IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Byzantine-Robust Federated Learning over Ring-All-Reduce Distributed Computing

Sathe Rohit Adinath[1], Prof. D. S. Kshirsagar[2], Prof. N. M. Kolhe[3], Dr. B. H Jadhav[4]
Student, Department of Computer Engineering
Guide, Department of Computer Engineering
Co-Guide, Department of Computer Engineering
Adsul Technical Campus Chas, Ahilyanagar, India

Abstract: *Federated learning enables multiple clients to collaboratively train a shared machine-learning model without directly poexchanging their raw data. In distributed implementations, Ring-All-Reduce is commonly used to aggregate model updates efficiently, but its standard aggregation operations can be vulnerable when one or more participants send corrupted, manipulated, or intentionally malicious updates. This paper presents a conceptual framework for combining Byzantine-resilient update validation validation with Ring-All-Reduce communication. The proposed design uses lightweight local checks, robust aggregation at designated validation points, consistency monitoring, and recovery procedures to reduce the influence of faulty participants while retaining the communication advantages of ring-based training. The paper describes the problem, reviews relevant concepts, outlines the architecture and algorithm, and discusses evaluation metrics and practical limitations. It does not claim experimental results; instead, it provides a reproducible evaluation plan for comparing the proposed approach with ordinary Ring-All-Reduce an aggregation baselines. The central objective is to improve the reliability of distributed federated training under adversarial or unreliable client behaviour without unnecessarily exposing private training data.*

Keywords: Federated Learning; Byzantine Fault Tolerance; Ring-All-Reduce; Distributed Computing; Robust Aggregation; Model security

I. INTRODUCTION

Federated learning (FL) is a distributed machine-learning approach in which multiple clients train a shared model while keeping their raw data locally. A coordinating process combines information derived from local training, allowing collaboration across devices or organisations that cannot conveniently pool their datasets. This structure can support privacy-conscious learning, although it does not automatically guarantee privacy or security.

Large-scale training often relies on collective communication operations to exchange and combine model updates. Ring-All-Reduce arranges participating workers in a logical ring and performs reduction and distribution phases so that each worker can obtain the aggregated result. Its structured communication can be efficient for synchronising large parameter vectors, particularly when the network and participants behave reliably.

A key limitation is that ordinary aggregation may assume that all participants contribute valid updates. A Byzantine participant—whether compromised, faulty, or malicious—may submit arbitrary values, scale its update, or coordinate with other participants to influence the global model. In a ring-based system, the impact can be difficult to diagnose if validation and monitoring are not explicitly included in the protocol.

This paper explores a Byzantine-robust federated-learning design over Ring-All-Reduce. The intended contribution is a system-level framework that adds update validation, robust aggregation checkpoints, anomaly monitoring, and recovery mechanisms while preserving ring-based communication where possible. The design is conceptual and requires empirical testing before any performance or security advantage can be asserted.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40359

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

The remainder of the paper is organised as follows. Section II discusses the background and related concepts.Section III defines the problem and threat model. Section IV presents the proposed architecture and workflow. Section V outlines an algorithm. Section VI describes an evaluation plan, Section VII discusses limitations, and Section VIII concludes the paper.

II. BACKGROUND AND RELATED CONCEPTS

A. Federated Learning

In a typical federated-learning round, a server or distributed coordination layer sends the current model to selected clients. Each client performs local optimisation using its own data and returns a model update or model parameters. The system aggregates these contributions to produce the next global model. Because client datasets may differ substantially, the update distribution can be Research Paper | Byzantine-Robust Federated Learning Page 2 non-identically distributed (non-IID), making simple outlier detection challenging.

B. Ring-All-Reduce

Ring-All-Reduce is a collective communication pattern commonly implemented using a reduce-scatter phase followed by an all-gather phase. Workers communicate with neighbours in a logical ring rather than sending every parameter vector directly to a central server. The approach can distribute communication load, but its standard reduction operation is not inherently Byzantine-robust: a malicious or faulty contribution may affect the aggregate unless additional safeguards are applied.

C. Byzantine Behaviour and Robust Aggregation

Byzantine behaviour includes arbitrary or strategically manipulated updates, such as sign-flipped gradients, unusually large updates, random noise, or coordinated contributions intended to shift the model. Robust aggregation methods attempt to limit such influence using statistics or distance-based rules. Examples discussed in the literature include coordinate-wise median, trimmed mean, and multi-update distance methods. Their assumptions differ, and performance may deteriorate when honest clients have highly heterogeneous data or when adversaries coordinate.

D. Security and Privacy Considerations

Keeping raw data local reduces the need to centralise sensitive datasets, but model updates can still leak information or be manipulated. Robustness, privacy, availability, and communication efficiency are distinct properties. A design should therefore avoid claiming that Byzantine resilience alone guarantees privacy, and should assess the costs of validation, extra communication, and any trusted coordination component.

III. PROBLEM STATEMENT AND THREAT MODEL

The problem considered is collaborative model training in which a set of clients exchange updates through a Ring-All-Reduce communication topology, while a limited subset of clients may be faulty or adversarial. The objective is to prevent invalid contributions from dominating the global update and to detect suspicious behaviour without requiring all raw client data to leave local devices.

The threat model assumes that some participants may submit arbitrary update vectors or behave inconsistently across rounds.The system may observe update norms, selected summary statistics, validation loss on permitted reference data, and consistency across rounds. It does not assume that every malicious update can be identified perfectly. In particular, honest clients with unusual or non-IID data may naturally produce updates that appear statistically different.

The design goals are: (1) limit the influence of extreme or inconsistent updates; (2) retain the communication benefits of Ring-All-Reduce when practical; (3) detect and recover from faulty contributions; (4) minimise additional computation and communication; and (5) report robustness and utility trade-offs transparently.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40359

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

IV. PROPOSED SYSTEM ARCHITECTURE

The proposed framework adds a robustness layer to the distributed training pipeline. Each worker trains locally and performs pobasic sanity checks before its update enters the collective communication process. At configured validation checkpoints, the system estimates whether updates are consistent with expected bounds and applies a robust aggregation rule to suspicious or untrusted contributions. Monitoring records are used to identify repeated anomalies, while a recovery controller can request retransmission, exclude a contribution for a round, or restart aggregation from a verified checkpoint.

A. Local Update Validation

Before transmission, a client checks that its update contains finite values, has the expected dimensions, and does not exceed configured norm limits. Clipping can reduce the effect of excessively large updates, but clipping thresholds must be selected carefully: overly strict thresholds may suppress legitimate learning signals, especially for clients with non-IID data.

B. Robust Aggregation Checkpoints

Instead of assuming that the ordinary sum or average is safe, the system can apply a robust rule at a designated aggregation stage. Depending on the deployment, this may be implemented by a trusted coordinator, a secure aggregation service with suitable support, or a distributed protocol that explicitly tolerates Byzantine participants. The exact mechanism must match the threat model; ordinary Ring-All-Reduce alone does not provide this guarantee.

C. Consistency Monitoring and Recovery

The monitoring component compares update norms, direction similarity, round-to-round changes, and available validation signals. A single anomaly should not automatically prove malicious intent. The recovery component can flag a worker for additional checks, ignore an update for one round, or restore a previous verified model if training quality drops sharply. Decisions should be logged and auditable.

D. Privacy-Aware Operation

The framework should avoid collecting raw examples. Any update statistics retained for monitoring should be minimised and protected according to the deployment's privacy requirements. If secure aggregation hides individual updates, the system must reconcile that protection with robust filtering, because many robust aggregation rules require access to per-client contributions or specialised cryptographic protocols.

V. PROPOSED WORKFLOW AND ALGORITHM

The workflow begins with initial model distribution and local training. Each participant computes an update, validates its format and numerical range, and joins the collective communication round. At robustness checkpoints, the system applies the selected Byzantine-resilient rule and checks the resulting model against agreed monitoring criteria. The global model is accepted if the checks pass; otherwise, the round is investigated or repeated under a recovery policy.

Algorithm 1: Robust Ring-Based Federated Training

Input: initial model w, participating clients C, number of rounds T, local learning procedure, robust aggregation rule R, validation policy V.

1. For each round t = 1, …, T, distribute the current model w to participating clients.
2. Each client k trains locally and computes update Δ.
3. Validate update shape and finite values; apply configured clipping and record minimal diagnostic statistics. 4. Execute the configured Ring-All-Reduce communication procedure for the eligible updates.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

5. At the robustness checkpoint, apply R to the contributions available under the chosen protocol; do not treat a plain sum as
Byzantine-resistant.
6. Evaluate the candidate model using V and compare monitoring signals with the recovery policy.
7. If checks pass, accept the candidate model as w; otherwise, flag the round and apply the predefined recovery action.
8. Repeat until T rounds are complete, then report model quality, robustness, communication cost, and false-positive rate.

This algorithm is a high-level design. A practical implementation must specify where robust aggregation occurs, what information each worker can observe, how adversarial participants are bounded, and how the chosen protocol handles collusion and network failures.

VI. EVALUATION PLAN

A reproducible evaluation should compare the proposed design against at least two baselines: ordinary Ring-All-Reduce without robust filtering, and a federated-learning baseline using a robust aggregation method without the ring communication optimisation. The same model architecture, data split, training budget, and random seeds should be used where possible.

Suggested experimental conditions include (1) no malicious clients, (2) a small fraction of arbitrary-update clients, (3) sign-flipping or scaled updates, (4) random-noise updates, and (5) heterogeneous non-IID client data. The proportion of adversarial clients and attack strength should be varied systematically. Experiments should be conducted only in a controlled test environment using datasets for which the researcher has permission to run the study.

The primary metrics are test accuracy or task-specific quality, convergence speed, communication volume, round duration, additional compute cost, and robustness under each attack. Security-related metrics can include the degradation in model quality under attack and the rate at which honest clients are incorrectly flagged. Results should include multiple runs and variability measures rather than a single best result.

No measured results are asserted in this paper. The evaluation plan is intended to guide implementation and to prevent unsupported claims. The proposed approach should be considered successful only if experiments show a meaningful reduction in Research Paper | Byzantine-Robust Federated Learning Page 4 attack impact while keeping utility and overhead within the deployment's requirements.

VII. CONCLUSION

This paper presented a conceptual framework for Byzantine-robust federated learning over Ring-All-Reduce distributed computing. The design combines local update checks, robust aggregation checkpoints, consistency monitoring, and recovery procedures to reduce the risk that faulty or malicious participants dominate training. Ring-All-Reduce can provide an efficient 0communication structure, but it must not be mistaken for a Byzantine-resilient aggregation protocol. Future work should implement the framework, define a precise adversary model, and evaluate accuracy, attack resistance, communication overhead, and false-positive behaviour under controlled conditions.

REFERENCES

[1] B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in Proceedings of AISTATS, 2017.
[2] J. Konený et al., “Federated Learning: Strategies for Improving Communication Efficiency,” arXiv preprint arXiv:1610.05492, 2016.
[3] P. Blanchard et al., “Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent,” in Advances in Neural Information Processing Systems, 2017.
[4] D. Yin et al., “Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates,” in Proceedings of ICML, 2018.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

[5] L. Barroso, J. Dean, and U. Holzle, "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines," Morgan & Claypool, 2009.
[6] M. Li et al., "Scaling Distributed Machine Learning with the Parameter Server," in Proceedings of OSDI, 2014. DOI: 10.48175/568