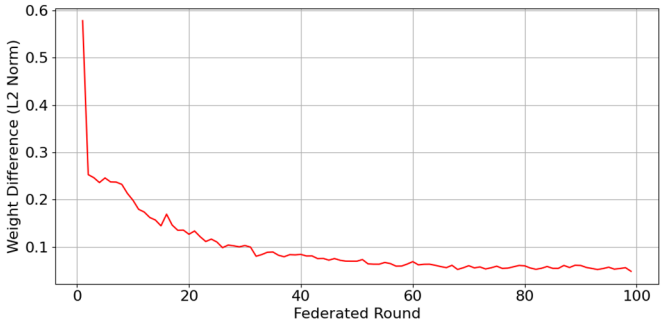
(**i**) Vanilla



(**j**) KRUM



(**k**) Multi KRUM



(**l**) Trimmed Mean

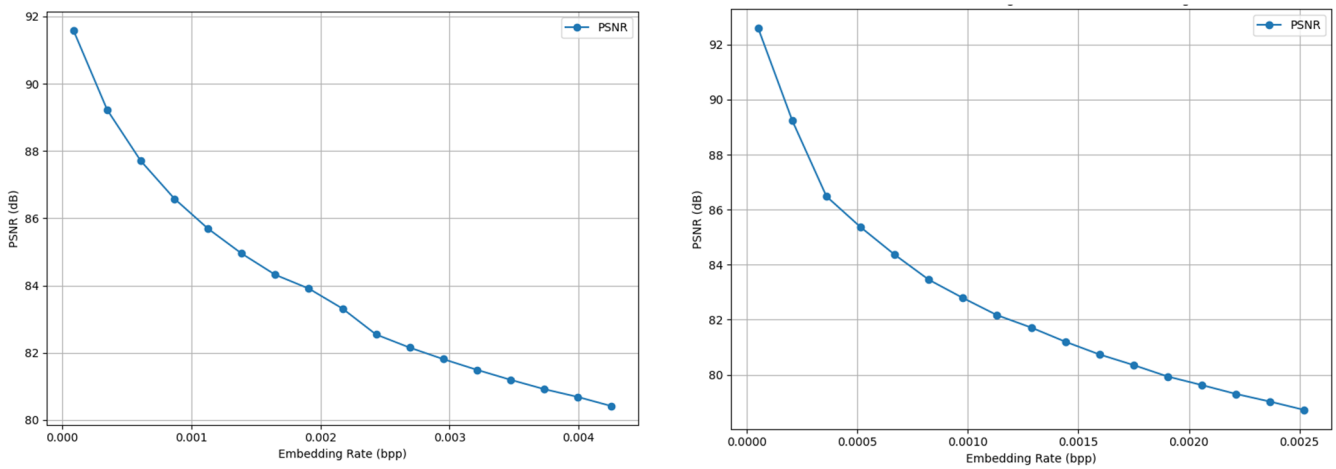
Figure 9. Global model convergence vs. rounds for different strategies (Vanilla, KRUM, Multi KRUM, and Trimmed Mean global) for (**a**–**d**) FedAvg, (**e**–**h**) FedPer, and (**i**–**l**) FedProx.

Across strategies and aggregators, convergence reflects the interplay among robustness, update regularization, and client heterogeneity. FedAvg converges fast yet remains volatile with non-robust aggregators such as Vanilla and KRUM, with L2 norm swings past round 50 that suggest overfitting from directly averaging divergent updates from heterogeneous WAAM clients. FedPer reduces oscillations through personalized heads that decouple client-specific representations, letting the global backbone settle while locals diverge; FedPer Multi KRUM and FedPer Trimmed Mean flatten after about round 30, indicating personalization preserves diversity while robust filtering limits outliers. FedProx uses proximal anchoring, yielding tighter L2 norm profiles, especially with Multi KRUM, which suppresses gradient drift for uniform descent, whereas Trimmed Mean often flattens early and underfits by removing too much gradient diversity. Thus, optimal convergence in federated WAAM aligns each strategy with an aggregation rule that balances noise suppression and representational diversity.

5.5. Reversible Data Hiding Evaluation

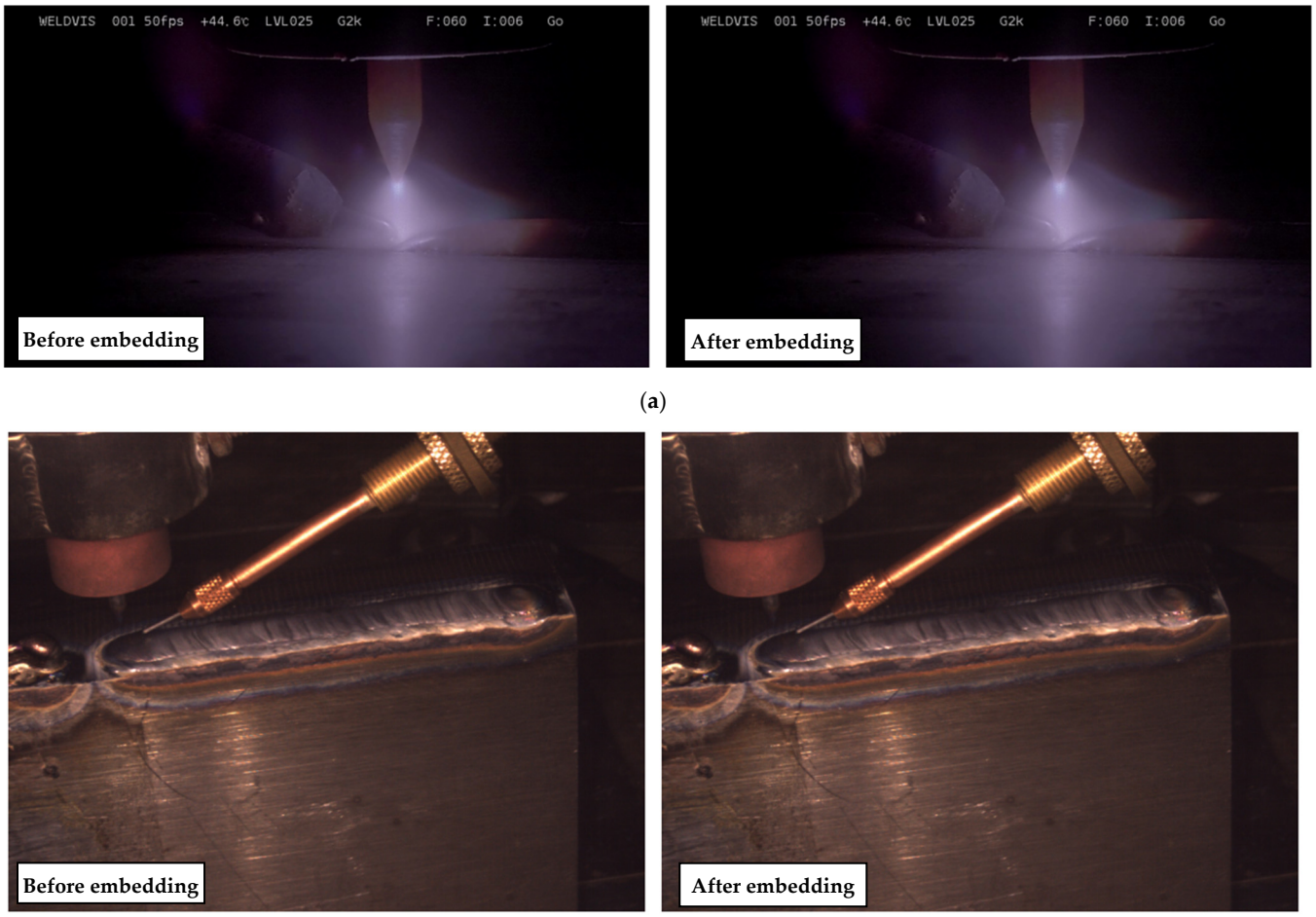
To assess RDHE in the encrypted domain on WAAM image streams, we combined quantitative distortion analysis with visual checks across embedding rates. PSNR stayed above 90 dB at the very low rate 0.0001 bpp. As payload grew, PSNR declined smoothly to 79.1 dB at 0.0043 bpp for Image 1 and 78.9 dB at 0.0025 bpp for Image 2, showing controlled distortion as in Figure 10. Side-by-side comparisons of original and encoded frames in pre-deposition and post-deposition scenes in Figure 11 showed no visible structural loss. Channel-wise pixel index plots for red, green, and blue across 10,000 sampled pixels were nearly identical, confirming below-perceptual embedding (Figure 12; Image 1) during deposition at 0.0007 bpp, yielded PSNR 91.6 dB and MSE 0.35, while Image 2, after deposition with richer chromatic content, reached 92.5 dB and MSE 0.31 at 0.0004 bpp. Despite spectral

and context differences, both cases show RDHE preserves pixel-level integrity needed for later DNN-based segmentation or anomaly localization.



(**a**) (**b**)

Figure 10. PSNR vs. embedding rate for (**a**) test image 1, before embedding, and (**b**) test image 2, after embedding.



(**a**)

(**b**)

Figure 11. Condition of test images before and after embedding for (**a**) test image 1, and (**b**) test image 2.

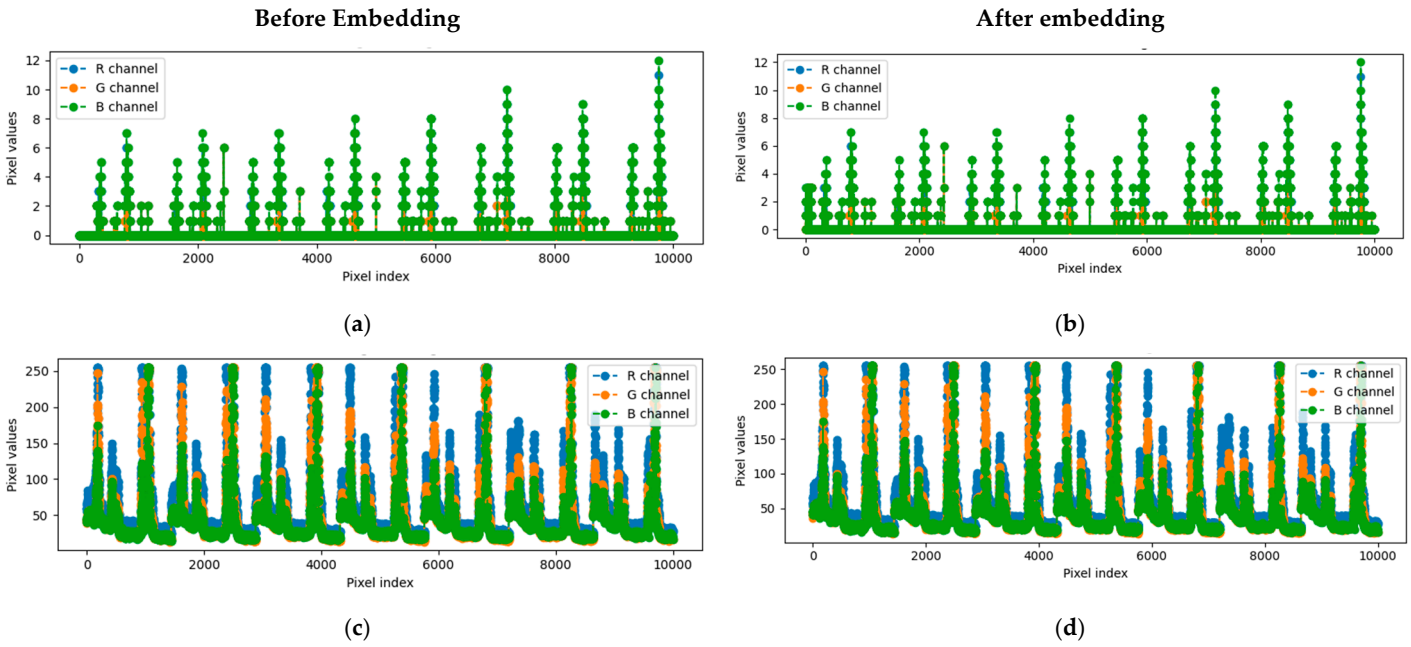


Figure 12. Pixel distribution of test image 1 (**a**-*before embedding*, **b**-*after embedding*) and test image 2 (**c**-*before embedding*, **d**-*after embedding*).

Privacy-preserving advantages for federated WAAM. It helps when raw signals, such as arc current, voltage, or bead geometry, cannot be sent due to bandwidth or regulation. Embedding encrypted payloads in weld images enables out-of-band exchange for cross-client calibration, asynchronous aggregation, and offline defect labeling while avoiding plaintext exposure. Pixel domain invariance shows that convolutional features used by extraction pipelines remain unchanged after embedding, preserving model compatibility. Image 2 carried 4556.25 kbits at 0.0004 bpp, and Image 1 carried 2700 kbits at 0.0007 bpp, enough for multichannel sensor streams with timestamps. RDHE is completely reversible, decoded images match originals bit for bit with consistent PSNR and MSE parity. Although FL reduces centralized risk, RDHE further secures image-based transfer during client dropout or hybrid protocols, adding a confidentiality layer with minimal compute cost and no impact on downstream analytics.

5.6. Optimal Configuration Synthesis

Across metrics, client consistency, convergence, and confusion matrices, the best WAAM federation is FedPer with Trimmed Mean. In Section 5.1, it achieved the top global F1 score and ROC AUC for both classes. Section 5.2 shows about 95 percent accuracy and smooth loss decay for almost all clients, including noisy geometric and temporal cases such as Clients 4 and 7. Section 5.3 confirms minimal false positives and false negatives and perfect classification for structured Clients 1 and 8. Section 5.4 reports stable convergence that balances early learning speed with long-term stability. FedPer provides client-specific adaptation, while Trimmed Mean suppresses gradient anomalies without discarding useful updates, enabling generalization across non-IID clients and making it the most suitable decentralized WAAM choice.

Context matters. FedProx with Multi KRUM gives stable convergence and variance control for temporally regular clients, as in Section 5.4, but weakens on geometrically noisy or irregular data, such as Client 4, with lower accuracy and slower loss convergence than FedPer. FedAvg with KRUM, though theoretically robust to adversaries, showed unstable learning and poor recall, especially for Clients 5 and 7, due to over-pruning valid noisy updates. The results expose trade-offs between personalization and generalization and

between robustness and inclusivity. FedPer keeps client specificity essential in sensor diverse WAAM, and Trimmed Mean aggregates without sidelining atypical distributions. Given asynchronous sampling, cross-sensor drift, and environmental variation in practice, we recommend FedPer with Trimmed Mean as the deployment baseline.

6. Conclusions

The framework delivers an FL-based anomaly detection system for WAAM by combining privacy-preserving transmission with multi-sensor feature fusion. Reversible data hiding embeds encrypted variables such as voltage, current, arc geometry, and travel speed in high-resolution weld images, so raw logs never leave the cell. Clients decode these images and train temporal and spatial models on fused features from vision, signal streams, and geometric estimates. Although clients use different input dimensions, the global model updated with FedAvg, FedProx, or FedPer generalizes under strong non-IID fragmentation. Across 100 rounds, the system reaches about 95 percent accuracy with gains in F1 score, convergence stability, and ROC AUC over the baseline FedPer with Trimmed Mean. These results show the framework preserves privacy while enabling accurate anomaly detection in WAAM-based smart manufacturing.

Future work will increase adaptability and semantic depth. Domain generalized federated transfer learning will align client domains with a shared space using personalized heads or latent mappings, improving detection for new tool wear or substrate conditions. Transformer encoders and modality-specific attention can fuse image descriptors, geometric estimates, and dynamic signals to learn cross-stream dependencies with limited data per client. For secure collaboration, blockchain-backed aggregation or trusted execution environments can resist adversarial updates and provide audit trails. Adaptive client selection using contribution entropy or task-specific gradients can reduce communication and speed convergence in diverse WAAM networks. The framework already handles non-IID variation; full out-of-distribution and fast drift tests, and formal privacy guarantees are the next steps. The pipeline remains compatible with Transformer and TCN backbones for longer horizons and cross-sensor attention, independent of FL security and aggregation. Together, these steps guide a secure and generalizable federation toward autonomous anomaly-aware cyber-physical manufacturing.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/machines13111063/s1, Algorithm S1: Centroid and Speed Extraction from Video. Algorithm S2: Temporal alignment of multi-sensor data to a common timeline. Algorithm S3: Arc Length and Feed Angle Extraction. Algorithm S4: Binary Classification Using LSTM Sequence Encoding. Algorithm S5: Non-IID client partitioning with explicit rationale metadata. Algorithm S6: Federated Client-Side Training Procedure. Algorithm S7: Client-Side Evaluation Procedure. Algorithm S8: FedAvg Strategy. Algorithm S9: FedProx Strategy. Algorithm S10: FedPer Strategy. Algorithm S11: KRUM Aggregation. Algorithm S12: MULTIKRUM Aggregation. Algorithm S13: TRIMMED MEAN Aggregation. Algorithm S14: Strategy Selection and Server Launch. Algorithm S15: Round-wise Robust Aggregation and Convergence. Algorithm S16: Evaluation Metric Aggregation. Table S1. Client-wise confusion matrices after 100 communication rounds for the FedAvg strategy. Table S2. Client-wise confusion matrices after 100 communication rounds for the FedPer strategy. Table S3. Client-wise confusion matrices after 100 communication rounds for the FedProx strategy.

Author Contributions: Conceptualization, M.A.G.R., K.-C.L. and D.B.K.; Methodology, M.M.M. and D.B.K.; Software, M.M.M.; Validation, M.M.M.; Formal analysis, M.M.M. and M.A.G.R.; Investigation, K.-C.L.; Resources, D.B.K.; Data curation, M.M.M. and M.A.G.R.; Writing—original draft, M.M.M.; Writing—review & editing, M.A.G.R., K.-C.L. and D.B.K.; Visualization, M.M.M.; Supervision, D.B.K.; Project administration, D.B.K.; Funding acquisition, D.B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by the National Science Foundation under Grant No. 2015693. This work was supported by the National Research Foundation (NRF), South Korea, under project BK21 FOUR (Smart Robot Convergence and Application Education Research Center) (5120201213760). The authors of this paper appreciate the continuous support provided by the Center for Manufacturing Research (CMR) and the Department of Manufacturing and Engineering Technology at Tennessee Technological University.

Data Availability Statement: The original contributions presented in this study are included in the article/Supplementary Materials. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Agarwala, M.; Bourell, D.; Beaman, J.; Marcus, H.; Barlow, J. Direct selective laser sintering of metals. *Rapid Prototyp. J.* **1995**, *1*, 26–36. [CrossRef]
2. Gibson, I.; Rosen, D.; Stucker, B.; Khorasani, M.; Rosen, D.; Stucker, B.; Khorasani, M. *Additive Manufacturing Technologies*; Springer: Berlin/Heidelberg, Germany, 2021.
3. Jadhav, S.; Jeong, G.H.; Bajestani, M.S.; Islam, S.; Lee, H.; Cho, Y.T.; Kim, D.B. Investigation of surface roughness, microstructure, and mechanical properties of overhead structures fabricated by wire + arc additive manufacturing. *Int. J. Adv. Manuf. Technol.* **2024**, *131*, 5001–5021. [CrossRef]
4. Jadhav, S.; Bajestani, M.S.; Islam, S.; Karim, M.A.; Kim, C.J.; Lee, H.; Cho, Y.T.; Kim, D.B. Materials characterization of Ti6Al4V to NbZr1 bimetallic structure fabricated by wire arc additive manufacturing. *Mater. Today Commun.* **2023**, *36*, 106934. [CrossRef]
5. Taşdemir, A.; Nohut, S. An overview of wire arc additive manufacturing (WAAM) in shipbuilding industry. *Ships Offshore Struct.* **2021**, *16*, 797–814. [CrossRef]
6. Pant, H.; Arora, A.; Gopakumar, G.S.; Chadha, U.; Saeidi, A.; Patterson, A.E. Applications of wire arc additive manufacturing (WAAM) for aerospace component manufacturing. *Int. J. Adv. Manuf. Technol.* **2023**, *127*, 4995–5011. [CrossRef]
7. Viliš, J.; Pokorný, Z.; Zouhar, J.; Vítek, R.; Fornůsek, T. Comparative Analysis of Metal Components Manufactured by WAAM Technology for the Strike Face Layer of Ballistic Protection. *Mater. Sci. Forum* **2025**, *1151*, 3–12. [CrossRef]
8. Wang, Y.; Xu, X.; Zhao, Z.; Deng, W.; Han, J.; Bai, L.; Liang, X.; Yao, J. Coordinated monitoring and control method of deposited layer width and reinforcement in WAAM process. *J. Manuf. Process.* **2021**, *71*, 306–316. [CrossRef]
9. So, M.S.; Mahdi, M.M.; Kim, D.B.; Shin, J. Prediction of Metal Additively Manufactured Bead Geometry Using Deep Neural Network. *Sensors* **2024**, *24*, 6250. [CrossRef]
10. Xia, C.; Pan, Z.; Polden, J.; Li, H.; Xu, Y.; Chen, S.; Zhang, Y. A review on wire arc additive manufacturing: Monitoring, control and a framework of automated system. *J. Manuf. Syst.* **2020**, *57*, 31–45. [CrossRef]
11. Hamrani, A.; Agarwal, A.; Allouhi, A.; McDaniel, D. Applying machine learning to wire arc additive manufacturing: A systematic data-driven literature review. *J. Intell. Manuf.* **2024**, *35*, 2407–2439. [CrossRef]
12. Guo, Y.; Zhang, Y.; Pan, Z.; Zhou, W. Recent progress of sensing and machine learning technologies for process monitoring and defects detection in wire arc additive manufacturing. *J. Manuf. Process.* **2024**, *125*, 489–511. [CrossRef]
13. Mattera, G.; Nele, L.; Paolella, D. Monitoring and control the Wire Arc Additive Manufacturing process using artificial intelligence techniques: A review. *J. Intell. Manuf.* **2024**, *35*, 467–497. [CrossRef]
14. Mahdi, M.M.; Bajestani, M.S.; Do Noh, S.; Kim, D.B. Digital twin-based architecture for wire arc additive manufacturing using OPC UA. *Robot. Comput. Integr. Manuf.* **2025**, *94*, 102944. [CrossRef]
15. Balasubramanian, A. *Integration of Artificial Intelligence and Blockchain in Robotic Wire Arc Additive Manufacturing: A Comprehensive Review and Patent Landscape Analysis*; SSRN: New York, NY, USA, 2024.
16. Wattanapanit, S. Application of Federated Machine Learning for Cross-Platform Knowledge Sharing in Distributed Additive Manufacturing Environments. *Open J. Robot. Auton. Decis.-Mak. Hum.-Mach. Interact.* **2025**, *10*, 1–11.
17. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.

18. Aggarwal, M.; Khullar, V.; Goyal, N. A comprehensive review of federated learning: Methods, applications, and challenges in privacy-preserving collaborative model training. In *Applied Data Science and Smart Systems*; CRC Press: Boca Raton, FL, USA, 2024; pp. 570–575.
19. Eslaminia, A.; Meng, Y.; Nahrstedt, K.; Shao, C. Federated domain generalization for condition monitoring in ultrasonic metal welding. *J. Manuf. Syst.* **2024**, *77*, 1–12. [CrossRef]
20. Leng, J.; Li, R.; Xie, J.; Zhou, X.; Li, X.; Liu, Q.; Chen, X.; Shen, W.; Wang, L. Federated learning-empowered smart manufacturing and product lifecycle management: A review. *Advanced Engineering Informatics* **2025**, *65*, 103179. [CrossRef]
21. Sagar, K.G. Federated Learning in Digital Manufacturing: A Review. *IUP J. Mech. Eng.* **2024**, *17*, 24–40.
22. Mehta, M.; Shao, C. Federated learning-based semantic segmentation for pixel-wise defect detection in additive manufacturing. *J. Manuf. Syst.* **2022**, *64*, 197. [CrossRef]
23. Aggour, K.S.; Kumar, V.S.; Cuddihy, P.; Williams, J.W.; Gupta, V.; Dial, L.; Hanlon, T.; Gambone, J.; Vinciquerra, J. Federated multimodal big data storage & analytics platform for additive manufacturing. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; IEEE: New York, NY, USA, 2019; pp. 1729–1738.
24. Shi, Z.; Li, Y.; Liu, C. Knowledge distillation-based information sharing for online process monitoring in decentralized manufacturing system. *J. Intell. Manuf.* **2025**, *36*, 2177–2192. [CrossRef]
25. Russell, M.; Wang, P. Maximizing model generalization for machine condition monitoring with Self-Supervised Learning and Federated Learning. *J. Manuf. Syst.* **2023**, *71*, 274–285. [CrossRef]
26. Kim, D.B.; Bajestani, M.S.; Lee, J.Y.; Shin, S.; Kim, G.; Sajadieh, S.M.M.; Noh, S. Human-in-the-loop in smart manufacturing (H-SM): A review and perspective. *J. Manuf. Syst.* **2025**, *82*, 178–199. [CrossRef]
27. Huong, T.T.; Bac, T.P.; Ha, K.N.; Hoang, N.V.; Hoang, N.X.; Hung, N.T.; Tran, K.P. Federated Learning-Based Explainable Anomaly Detection for Industrial Control Systems. *IEEE Access* **2022**, *10*, 53854. [CrossRef]
28. Da Silveira Dib, M.A.; Ribeiro, B.; Prates, P. Federated Learning as a Privacy-Providing Machine Learning for Defect Predictions in Smart Manufacturing. *Smart Sustain. Manuf. Syst.* **2021**, *5*, 1–17. [CrossRef]
29. Chen, D.; Wang, D.; Zhu, Y.; Han, Z. Digital Twin for Federated Analytics Using a Bayesian Approach. *IEEE Internet Things J.* **2021**, *8*, 16301. [CrossRef]
30. Kanagavelu, R.; Li, Z.; Samsudin, J.; Yang, Y.; Yang, F.; Siow, R.; Goh, M.; Cheah, M.; Wiwatphonthana, P.; Akkarajitsakul, K.; et al. Two-Phase Multi-Party Computation Enabled Privacy-Preserving Federated Learning. In Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, VIC, Australia, 11–14 May 2020.
31. Gao, W.; Zhao, Z.; Min, G.; Ni, Q.; Jiang, Y. Resource Allocation for Latency-Aware Federated Learning in Industrial Internet of Things. *IEEE Trans. Ind. Inf.* **2021**, *17*, 8505. [CrossRef]
32. Brik, B.; Messaadia, M.; Sahnoun, M.; Bettayeb, B.; Benatia, M.A. Fog-supported Low-latency Monitoring of System Disruptions in Industry 4.0: A Federated Learning Approach. *ACM Trans. Cyber-Phys. Syst.* **2022**, *6*, 1–23. [CrossRef]
33. Kusiak, A. Federated explainable artificial intelligence (fXAI): A digital manufacturing perspective. *Int. J. Prod. Res.* **2023**, *62*, 171–182. [CrossRef]
34. Paramartha Putra, M.A.; Rachmawati, S.M.; Alief, R.N.; Chijioke Ahakonye, L.A.; Gohil, A.; Kim, D.; Lee, J. Federated Learning-Enabled Digital Twin for Smart Additive Manufacturing Industry. In Proceedings of the 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Bali, Indonesia, 20–23 February 2023; pp. 806–811. [CrossRef]
35. Verma, P.; Breslin, J.G.; O'shea, D. FLDID: Federated Learning Enabled Deep Intrusion Detection in Smart Manufacturing Industries. *Sensors* **2022**, *22*, 8974. [CrossRef]
36. Sun, F.; Diao, Z. Federated Learning and Blockchain-Enabled Intelligent Manufacturing for Sustainable Energy Production in Industry 4.0. *Processes* **2023**, *11*, 1482. [CrossRef]
37. Liu, Y.; Kang, Y.; Zou, T.; Pu, Y.; He, Y.; Ye, X.; Ouyang, Y.; Zhang, Y.; Yang, Q. Vertical Federated Learning: Concepts, Advances, and Challenges. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3615–3634. [CrossRef]
38. Zhang, Z.; He, N.; Li, Q.; Wang, K.; Gao, H.; Gao, T. DetectPMFL: Privacy-Preserving Momentum Federated Learning Considering Unreliable Industrial Agents. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7696–7706. [CrossRef]
39. Ranathunga, T.; Mcgibney, A.; Rea, S.; Bharti, S. Blockchain-Based Decentralized Model Aggregation for Cross-Silo Federated Learning in Industry 4.0. *IEEE Internet Things J.* **2023**, *10*, 4449–4461. [CrossRef]
40. Kuo, T.; Yang, H. Federated learning on distributed and encrypted data for smart manufacturing. *J. Comput. Inf. Sci. Eng.* **2024**, *24*, 071007. [CrossRef]
41. Li, W.; Fan, K.; Yang, K.; Yang, Y.; Li, H. PBFL: Privacy-Preserving and Byzantine-Robust Federated-Learning-Empowered Industry 4.0. *IEEE Internet Things J.* **2024**, *11*, 7128–7140. [CrossRef]
42. Zhang, L.; Wu, S.; Xu, H.; Liu, Q.; Hong, C.S.; Han, Z. Optimizing Tradeoff Between Learning Speed and Cost for Federated-Learning-Enabled Industrial IoT. *IEEE Internet Things J.* **2024**, *11*, 11663–11678. [CrossRef]
43. Kang, M.; Lee, D.; Bajestani, M.S.; Kim, D.B.; Noh, S.D. Edge Computing-Based Digital Twin Framework Based on ISO 23247 for Enhancing Data Processing Capabilities. *Machines* **2025**, *13*, 19. [CrossRef]

44. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *8*, 679–698.
45. Ballard, D.H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* **1981**, *13*, 111–122. [CrossRef]
46. Mahdi, M.M.; Nyeem, H. Data-driven Embedding with Pixel Repetition for High Capacity Reversible Data Hiding. In Proceedings of the 2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 18–20 November 2021; IEEE: New York, NY, USA, 2021; pp. 1–4.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.