Transactions on Computational and Scientific Methods | Vo. 5, No. 6, 2025
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Resilient and Context-Aware Edge AI: Bridging Intelligence, Privacy, and Autonomy

Ilva Drechsler
University of Central Arkansas, Conway, USA
Ilva.989@uca.edu

Abstract: Federated learning (FL) is an emerging distributed machine learning paradigm that enables collaborative model training across decentralized data sources while preserving data privacy and reducing communication overhead. This survey presents a comprehensive review of the current state of federated learning from both algorithmic and system perspectives. It begins by examining the core architectures of FL, including client-server and peer-to-peer designs, followed by detailed discussions of aggregation mechanisms, optimization strategies under data heterogeneity, and essential privacy-preserving techniques such as differential privacy and secure aggregation. Real-world applications are analyzed across industries such as healthcare, finance, Internet of Things, and education, demonstrating FL's practical viability. The paper also outlines major open research challenges, including personalization, scalability, communication efficiency, and regulatory constraints. By unifying advances from both academic and industry settings, this work provides a foundational resource for researchers and practitioners aiming to deploy federated learning in production-grade AI systems.

Keywords: Federated learning, decentralized AI, privacy-preserving machine learning, edge computing

1. Introduction

In recent years, the exponential growth of decentralized data generated by personal devices, edge sensors, and distributed platforms has fundamentally challenged traditional machine learning paradigms. Conventional centralized learning approaches require aggregating raw data into a central server for model training, raising significant concerns over privacy, communication efficiency, and regulatory compliance. In response to these limitations, Federated Learning (FL) has emerged as a transformative paradigm that enables collaborative model training across distributed clients without requiring direct access to their raw data [1]. By keeping data local and only sharing model updates, FL preserves user privacy while leveraging the collective intelligence of distributed systems.

First introduced by Google in 2016 to improve keyboard prediction in Android devices [2], FL has since evolved into a core enabling technology for privacy-preserving machine learning. Its applications now span healthcare, finance, smart cities, autonomous vehicles, and beyond. At its core, FL orchestrates a cyclical process in which a central server distributes a global model to participating clients. These clients train the model locally on their private data and send back encrypted or obfuscated model updates, which the server aggregates to refine the global model. This decentralized framework offers several

advantages: it reduces the risk of data leakage, alleviates communication bottlenecks, and facilitates compliance with data protection regulations such as GDPR and HIPAA [3].

However, the implementation of FL is fraught with unique challenges that are not present in traditional centralized training. These include statistical heterogeneity due to non-independent and identically distributed (non-IID) data across clients, system heterogeneity stemming from variations in hardware and connectivity, and privacy vulnerabilities introduced through gradient leakage and inference attacks [4]. Furthermore, FL systems must contend with issues related to client availability, unreliable communication channels, and limited computational resources, particularly in edge or mobile environments [5].

To address these challenges, researchers have proposed a wide array of solutions, including advanced aggregation algorithms like FedAvg, FedProx, and FedNova, as well as privacy-preserving mechanisms based on differential privacy, secure multiparty computation, and homomorphic encryption [6][7]. Moreover, recent developments in personalized federated learning, adaptive client selection, and hierarchical FL architectures have significantly improved the scalability and robustness of federated systems in real-world deployments [8].

The growing interest in FL has led to its adoption by leading technology companies and research institutions. Google has deployed FL in Gboard for next-word prediction, Apple uses it to improve Siri and keyboard suggestions, and NVIDIA incorporates FL in its Clara platform for medical imaging [9]. Meanwhile, open-source frameworks such as TensorFlow Federated, PySyft, and Flower have accelerated academic and commercial exploration of FL applications.

2. Fundamentals of Federated Learning

Federated Learning (FL) represents a shift from centralized model training to a collaborative, decentralized framework. It enables multiple clients—ranging from smartphones to hospitals—to jointly train a shared machine learning model while keeping their local data private. To understand how this process functions in practice, we examine the fundamental system architecture and core mechanisms that underlie typical FL deployments.

2.1 Federated Learning Architectures

FL systems are typically categorized into two primary architectures: cross-device FL and cross-silo FL.

In *cross-device FL*, the clients are large numbers of personal devices such as mobile phones, wearables, or IoT sensors. These systems are designed to tolerate unreliable communication, limited bandwidth, and frequent device dropout. This setting is highly scalable but also introduces challenges such as client availability and data heterogeneity. Google's implementation of FL in Gboard—a mobile keyboard prediction app—was one of the first real-world examples of this model, coordinating thousands of Android devices to train a shared language model without centralizing user text data.

In contrast, *cross-silo FL* involves a smaller number of reliable and often institutionally-managed nodes, such as hospitals, banks, or regional data centers. These systems tend to have better communication capabilities and higher computational resources, which simplifies synchronization and allows for more complex privacy-preserving protocols. Cross-silo FL is often adopted in regulated industries where data centralization is legally restricted, such as healthcare or finance.

Both architecture types require a central server (or coordinator) to manage the training process: initializing models, orchestrating communication rounds, collecting updates, and aggregating results. In

recent developments, decentralized versions of FL—where no central server exists—are also being explored, particularly in peer-to-peer or blockchain-enhanced environments.

2.2 Model Update and Aggregation Mechanisms

At the heart of FL lies the federated averaging (FedAvg) algorithm. In each round, the server distributes the current global model to a subset of participating clients. Each client trains the model on its own local data for one or more epochs and returns the updated model parameters (or gradients). The server then aggregates these updates—usually via a weighted average based on the number of local samples—and updates the global model accordingly.

FedAvg is computationally simple and communication-efficient, making it widely adopted in early FL implementations. However, its performance can degrade in the presence of non-IID data or unbalanced participation, which are common in real-world FL systems. To mitigate these issues, variants such as FedProx introduce proximal terms in the local loss function to restrict divergence from the global model, while FedNova normalizes updates to account for different local training efforts.

Communication overhead is another bottleneck in FL, especially in mobile or rural edge environments. To address this, techniques like update sparsification, quantization, and periodic communication (e.g., local epochs between global rounds) are often employed. These strategies reduce the size and frequency of transmitted updates without severely impacting model performance.

Finally, in more advanced setups, hierarchical aggregation structures are introduced. Here, intermediary nodes (e.g., edge gateways or regional servers) perform local aggregation before passing results to the central server, thereby reducing uplink communication and improving fault tolerance.

In summary, federated learning frameworks are structured around a flexible client-server interaction pattern, tailored to specific device contexts and privacy requirements. Whether deployed across millions of smartphones or a consortium of hospitals, FL architectures must balance scalability, personalization, and system efficiency to meet the diverse needs of distributed AI applications.

3. Privacy and Security Mechanisms in Federated Learning

While federated learning improves data privacy by design — keeping raw data local rather than centralized—it is by no means immune to privacy breaches or malicious manipulation. Because model updates can still leak sensitive information, and the training process itself is exposed to communication and system vulnerabilities, FL systems must be equipped with additional privacy-preserving and security mechanisms to be considered trustworthy and robust.

3.1 Differential Privacy and Gradient Obfuscation

Differential privacy (DP) is one of the most widely adopted techniques for formalizing privacy guarantees in FL. The core idea is to introduce statistical noise into the model updates or gradients in a way that conceals the presence or absence of any individual user’ s data in the training process. When applied correctly, DP ensures that adversaries cannot confidently infer specific attributes or membership of training records, even with access to aggregated model parameters.

In FL, DP can be implemented either locally on the client side (local DP) or globally at the aggregator level. Local DP adds noise to updates before they are sent, offering stronger privacy at the cost of utility degradation. Global DP, on the other hand, aggregates unmodified client updates and then injects noise

into the aggregated result. This often results in a better privacy-utility tradeoff, especially when combined with secure aggregation protocols that prevent the server from seeing individual contributions.

Notably, balancing differential privacy with model accuracy remains an open problem, particularly when dealing with non-IID data distributions. Recent work attempts to calibrate noise adaptively based on client data sensitivity or to integrate DP with adaptive learning algorithms to minimize performance loss.

3.2 Secure Aggregation and Encryption Techniques

To further reduce the risk of data leakage, secure aggregation protocols are used to ensure that the central server can only access the aggregated model updates—not individual client contributions. These protocols leverage cryptographic tools such as homomorphic encryption and secure multiparty computation (SMPC). For example, SMPC allows clients to jointly compute the sum of their model updates without revealing their individual inputs to each other or to the server.

One practical implementation of this approach was introduced by Google, enabling the secure summation of model parameters from thousands of mobile clients while resisting collusion and dropout scenarios. While cryptographic protocols can be computationally expensive, they are increasingly being optimized for edge-scale deployment, balancing security with efficiency.

Homomorphic encryption (HE) offers another path by enabling computation directly on encrypted data. However, fully homomorphic encryption remains computationally prohibitive for most real-time FL applications. Instead, partially homomorphic schemes (e.g., additive HE) are used for summation tasks, although they still impose non-trivial latency and memory overheads.

3.3 Threat Models and Potential Attacks

Even with privacy-preserving mechanisms in place, FL remains susceptible to a variety of adversarial threats:

Inference attacks: Adversaries may reconstruct sensitive features or infer data membership from shared gradients or model updates. This risk is particularly high when models are overparameterized or clients contribute updates sparsely.

Poisoning attacks: Malicious clients may inject manipulated data or gradients into the training process with the goal of corrupting the global model. This includes backdoor attacks, where the model is trained to misclassify inputs with specific patterns while performing normally otherwise.

Byzantine behavior: In decentralized or large-scale federated systems, some clients may behave arbitrarily or inconsistently, either due to software errors or intentional sabotage. Robust aggregation algorithms such as Krum or Bulyan are designed to limit the influence of such clients by detecting and down-weighting outliers.

To counter these threats, a multilayered defense strategy is typically required. This may include anomaly detection during model aggregation, cryptographic audit trails, and hybrid federated learning models that combine secure enclaves with probabilistic trust scoring.

In summary, while federated learning offers a more privacy-conscious alternative to centralized training, its security depends on the careful design and integration of cryptographic, statistical, and systemic safeguards. As FL is deployed in sensitive domains like finance and healthcare, the demand for formal security guarantees and certified compliance will only continue to grow.

4. Optimization and Model Training Techniques in Federated Learning

Effective model training in federated settings poses substantial challenges due to limited communication bandwidth, data heterogeneity, and client resource constraints. To address these issues, researchers have developed a wide range of optimization strategies tailored to the unique characteristics of FL. This section explores several key techniques that enhance training performance, stability, and personalization in federated environments.

4.1 Communication-Efficient Training

One of the primary bottlenecks in federated learning is the cost of communicating large model updates across potentially millions of devices. As such, communication-efficient training has become a central area of research.

A common approach involves increasing the number of local update steps each client performs before sending model updates to the server. This technique, popularized through the FedAvg algorithm, reduces the communication frequency while still achieving convergence in many settings. However, excessive local training can lead to client drift, especially when client data is non-IID, causing local models to diverge from the global objective.

To reduce the size of model updates, gradient compression methods are widely adopted. These include update quantization, sparsification (only sending the top-k significant gradient values), and structured updates using low-rank approximations. For example, deep gradient compression techniques have been shown to reduce communication costs by over 90% without significant loss in model accuracy [1].

Another promising direction is event-triggered communication, where clients only transmit updates when significant local changes are detected. This adaptive strategy avoids unnecessary transmissions and enables more flexible client participation.

4.2 Optimization Under Data Heterogeneity

Federated clients typically have access to local datasets that vary significantly in size, quality, and distribution. This non-IID data scenario breaks many assumptions of classical optimization and may result in slower convergence or biased global models.

To address this, several optimization techniques have been proposed. FedProx introduces a proximal term to the local objective function, penalizing large deviations from the global model and improving stability under data heterogeneity. Similarly, SCAFFOLD incorporates server and client control variates to reduce variance in stochastic gradients and correct for client drift.

Adaptive federated optimization is another line of work that applies adaptive learning rates at either the client or server level. Algorithms such as FedAdam and FedYogi adapt ideas from centralized optimizers (Adam, Yogi) to the federated setting, allowing better control of momentum and gradient variance during aggregation.

4.3 Personalization and Client Adaptation

In many federated learning applications—such as mobile keyboard prediction or medical diagnostics—clients require personalized models that reflect their own usage patterns or patient populations. A one-size-fits-all global model may underperform in such settings.

To support personalization, various strategies have been explored:

a. Fine-tuning: After receiving the global model, each client continues training locally to refine the model for its specific data.

b. Meta-learning approaches: Algorithms such as pFedMe and FedMeta train global models to be easily adaptable to each client’ s data using minimal local updates.

Clustered federated learning: Clients with similar data distributions are grouped, and separate models are trained per cluster to improve local relevance.

Personalization methods help balance the trade-off between model generalizability and user-specific accuracy, a core issue in client-centric federated learning.

In summary, training optimization in FL extends beyond classical algorithm design. It must consider dynamic participation, limited bandwidth, data non-IIDness, and the need for personalization. The ongoing development of communication-efficient, heterogeneity-resilient, and user-adaptive optimization methods is central to making FL viable in production-scale systems.

5. System Design and Scalability in Federated Learning

Real-world deployment of federated learning requires more than algorithmic correctness—it demands careful attention to system-level challenges such as heterogeneity, resource constraints, communication instability, and fault tolerance. This section discusses essential design strategies to build scalable, robust FL systems that can operate across millions of diverse edge devices or institutional nodes.

5.1 Heterogeneity and Client Selection

A defining feature of FL environments is system heterogeneity. Clients differ widely in terms of processing power, memory capacity, energy availability, and network reliability. For instance, an FL system involving smartphones must handle devices ranging from high-end models with ample resources to low-power devices with intermittent connectivity.

This variability complicates synchronous training, where straggler devices can delay entire communication rounds. To address this, many FL systems adopt asynchronous training or use partial client participation, where only a randomly selected subset of available clients participates in each round. This not only reduces synchronization delays but also balances system load and enhances fairness over time.

Client selection strategies are crucial for maintaining both performance and inclusivity. Techniques such as importance sampling prioritize clients with more representative or diverse data, while resource-aware scheduling ensures that slower or energy-constrained devices are not overwhelmed. Some recent works also consider reputation-based selection, excluding clients with suspicious behavior or inconsistent updates, thereby improving both efficiency and security.

5.2 Fault Tolerance and Hierarchical Architectures

Large-scale FL deployments must also contend with unreliable communication and device dropout. A significant number of clients may become temporarily unavailable during a training round due to disconnections, app closures, or system restarts. Systems must be resilient to such disruptions without compromising convergence.

Fault-tolerant aggregation mechanisms, such as dropout-resilient averaging and stochastic aggregation, have been proposed to maintain robustness under partial participation. These techniques allow the server to update the global model even when only a subset of clients return updates. Some implementations also allow late-arriving updates to be integrated asynchronously, smoothing convergence over unstable communication links.

Another powerful design principle is hierarchical federated learning, where clients are grouped under intermediate aggregators — such as edge servers or regional gateways — which perform local model aggregation before forwarding results to a central coordinator. This reduces uplink traffic, localizes learning to similar environments, and improves scalability. Hierarchical FL is especially effective in smart cities, autonomous vehicle networks, and healthcare consortia where edge clusters naturally form.

In addition, model caching and checkpointing mechanisms are employed to enhance system recoverability. Clients store local model states periodically, enabling them to resume training after failure or update loss. Combined with versioning and rollback strategies on the server side, these methods strengthen system reliability and user trust.

In summary, scalable FL system design requires intelligent handling of client diversity, robust aggregation under partial participation, and architecture-aware coordination strategies. By integrating these elements, federated systems can achieve high performance even in highly variable and unreliable environments.

6. Applications of Federated Learning

Federated learning has rapidly expanded beyond academic prototypes to power real-world intelligent systems across a diverse set of industries. Its unique ability to preserve data privacy while enabling collaborative learning has made it particularly attractive in domains that are both data-rich and regulation-sensitive. In this section, we examine four key sectors where FL has demonstrated substantial practical value: healthcare, finance, Internet of Things (IoT), and education.

6.1 Healthcare and Biomedical Applications

In healthcare, FL offers a transformative solution for privacy-preserving collaboration across hospitals, research centers, and diagnostic devices. Medical data—such as imaging scans, electronic health records (EHRs), and genomics — are highly sensitive and often cannot be centralized due to regulatory restrictions like HIPAA and GDPR. FL enables institutions to jointly train diagnostic models without ever exposing patient data.

One prominent example is NVIDIA's Clara FL platform, which enables medical institutions to collaboratively build AI models for tumor detection, brain segmentation, and COVID-19 diagnosis using radiology data from distributed sources . Similarly, the Federated Tumor Segmentation (FeTS) initiative coordinated by Intel and UPenn has shown how FL can be applied across dozens of global institutions to train robust models for glioblastoma segmentation in brain MRIs.

In wearable health monitoring, FL has been used to develop personalized models for arrhythmia detection, sleep stage classification, and glucose trend forecasting on devices like smartwatches and continuous glucose monitors—reducing cloud dependency and improving on-device responsiveness.

6.2 Financial Services and Credit Risk Modeling

In the financial sector, institutions are constrained from sharing customer transaction data, credit histories, or fraud patterns due to legal and competitive concerns. FL provides a way for banks, insurers, and fintech platforms to collaborate on predictive models for credit scoring, fraud detection, and customer segmentation while maintaining strict data boundaries.

For instance, WeBank in China has developed a federated learning framework called FATE (Federated AI Technology Enabler), which supports secure credit scoring across different financial institutions. FATE incorporates vertical FL, enabling model training across different feature spaces owned by different parties (e.g., bank and telecom) for the same user base without violating data protection regulations.

In fraud detection, FL helps institutions detect complex cross-channel anomalies by leveraging distributed behavioral signals without aggregating raw transaction data. This has significant value in online payments, digital wallets, and shared payment gateways.

6.3 Internet of Things (IoT) and Smart Devices

IoT ecosystems, comprising millions of distributed sensors, smart meters, and connected devices, are naturally aligned with the federated learning paradigm. Due to limited bandwidth and local data privacy requirements, transmitting all data to a central cloud is often impractical.

In smart homes and cities, FL has been applied to enable adaptive energy management, context-aware automation, and anomaly detection. For example, smart thermostats can collaboratively learn energy-efficient temperature control policies based on user preferences while keeping local usage data private.

Edge-based FL is also increasingly used in connected vehicles. Autonomous driving systems leverage FL to share learned representations of traffic patterns, road conditions, or pedestrian behavior across fleets, improving performance without requiring raw video uploads.

6.4 Personalized Learning and Educational Platforms

The COVID-19 pandemic accelerated the adoption of digital education tools, many of which rely on AI-driven personalization. FL has emerged as a tool to train recommendation models, adaptive testing systems, and student performance predictors across schools or student devices, without transmitting sensitive learning histories.

For example, educational platforms may use FL to train engagement models that adapt learning content to student attention span or topic mastery, while complying with data protection laws that restrict student data collection.

Additionally, FL supports equitable model training across underrepresented regions by incorporating local data from rural or low-resource schools, enabling global AI systems to better generalize across diverse learning contexts.

7. Open Research Challenges in Federated Learning

Despite significant progress in algorithms and systems, federated learning (FL) remains a rapidly evolving area filled with complex research challenges. These challenges stem not only from technical limitations but also from the demands of fairness, robustness, usability, and real-world applicability. In this section, we summarize several open problems that continue to motivate exploration in FL research.

7.1 Personalization vs. Generalization

A central tension in FL is the trade-off between learning a global model that generalizes across clients and adapting to each client's unique local distribution. As highlighted earlier, approaches such as meta-learning and local fine-tuning improve personalization but may sacrifice collaborative benefits. Research is ongoing to develop frameworks that can dynamically balance global and local objectives, possibly through multi-task learning or adaptive aggregation techniques.

Furthermore, current personalization solutions often assume that client data remains stationary over time, which rarely holds in practice. As such, lifelong federated learning—which accounts for evolving client data distributions—remains a challenging frontier.

7.2 Communication and Energy Efficiency

Although many communication-reduction techniques have been proposed, bandwidth and energy efficiency remain significant bottlenecks in real-world deployments, especially on edge and mobile devices. This problem is aggravated in dense deployments such as IoT environments or rural regions with unstable connectivity.

New directions such as over-the-air aggregation, wireless FL scheduling, and event-driven update triggering are being explored to further reduce energy consumption and latency. Integrating FL with 5G/6G network architectures is also a promising path, but the co-design of wireless communication protocols and FL algorithms is still in its infancy.

7.3 Privacy Beyond the Client

Most FL privacy protections focus on individual clients. However, in practice, inter-client privacy and model-level privacy are equally important. For example, a malicious participant might attempt to extract information about another client's contribution via model update differentials, even without accessing raw data.

Emerging research explores federated differential privacy with group constraints, gradient masking, and proof-of-contribution verification to defend against such threats. At the same time, ensuring auditable and verifiable training processes becomes critical in high-stakes environments such as finance and healthcare.

7.4 Benchmarking, Evaluation, and Reproducibility

The FL community still lacks standardized benchmarks that reflect the complexities of real-world settings. Existing benchmarks often simplify assumptions around data distributions, client availability, and participation patterns.

Efforts like LEAF, FedML, and FLSim have begun to offer diverse and realistic benchmarks, but they are far from comprehensive. Evaluation metrics beyond accuracy—such as fairness, robustness, energy use, and personalization accuracy—need to be formalized and adopted more widely.

In addition, reproducibility in FL experiments remains difficult due to heterogeneity in hardware, network latency, and simulator assumptions. Federated learning research would benefit from shared testbeds and open-source deployments that enable controlled experimentation at scale.

7.5 Regulatory and Ethical Constraints

Finally, federated learning raises new questions around regulatory compliance and ethical governance. While FL is often framed as a solution to privacy regulation, it must itself adhere to transparency, accountability, and auditability standards.

This includes ensuring that model updates do not embed biases, consent mechanisms are honored, and client control over participation is respected. These ethical dimensions are particularly salient in applications involving vulnerable populations, such as healthcare or education.

In summary, federated learning remains a fertile field for exploration. Its success in the real world will depend not only on algorithmic breakthroughs but also on solving systemic, regulatory, and ethical challenges. Cross-disciplinary collaboration among AI researchers, system architects, legal scholars, and practitioners will be essential to fully unlock the potential of FL.

8. Conclusion

Federated learning has emerged as a powerful paradigm that reconciles the need for collaborative model training with growing concerns around data privacy, system heterogeneity, and regulatory compliance. By enabling decentralized learning across devices and institutions, FL offers a scalable and privacy-conscious alternative to centralized AI pipelines, making it particularly suitable for industries such as healthcare, finance, IoT, and education.This paper has reviewed the foundational concepts of FL, including its system architecture, aggregation mechanisms, optimization strategies, and privacy-preserving techniques. We have also explored practical implementations and highlighted a range of successful applications. Throughout, we emphasized the interplay between algorithmic design and real-world deployment, showing how practical FL systems must adapt to data non-IIDness, network unreliability, and diverse resource constraints.At the same time, we have identified persistent research challenges—such as balancing personalization and generalization, enhancing communication efficiency, and developing comprehensive evaluation benchmarks—that continue to shape the evolution of the field. Addressing these challenges requires not only algorithmic innovation but also careful attention to engineering, ethics, and policy.As federated learning continues to mature, we anticipate its integration into a broader ecosystem of distributed intelligence, including edge computing, privacy-preserving machine learning, and responsible AI governance. By fostering collaboration across technical, legal, and social domains, federated learning holds the potential to unlock collective intelligence while respecting individual data autonomy—an increasingly vital objective in the age of ubiquitous AI.

References

[1] B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proc. AISTATS, 2017, pp. 1273–1282.

[2] T. Li, A. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50–60, May 2020.

[3] K. Bonawitz et al., "Towards Federated Learning at Scale: System Design," in Proc. MLSys, 2019, pp. 374–388.

[4] S. Wang et al., "Federated Learning with Matched Averaging," in Proc. ICLR, 2020.

[5] J. Konecny et al., "Federated Learning: Strategies for Improving Communication Efficiency," arXiv preprint arXiv:1610.05492, 2016.

[6] Y. Lin et al., "Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training," in Proc. ICLR, 2018.

[7] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," Foundations and Trends in Theoretical Computer Science, vol. 9, no. 3–4, pp. 211–407, 2014.
[8] K. Bonawitz et al., "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in Proc. ACM CCS, 2017, pp. 1175–1191.
[9] R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," in Proc. ACM CCS, 2015, pp. 1310–1321.