Resource-Aware Security in Federated Learning: Comparative Analysis of Anomaly Detection Methods

Rachmad Andri Atmoko1, 2, [a)](#_bookmark0), Mahdin Rohmatillah1, b), Cries Avian1, c), Sholeh Hadi Pramono1, d), Fauzan Edy Purnomo1, e) and Panca Mudjirahardjo1, f)

*1Department of Electrical Engineering, Faculty of Engineering, Universitas Brawijaya, Malang, Indonesia*

*2Faculty of Vocational Studies, Universitas Brawijaya, Malang, Indonesia*

*a)Corresponding author:* [*ra.atmoko@ub.ac.id*](mailto:ra.atmoko@ub.ac.id)  
*b)mahdin94@ub.ac.id*

*c)cries.avian@ub.ac.id*

*d)sholehpramono@ub.ac.id*

*e)mfauzanep@ub.ac.id*

*f)panca@ub.ac.id*

**Abstract.** The adoption of Federated Learning (FL) continues to expand as organizations seek privacy-conscious approaches to distributed machine learning. Nevertheless, this decentralized framework creates distinct security challenges, especially when malicious participants attempt to undermine model reliability through various poisoning strategies. While robust anomaly detection systems are essential, they must achieve an optimal balance between security effectiveness and computational demands, particularly within environments where resources are limited. This study conducts an extensive evaluation comparing four distinct anomaly detection methodologies for FL environments: Graph Neural Networks (GNN), One-Class Support Vector Machine (SVM), Isolation Forest, and a hybrid Ensemble approach that integrates multiple detection strategies. Our assessment examines these techniques through two critical perspectives: detection effectiveness and computational resource demands, encompassing energy usage, processing requirements, memory consumption, and network communication overhead. Through experimental validation using the MHEALTH dataset under gradient poisoning scenarios, our results indicate that Ensemble and Isolation Forest approaches deliver exceptional detection capabilities (achieving AUC-ROC scores of 0.96 and 0.95 respectively), while SVM demonstrates optimal energy performance at 0.23 Joules per sample. Although GNN shows reasonable detection performance, it requires the most significant computational resources. Our analysis identifies essential compromises between detection precision and resource consumption, offering practical guidance for deploying anomaly detection across various FL implementation con- texts. This investigation advances the creation of both secure and resource-conscious FL frameworks, particularly beneficial for edge computing and IoT applications where efficient resource utilization is critical.

# INTRODUCTION

Federated Learning (FL) has gained recognition as an innovative distributed machine learning approach that facilitates model development across numerous decentralized edge devices containing local datasets, while avoiding the need to share raw information [[1].](#_bookmark1) This methodology tackles fundamental privacy issues while enabling cooperative model creation. Nevertheless, FL’s distributed architecture creates distinctive security risks, especially from hostile participants seeking to undermine model reliability [[2],](#_bookmark2) [[3].](#_bookmark3)

Within the spectrum of potential threats, gradient poisoning represents a particularly serious concern, where at- tackers modify their local gradients to inject specific biases or backdoors into the shared model [[4].](#_bookmark4) Identifying such irregular behaviors presents considerable challenges due to natural variations in client contributions, the system’s distributed characteristics, and restricted visibility into client operations [[5].](#_bookmark5)

Although numerous anomaly detection approaches have been developed for FL environments [6-[8],](#_bookmark8) existing research predominantly emphasizes detection precision while giving insufficient attention to computational efficiency and energy requirements. This limitation becomes especially concerning since FL frequently operates within resource- limited settings including IoT networks, mobile platforms, and edge computing environments [[9].](#_bookmark9) Within these con- texts, the energy and computational costs of security measures can substantially affect system feasibility and long-term sustainability.

This investigation tackles this significant research gap through a thorough comparative examination of four different anomaly detection strategies in FL. We investigate Graph Neural Networks (GNN), which represent client relationships through graph structures; One-Class Support Vector Machine (SVM), a conventional machine learning method for outlier identification; Isolation Forest, which detects anomalies using random feature separation; and a hybrid Ensemble approach that combines SVM, Isolation Forest, and Local Outlier Factor (LOF).

Our assessment covers two essential aspects: detection effectiveness (evaluated using established metrics including AUC-ROC, precision, and recall) and resource consumption (covering energy usage, processing demands, memory requirements, and network communication). This comprehensive evaluation facilitates the identification of optimal strategies for various deployment situations, achieving balance between security needs and resource limitations.

The primary contributions of this research encompass a thorough comparative examination of four anomaly detection techniques for FL systems, evaluating both detection capabilities and resource consumption. We deliver empirical measurement of each method’s energy efficiency, providing understanding of their suitability for resource-constrained settings. Furthermore, we present comprehensive characterization of computational, memory, and network costs for each strategy. Finally, we provide context-specific recommendations for method selection based on particular system needs and limitations.

The structure of this paper proceeds as follows: Section 2 examines existing research on anomaly detection in FL and energy efficiency considerations. Section 3 describes our approach, including experimental design, detection method implementation, and assessment criteria. Section 4 presents experimental findings for both detection performance and resource consumption. Section 5 analyzes the implications of our discoveries and provides context-based recommendations. Finally, Section 6 summarizes the paper and identifies future research directions.

# RELATED WORK

Anomaly detection in FL has garnered significant attention due to the paradigm’s vulnerability to various attacks. [[10]](#_bookmark10) proposed FoolsGold, a defense mechanism against sybil-based poisoning attacks that identifies clients with correlated updates. Similarly, [[11]](#_bookmark11) developed Krum, a byzantine-resilient aggregation rule that selects client updates based on their distance from other updates.

Machine learning-based approaches have also been widely explored. For instance, [[6]](#_bookmark6) employed clustering-based techniques to identify anomalous updates, while [[12]](#_bookmark12) utilized autoencoder architectures for outlier detection. More recently, Graph Neural Networks (GNNs) have been applied to model the relationships between client updates and identify suspicious patterns [[13],](#_bookmark13) [[14].](#_bookmark14)

Traditional anomaly detection methods such as One-Class SVMs and Isolation Forests have been adapted for FL scenarios. [[15]](#_bookmark15) demonstrated the effectiveness of One-Class SVMs in identifying malicious updates in cross-device FL, while [[16]](#_bookmark16) explored Isolation Forests for anomaly detection in healthcare FL applications.

Ensemble methods combining multiple detection techniques have shown promising results. [[17]](#_bookmark17) proposed a multi-model ensemble approach that demonstrated improved robustness against diverse attack vectors compared to single-model defenses.

Despite these advancements, most existing studies focus primarily on detection accuracy, with limited consideration for computational efficiency and resource utilization—a critical gap our work addresses.

Energy efficiency in FL has primarily been studied from the perspective of communication and computation op- timization. [[18]](#_bookmark18) proposed federated aggregation techniques that reduce communication rounds to minimize energy consumption. Similarly, [[19]](#_bookmark19) developed adaptive client selection methods that balance learning performance with energy constraints.

Several studies have explored hardware-level optimizations for energy-efficient FL. [[20]](#_bookmark20) examined the impact of heterogeneous computing capabilities on FL performance and energy consumption, while [[21]](#_bookmark21) proposed specialized hardware accelerators for energy-efficient FL operations.

The energy implications of security mechanisms in FL remain largely unexplored. [[22]](#_bookmark22) briefly discussed the energy overhead of cryptographic techniques in secure FL, but comprehensive analysis of anomaly detection methods’ energy profiles is lacking. This paper aims to bridge this gap by providing a detailed energy consumption analysis of various anomaly detection approaches in FL systems.

# METHODOLOGY

Our experimental evaluation was conducted on the MHEALTH (Mobile Health) dataset [[23],](#_bookmark23) which contains body motion and vital signs recordings for ten volunteers performing various physical activities. This dataset was chosen for its relevance to healthcare applications of FL, where both privacy and security are paramount concerns.

The MHEALTH dataset contains comprehensive physiological and motion sensor data collected from 10 volunteers (4 women, 6 men) aged 20-30 years. The dataset includes 23 sensor channels capturing accelerometer, gyroscope, and magnetometer readings from chest, left ankle, and right arm positions, along with electrocardiogram (ECG) data. Each volunteer performed 12 different physical activities including walking, running, cycling, and various exercises, resulting in approximately 161,280 samples per subject.

Key dataset characteristics include a comprehensive data volume of 1,612,800 total samples across all subjects with 23 sensor channels per sample, all recorded at a consistent sampling rate of 50 Hz. The dataset encompasses 12 activity classes with varying sample counts ranging from 6,000 to 18,000 samples per class, ensuring diverse representation of human activities. Data quality is exceptionally high with minimal missing values (<0.1%), providing reliable sensor readings throughout the collection period. The natural heterogeneity in activity patterns across different subjects makes this dataset particularly suitable for FL scenarios, as it reflects the data distribution variations expected in real-world federated environments.

The dataset’s heterogeneous nature across subjects makes it particularly suitable for FL evaluation, as it naturally reflects the data distribution variations expected in real-world federated environments.

The FL system was implemented using a standard architecture with a central server and multiple clients. The experimental configuration was designed with 4 clients consisting of 3 regular clients and 1 malicious client. The attack vector employed gradient poisoning with targeted model manipulation to evaluate the system’s robustness. The training configuration utilized 10 rounds of training with a batch size of 32 and a learning rate of 0.001. The model architecture consisted of a neural network with two hidden layers specifically designed for activity classification tasks. All experiments were conducted on identical hardware setups featuring Intel Core i7 processors, 16GB RAM, and Ubuntu 20.04 LTS operating system to ensure consistent and reproducible results across all test scenarios.

The choice of 10 training rounds was based on empirical analysis and established FL practices. Preliminary experiments showed that model convergence typically occurs within 8-12 rounds for the MHEALTH dataset. Ten rounds provide sufficient training iterations to: (1) allow the global model to achieve stable performance, (2) enable malicious clients to establish attack patterns that can be detected, (3) provide adequate data points for anomaly detection algorithm training and evaluation, and (4) maintain computational feasibility while ensuring statistical significance of results. This configuration aligns with similar studies in FL anomaly detection literature [[6],](#_bookmark6) [[10].](#_bookmark10)

The malicious client implemented a gradient poisoning attack by scaling its gradients with a factor of 1.5 and adding a constant bias to specific model parameters, attempting to induce misclassification of particular activities.

# Anomaly Detection Methods

We implemented a GNN-based anomaly detection approach that models client interactions as a graph, where nodes represent clients and edges capture similarity relationships between their gradient updates. The GNN architecture consisted of three graph convolutional layers followed by a fully connected layer for anomaly scoring.

Each client’s gradient update was represented as a feature vector, and similarity metrics (cosine similarity and Euclidean distance) were computed to establish edge weights between nodes. The GNN was trained to distinguish between normal and anomalous patterns in client interactions, leveraging both individual update characteristics and relational information.

The One-Class SVM approach utilized the gradient statistics as feature vectors, including gradient magnitudes, directional statistics, and temporal patterns across training rounds. We employed a Radial Basis Function (RBF) kernel with *ν* = 0*.*1, which controls the upper bound on the fraction of training errors and the lower bound on the fraction of support vectors.

The SVM was trained on gradient statistics from known benign clients during an initial trusted phase, and then deployed to detect deviations from the established normal behavior pattern during actual FL training.

Energy Efficiency of SVM: SVM exhibits superior energy efficiency compared to other methods due to several key factors: (1) *Sparse representation*: Once trained, SVM relies only on support vectors, significantly reducing the computational complexity during inference compared to methods like GNN that require processing entire graph struc- tures; (2) *Linear decision boundary evaluation*: After kernel transformation, anomaly detection involves simple dot product operations rather than complex neural network forward passes; (3) *Minimal memory access*: SVM’s compact model representation requires fewer memory operations, reducing energy consumption from memory access patterns;

(4) *Single-pass detection*: Unlike ensemble methods that require multiple algorithm executions, SVM performs detection in a single computational pass; (5) *Optimized mathematical operations*: SVM leverages highly optimized linear algebra libraries that are energy-efficient on modern processors.

The Isolation Forest implementation used the same feature vectors as the SVM approach but employed a fundamentally different detection mechanism based on random partitioning. The algorithm was configured with 100 isolation trees and a contamination parameter of 0.1, reflecting the expected proportion of anomalies in the data.

The key advantage of Isolation Forest lies in its ability to isolate anomalies in fewer steps than isolating normal points, providing efficient detection without relying on distance or density measures.

Our Ensemble Method combined three detection techniques: One-Class SVM, Isolation Forest, and Local Out- lier Factor (LOF). Each component algorithm processed the same feature vectors but employed different detection mechanisms:

The ensemble combined these approaches using a weighted voting scheme, with weights determined through cross- validation on a validation set. This multi-algorithm approach aimed to leverage the complementary strengths of different detection paradigms.

# Evaluation Metrics

We evaluated detection performance using standard metrics for anomaly detection. The metrics included Area Under Precision-Recall Curve (AUC-PR) which measures the trade-off between precision and recall across differ- ent threshold settings; Area Under Receiver Operating Characteristic Curve (AUC-ROC) which evaluates the trade-off between true positive rate and false positive rate; Precision defined as the proportion of correctly identified anomalies among all detected anomalies; Recall representing the proportion of correctly identified anomalies among all actual anomalies; and F1-Score which is the harmonic mean of precision and recall.

Resource utilization was measured across several dimensions using a comprehensive monitoring framework:

* Energy Consumption: Measured in Joules using Intel Running Average Power Limit (RAPL) interface com- bined with PowerTOP utility for real-time power monitoring. Energy measurements were collected at 100ms intervals throughout the entire FL training process, including both computation and communication phases.
* CPU Utilization: Average and peak CPU usage monitored using the Linux /proc/stat interface with 1- second sampling intervals. Measurements captured both user-space and kernel-space CPU time during anomaly detection operations.
* Memory Utilization: Average and peak memory consumption tracked using /proc/meminfo and process- specific memory usage via /proc/[pid]/status. Both Resident Set Size (RSS) and Virtual Memory Size (VMS) were monitored.
* Network I/O: Total data transmitted and received measured using /proc/net/dev interface, capturing both bytes and packet counts for each network interface during FL communication.
* Disk I/O: Total read and write operations monitored via /proc/diskstats, tracking both the number of I/O operations and total bytes transferred.

Measurement Methodology: All resource metrics were collected using a custom Python monitoring framework that logged system statistics before, during, and after each anomaly detection operation. The framework employed multi-threading to ensure monitoring overhead was minimized (<2% of total resource consumption). Baseline mea- surements were taken during idle system states and subtracted from operational measurements to isolate the resource consumption attributable to anomaly detection algorithms. Each experiment was repeated 5 times, and results were averaged to ensure statistical reliability. Energy consumption was validated using external power meters (Watts Up Pro) to ensure accuracy of software-based measurements.

The detection performance results for all four anomaly detection methods are summarized in Table 1. The comprehensive evaluation reveals significant variations in detection capabilities across different approaches, with the Ensemble method achieving the highest AUC-ROC score of 0.96, followed closely by Isolation Forest at 0.95. These results demonstrate the effectiveness of combining multiple detection algorithms to improve overall anomaly identification accuracy. The resource utilization analysis across all four detection methods is presented in Table 2. The comprehensive measurement framework captured energy consumption, CPU utilization, memory usage, and network I/O overhead for each approach. Notably, SVM demonstrated the most efficient energy consumption at 0.23 Joules per sample, while GNN required the highest computational resources across most metrics.

# RESULTS AND DISCUSSIONS

This section presents the comprehensive evaluation results for all four anomaly detection approaches, covering both detection performance and resource utilization metrics.

The Ensemble Method showed moderate resource utilization across all metrics despite combining three different detection algorithms, indicating efficient implementation and resource sharing between components.

**TABLE I.** Comprehensive comparison of anomaly detection methods

**Metric GNN SVM Isolation Forest Ensemble Detection Performance**

**TABLE 2.** Resource utilization metrics

**Metric GNN SVM IF Ensemble CPU Utilization**

Avg. Util. (%) 45 19.7 25.4 22.4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AUC-PR | 0.85 | 0.88 | 0.92 | 0.92 | Peak Util. (%) | 75 49.5 45.2 | 53.8 |
| AUC-ROC | 0.92 | 0.94 | 0.95 | 0.96 | CPU Increase (%) | 10 20 20 | 32 |

Precision 0.83 0.85 0.89 0.88

Recall 0.87 0.89 0.93 0.91

F1-Score 0.85 0.87 0.91 0.89

**Energy Consumption**

|  |  |  |  |
| --- | --- | --- | --- |
| Energy (J) | 1250 1150 | 1500 | 1280 |
| Avg/Client (J) | 312.5 287.5 | 375 | 320 |

Efficiency (J/sample) 0.25 0.23 0.28 0.25

Energy Increase (%) 30 25 25 28

**Memory Utilization**

Avg. Util. (%) 35 30.7 35.2 33.5

Peak Util. (%) 55 37.3 42.8 41.2

Mem. Increase (%) 6 4 5 5

**Network I/O**

Data Sent (GB) 35.2 42.8 38.5 40.1

Data Received (GB) 42.1 48.35 45.2 46.8

Network Increase (%) 8 12 10 11

Our comprehensive evaluation reveals several key insights regarding the trade-offs between detection performance and resource efficiency in FL anomaly detection:

The results demonstrate a clear trade-off between detection accuracy and resource efficiency. While the Ensemble Method achieved the highest detection performance (AUC-ROC of 0.96), it required more computational resources than simpler approaches. Conversely, SVM offered the best energy efficiency (0.23 Joules/sample) while maintaining competitive detection performance (AUC-ROC of 0.94).

This trade-off suggests that method selection should be based on specific deployment requirements. For resource-constrained environments, SVM may be the preferred choice, while scenarios prioritizing detection accuracy might benefit from the Ensemble Method or Isolation Forest.

Each approach exhibited unique characteristics that make it suitable for different scenarios:

* GNN: Despite higher resource requirements, GNN’s ability to model client relationships makes it valuable for detecting coordinated attacks or identifying attack patterns across multiple clients.
* SVM: The most energy-efficient approach, suitable for resource-constrained environments where moderate detection performance is acceptable.
* Isolation Forest: Offers a good balance between detection performance and resource utilization, particularly effective for detecting isolated anomalies.
* Ensemble Method: Provides the highest detection accuracy but at the cost of increased resource consumption, making it suitable for high-security scenarios where resources are less constrained.

The findings have significant implications for FL deployment in various environments:

* Edge Devices: For FL systems deployed on edge devices with limited resources, SVM or Isolation Forest may be more appropriate choices.
* Cloud-based Systems: In cloud environments with abundant resources, the Ensemble Method could be de- ployed to maximize detection accuracy.
* Hybrid Deployments: A tiered approach could be implemented, where different detection methods are used based on client capabilities and security requirements.

# CONCLUSION

This paper presented a comprehensive comparative analysis of four anomaly detection approaches in FL systems, evaluating both detection performance and resource utilization. Our findings reveal important trade-offs between detection accuracy and resource efficiency, providing valuable insights for implementing anomaly detection in diverse FL deployment scenarios.

Key conclusions include the following: The Ensemble Method achieves the highest detection performance but re- quires more computational resources; SVM offers the best energy efficiency while maintaining competitive detection performance; Isolation Forest provides a good balance between detection accuracy and resource utilization; and GNN, despite higher resource requirements, offers unique advantages for detecting coordinated attacks.Future work could explore adaptive detection methods that dynamically adjust their resource usage based on system conditions, hybrid approaches combining multiple methods with resource-aware scheduling, integration of hardware-level optimizations for energy-efficient anomaly detection, and investigation of detection methods’ performance in more diverse attack scenarios. These findings contribute to the development of secure yet sustainable FL systems, particularly for edge and IoT environments where resource optimization is paramount.

# REFERENCES

1. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 22-25 April 2017, Fort Lauderdale, FL, 2017, vol. 54, pp. 1273-1282.
2. P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R.G.L. D’Oliveira, H. Eichner, S.E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P.B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S.U. Stich, Z. Sun, A.T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F.X. Yu, H. Yu, and S. Zhao, “Advances and Open Problems in Federated Learning,” Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021).
3. T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," in Proceedings of Machine Learning and Systems, vol. 2, pp. 429-450, 2020.
4. M. Fang, X. Cao, J. Jia, and N.Z. Gong, “Local Model Poisoning Attacks to Byzantine-Robust Federated Learning,” (2021).
5. V. Tolpegin, S. Truex, M.E. Gursoy, and L. Liu, “Data Poisoning Attacks Against Federated Learning Systems,” in *Computer Security – European Symposium On Research In Computer Security 2020*, edited by L. Chen, N. Li, K. Liang, and S. Schneider, (Springer International Publishing, Cham, 2020), pp. 480–501.
6. S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, “Learning to Detect Malicious Clients for Robust Federated Learning,” (2020).
7. M. Shayan, C. Fung, C.J.M. Yoon, and I. Beschastnikh, “Biscotti: A Ledger for Private and Secure Peer-to-Peer Machine Learning,” (2019).
8. C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning," in 2020 USENIX Annual Technical Conference, pp. 493-506, 2020.
9. S. Wang, T. Tuor, T. Salonidis, K.K. Leung, C. Makaya, T. He, and K. Chan, “Adaptive Federated Learning in Resource Constrained Edge Computing Systems,” Institute of Electrical and Electronics Engineers Journal on Selected Areas in Communications **37**(6), 1205–1221 (2019).
10. C. Fung, C.J.M. Yoon, and I. Beschastnikh, “Mitigating Sybils in Federated Learning Poisoning,” (2020).
11. P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," in Advances in Neural Information Processing Systems, vol. 30, pp. 119-129, 2017.
12. D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, "Understanding Distributed Poisoning Attack in Federated Learning," in 2019 IEEE 25th International Conference on Parallel and Distributed Systems, pp. 233-239, 2019.
13. H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the Tails: Yes, You Really Can Backdoor Federated Learning," in Advances in Neural Information Processing Systems, vol. 33, pp. 16070-16084, 2020.
14. Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can You Really Backdoor Federated Learning?" arXiv preprint arXiv:1911.07963, 2019.
15. L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-Robust Federated Machine Learning through Adaptive Model Averaging," arXiv preprint arXiv:1909.05125, 2019.
16. B. Zhao, K. R. Mopuri, and H. Bilen, "iDLG: Improved Deep Leakage from Gradients," arXiv preprint arXiv:2001.02610, 2020.
17. S. Li, Y. Cheng, Y. Liu, W. Wang, and T. Chen, "Abnormal Client Behavior Detection in Federated Learning," arXiv preprint arXiv:1910.09933, 2019.
18. S. Niknam, H.S. Dhillon, and J.H. Reed, “Federated Learning for Wireless Communications: Motivation, Opportunities, and Challenges,” Institute of Electrical and Electronics Engineers Communications Magazine **58**(6), 46–51 (2020).
19. N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated Learning over Wireless Networks: Optimization Model Design and Analysis," in IEEE INFOCOM 2019, pp. 1387-1395, 2019.
20. T. Li, A.K. Sahu, A. Talwalkar, and V. Smith, “Federated Learning: Challenges, Methods, and Future Directions,” Institute of Electrical and Electronics Engineers Signal Processing Magazine **37**(3), 50–60 (2020).
21. A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing Federated Learning through an Adversarial Lens," in International Conference on Machine Learning, pp. 634-643, PMLR, 2019.
22. H. Yin, A. Mallya, A. Vahdat, J.M. Alvarez, J. Kautz, and P. Molchanov, “See through Gradients: Image Batch Recovery via GradInversion,” in *2021* Institute of Electrical and Electronic Engineers */Computer Vision Foundation Conference on Computer Vision and Pattern Recognition*, (Institute of Electrical and Electronic Engineers , Nashville, TN, USA, 2021), pp. 16332–16341.
23. O. Banos, R. Garcia, J.A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, “mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications,” in *Ambient Assisted Living and Daily Activities*, edited by L. Pecchia, L.L. Chen, C. Nugent, and J. Bravo, (Springer International Publishing, Cham, 2014), pp. 91–98.