iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: III Month of publication: March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67878

Privacy-Preserving Federated Learning in TinyML

Lalit Sahu, Adarsh Patel, Sajjad Ahmed, Rizwan Ur Rehman
Department of Computing Science Engineering and Artificial Intelligence VIT Bhopal University, Sehore, India

Abstract: TinyML is a subdomain of machine learning that enables the implementation of real-time ML on resource constrained edge devices, such as microcontrollers, thereby reducing the need for cloud services and ensuring data privacy. FL completes this approach by decentralizing model training: it keeps the data local but shares model updates. However, FL comes with the risk of inference attacks, gradient leakage, and model poisoning, which is increased in the case of the low com- puting and energy capabilities of the TinyML device. This paper discusses the state-of-the-art privacy-preserving methods, such as Differential Privacy, Homomorphic Encryption, and Secure Multi-Party Computation, and provides an evaluation of their adaptability to TinyML. Optimized methods, including adaptive differential privacy and compressed secure aggregation, are proposed to balance strong privacy preservation with efficiency. The computational overhead, energy consumption, and real-time performance issues are discussed in future directions, such as lightweight cryptographic methods and hybrid techniques for privacy. This research will enable secure and efficient federated learning on TinyML devices, unlocking potential applications in privacy-critical domains such as healthcare, IoT, and smart cities.

Index Terms: TinyML, Federated Learning, Privacy Preservation, Resource-Constrained Edge Devices, Inference Attacks.

I. INTRODUCTION

In today's world, edge computing and IoT devices have increased, and a need is to run the machine learning models on tiny devices, like microcontrollers (MCUs), single-board computers (SBCs), etc. TinyML is a subfield of machine learning focused upon optimizations of models for these Tiny devices with the constraints of limited memory, energy, and computation. This allows such devices to perform real-time ML tasks instead of all the data going to the cloud, thus eliminating issues like latency and privacy over data [1]. Federated learning (FL) is a decentralized way of model training. They update shared models locally using their own data instead of sending raw data to a central server. Protects privacy by holding our confidential data on individual devices [2]. Federated learning is highly valuable in scenarios that involve a great deal of concern about data privacy, such as healthcare or finance.

However, Federated Learning shares updates to the common model with privacy risks such as inversion attacks, which reconstruct confidential data using updates to the shared model [3]. Compared to other contexts, this will be much more challenging to achieve privacy-preserving in Federated Learn- ing against these devices as TinyML devices are resource-constrained.

This paper discusses the state of the art privacy-preserving techniques used in federated learning, and we determine their applicability to TinyML. Technical challenges of maintaining privacy are discussed further, including new solutions aimed at optimizing privacy protection without any penalties on the system.

II. LITERATURE REVIEW

Reference	Focus	Key Contribution	Relation to TinyML and FL
Bonawitz et al. (2019) [1]	FL system design	Overview of architecture and scalability in FL	Addresses system level challenges relevant to FL in TinyML environments
Warden (2019) [4]	TinyML deployment	Introduction to deploying ML on microcontrollers	Core technology enabling FL on edge devices, forms foundation of the research
Kairouz et al. (2021) [8]	Advances in federated learning	Provides an extensive review of the open problems and advances in FL.	Addresses the need for privacy- preserving techniques in FL, which is highly relevant for TinyML deployments
Luong et al. (2021) [23]	FL applications in smart cities	Reviews the application of FL in smart cities, highlighting privacy and scalability challenges.	TinyML can be employed in smart city applications where privacy- preserving FL is essential for handling sensitive data.
Hu et al. (2022) [24]	Real-time FL for edge computing	Examines the challenges and design of real-time federated learning systems for edge computing.	Real-time performance is a challenge for privacy-preserving techniques in TinyML, where latency needs to be minimized.
Xu et al. (2021) [26]	Hybrid secure aggre- gation in FL for IoT	Proposes a hybrid approach to secure aggregation in FL for IoT networks.	Hybrid methods combining differential privacy and secure aggregation can offer an efficient balance of privacy and resource usage for TinyML.

III. TINYML AND FEDERATED LEARNING: A SYNERGISTIC APPROACH

A. *TinyML*

TinyML is emerging area of technology that aims at em- powering machine learning models to run on ultra-low-power devices like microcontrollers. Usually, these devices come with very limited computational powers, memory, and energy. They are used in a huge number in IoT systems, wearable tech- nologies, and sensor networks. The primary goal of TinyML is to transfer intelligence to the edge of a network, making devices work based on real-time inferences or decision-making without depending so much on cloud-based services [1,4]. This reduces constant data transmission, and thus it is good for both latency and data privacy. Still, there is no easy way of deploying machine learning models on TinyML devices. This requires substantial optimisation in such a way that these resource-constrained devices can undertake ML tasks without totally degrading the compute power, and power consumption should be kept exceedingly low for the devices in order not to waste battery life, thus allowing meeting operational requirements [5].

B. *Federated Learning*

Federated Learning: A Decentralized Approach toward Ma- chine Learning Federated Learning is an approach to machine learning that enables a multitude of devices-popularly known as clients-to contribute towards training a global model without transferring raw data to a central server. Instead of sharing data, each client trains its local model on data locally on the client device and sends model updates, specifically gradients, to a central aggregator. The aggregator aggregates such updates by averaging to produce a global model that can be sent back to the clients for additional training [2,6]. FL aims to help out with privacy since it is designed to prevent data leakage and unauthorized access while localizing the sensitive information on individual devices. FL is not without privacy risks either; threats like inference attacks and gradient leakage are still pending in which malicious parties can attempt to infer the secret information from the updates shared over the model [7]. TinyML coupled with FL presents an interesting synergy. It enables TinyML to be used for bringing in edge-based intelligence, but FL provides a privacy-aware technique for cooperative model training. However, as with the integration comes a new set of challenges. The resource constraints of TinyML devices combined with privacy vulnerabilities inherent in the FL framework makes it challenging to balance efficiency in processing with robust privacy preservation [8]. This especially underlines the need for further optimizing privacy-preserving methods that could successfully function within the constraints of the TinyML environment.

IV. PRIVACY RISKS IN FEDERATED LEARNING FOR TINYML

While federated learning maintains the requirement for centralized data aggregation due to the potential of sharing updates between models, it is still prone to several privacy risks. These risks include:

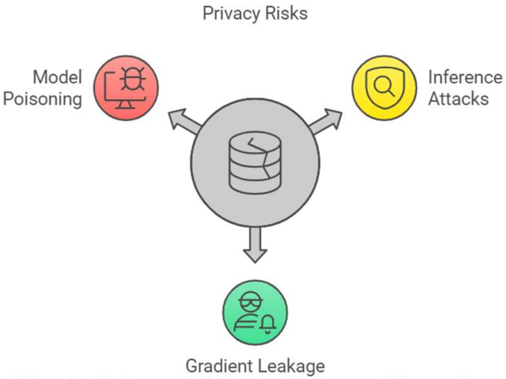

Fig. 1. Privacy Risks in Federated Learning

A. *Inference Attacks*

An attacker will try to infer sensitive information based on the gradients of the model exchanged in the course of training. Such an attack lets the adversary obtain insights into underlying data without requiring direct access to raw information [9].

B. *Gradient Leakage*

Sensitive information leaks in through the gradients shared between devices and central servers. Attackers can use leakage for reconstructing input data used for model updates, which is a considerable privacy risk, mainly for applications with sensitive input data like health care or finance [10].

C. Model Poisoning

Adversaries could manipulate the behavior of the global model or downgrade its accuracy to introduce malicious updates in the FL process. Such an attack type could deteriorate the integrity of the federated learning model and, if applied in real-world applications, may result in harmful circumstances [11].

These privacy risks become even more pronounced in the context of TinyML because of the limited computational and memory resources available on these devices. Advanced forms of cryptography, like differential privacy and secure multi-party computation, are often far too resource-intensive to easily be implemented on these constrained devices. As such, TinyML systems have faced problems in securing federated learning processes without sacrificing the efficiency needed to make real-time operations feasible [12].

This shall require developing privacy-preserving techniques based on the limitation of TinyML. These methods must strike a balance between robust privacy security and the necessity of resource efficiency in TinyML devices, to ensure that edge- based learning stays both secure and effective.

V. PRIVACY-PRESERVING TECHNIQUES IN FEDERATED LEARNING FOR TINYML

To address the threats arising from federated learning, different privacy-preserving techniques have been developed. Even so, the implementation of those techniques in TinyML environments should be well laid against the resource con- straints of edge devices.

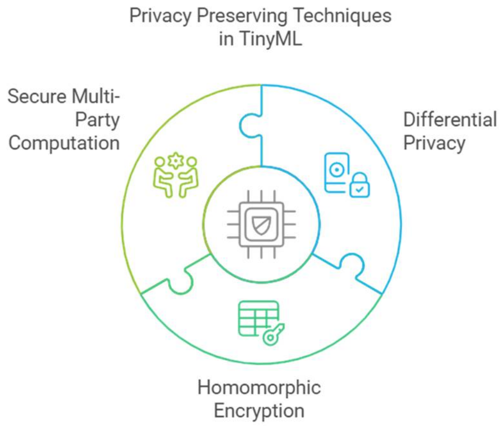
Privacy Preserving Techniques in TinyML
Secure Multi-Party Computation
Differential Privacy
Homomorphic Encryption

Fig. 2. Privacy Preserving Techniques in TinyML

A. Differential Privacy (DP)

DP is also an excellent mechanism to preserve data privacy by injecting statistical noise into updates towards the model to obscure the contribution of any given point in the data. In the context of TinyML, it asserts that though gradients leak during training, they do not leak specific information about the input data [13]. However, such noise will sometimes have the cost of decreasing the overall accuracy of the global model. Such balance is crucial in TinyML since the devices to be targeted are resource-constrained; too much noise degrades model performance, and too little provides inadequate privacy guarantees [14,15].

Optimization: Adaptive Differential Privacy

We can apply Adaptive Differential Privacy to achieve the preservation of performance. The secret behind it is that rather than adding uniform noise, we use an adaptive noise mecha- nism whereby the noise intensity depends on the sensitivity of the model. So while those gradients that are extremely sensitive receive more noise, a stable gradient will receive less noise.

$$\text{Noise Added} = \Delta f \times \frac{S}{} \qquad (1)$$

Where,

- Δf: Gradient (sensitivity) of the function,
- ϵ: Privacy parameter,
- S: Stability score, defined as the variance of the gradient across rounds.

By making this dynamic, TinyML devices can save re- sources by whitening noise only where it impacts the gradient while protecting privacy.

B. Homomorphic Encryption (HE)

Homomorphic Encryption (HE) is the encryption of compu- tations, allowing computation to happen directly on encrypted data. Devices could encrypt their local model updates before sending them to the central aggregator. The central aggregator could then compute the global update without decrypting the secret updates contributed by the local machines, thereby offering confidentiality within training [16]. However, HE is computationally expensive and is thus impractical in TinyML environments, where devices have drastically limited process- ing abilities and memory. Ensuring security and computational feasibility remains a significant challenge for practical appli- cations in TinyML environments [17].

Optimization: Pruning before Homomorphic Encryption

Since HE is computationally expensive, model size can be reduced by pruning unnecessary weights from the neural net- work before applying HE. Pruning reduces the computational load on TinyML devices while still enabling encryption.

The method is:

$$f_{\text{pruned}}(x) = \sum_{i=1}^{k} (w_i \cdot x_i) \quad \text{where} \quad k < n \qquad (2)$$

Here, w_i are the pruned weights that reduce the dimen- sionality from n to k. Such a reduction in the number of weights decreases the number of encrypted operations and thus optimizes the use of resources.

C. Secure Multi-Party Computation (SMPC)

In Secure Multi-Party Computation (SMPC), a set of parties can jointly compute some function while keeping their private inputs confidential. As such, SMPC can be used in federated learning to securely aggregate model updates without revealing individual contributions [18]. While SMPC protocols support privacy-related benefits, they involve several intercommuni- cation rounds among devices and execute computationally expensive operations, making them unsuitable for TinyML de- vices that are constrained by severe power and computational limits. This presents an obstacle to the efficient execution of SMPC in TinyML-based federated learning systems [19].

Optimization: Compressed Secure Aggregation (CSA)

A solution to this challenge is to use compression techniques instead of sending full model updates via SMPC. This reduces the communication cost involved in aggregating compressed updates while maintaining security, a method known as Com- pressed Secure Aggregation (CSA).

The approach is:

We assume that m_i is the update from device i, and compress(m_i) is its compressed version. The global update is:

$$\text{Global update} = \sum_{i=1}^{N} \text{compress}(m_i) \qquad (3)$$

Compression can be achieved using random projections or sketching techniques, which reduce the dimensionality of updates.

VI. ANALYSIS OF FEDERATED LEARNING PERFORMANCE ON TINYML DEVICES

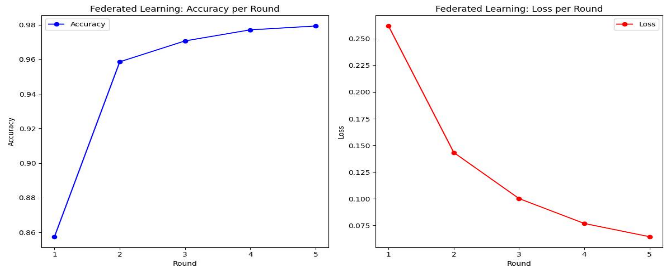

Fig. 3. Analysis of Federated Learning

This paper presents an exploration of the privacy-preserving techniques applied to federated learning in TinyML, with special attention to the key challenges and future opportunities that open up to deploy secure and efficient methods within such environments. Figure 3 illustrates the federative learn- ing results over several iterations, thereby displaying model accuracy along with loss trends.

A. Federated Learning and Privacy Preservation in Tiny ML

The primary reasoning behind applying FL to TinyML environments is the preservation of privacy without compromising performance. In the case of FL, data will be kept on the local device (in this case, the TinyML device) and model updates collected and shared with a central server. This would prevent the danger of privacy circumvention through data transfer, something that may be crucial in constrained resources and privacy applications, including healthcare, IoT devices, and smart homes.

The results of Figure 3 are as follows:

- Accuracy improves steadily from round to round from 0.86 to around 0.98, meaning that FL can indeed achieve strong performance even when there is only data spread across different TinyML devices.
- Loss diminishes steadily across rounds from 0.25 to below 0.05, showing that the model can learn to reduce prediction errors as the model gets updated across rounds.

These trends confirm the privacy-preserving potential of FL, since the model converges without sensitive data centraliza- tion, maintaining at the same time both high accuracy and low loss to achieve the dual objectives: privacy preservation and model efficacy. The plots of Figure 3 verify that, indeed, privacy-preserving federated learning can be implemented in TinyML without any performance degradation. Increases in accuracy and reductions in loss across rounds indicated that FL can be used in privacy-intensive and efficient applications.

VII. CHALLENGES AND OPPORTUNITIES

A. Computational Overhead

One challenge associated with embedding privacy- preserving techniques in TinyML arises from the high compute overhead associated with protocols such as Homo- morphic Encryption (HE) and Secure Multi-Party Computation (SMPC). Indeed, these protocols are too resource-intensive for the low-power low-memory devices used for the TinyML environment. In turn, such implementations on devices result in delays and system inefficiencies [20]. Therefore, future research needs to be on lightweight cryptographic methods that may provide adequate privacy protections but are computationally feasible for TinyML devices so that they do not strain too much on those limited resources [21].

B. Energy Consumption

In various energy-constrained applications like battery- powered sensors and wearable, it is offloading the devices with TinyML. Secure computations or encryption of the updated model can notably shorten the operation time of such devices due to the extra energy being consumed by them. Crypto- graphic operations, like those in HE and SMPC, are somewhat computationally heavy and therefore typically energy-hungry [22]. So, there is a tremendous need for energy-preserving privacy-preserving mechanisms that are really very efficient to preserve robust privacy protection without adversely affecting the battery life or use of energy. In these ways, it allows devices to function over extended periods with high efficiency [23].

C. Real-Time Performance

TinyML devices are applied extensively in many real-time applications, such as monitoring, controlling, and making decisions within the IoT environment. It has been shown that privacy-preserving technology should be preserved in systems that still provide real-time capabilities, since critical delays could degrade system functionality, mainly in sensitive applications such as healthcare, industrial automation, or even safety-critical systems [24]. Therefore, future solutions should find a delicate balance of trade-offs between ensuring privacy with latency. It aims at developing privacy-preserving meth- ods without any overhead delays and should allow TinyML systems to work seamlessly in real-time environments [25].

VIII. FUTURE DIRECTIONS

The future of the online, privacy-preserving federated learn- ing in TinyML will hence rely on developing lightweight cryp- tographic methods and optimizing existing privacy-preserving techniques to fit the resource-constrained environment. Some promising research and development directions include:

A. Lightweight Differential Privacy (DP)

Update the current differential privacy protocols to lighten up the computational constraint while maintaining a reason- able level of privacy for TinyML devices. This is focused on a delicate balance of protecting privacy against the related constraint of limited processing power in the devices [21].

B. Optimised Homomorphic Encryption (HE)

Energy-efficient implementation of Homomorphic Encryp- tion methods which are matched with the processing capabil- ities of TinyML devises. Optimized HE method development will ensure computations on encrypted data can be securely performed without any effect on the performance or energy consumption by the device [17].

C. Hybrid Privacy Techniques

Combining one or more privacy-preserving techniques together, such as a combination of differential privacy and secure aggregation may achieve an optimal solution. Hybrid approaches may amalgamate some merits from different techniques toward achieving higher accuracy with better privacy and computational efficiency for TinyML systems [26].

These solutions would evolve to maintain privacy in federated learning for TinyML and address challenges of edge devices.

IX. CONCLUSION

Privacy-preserving federated learning has high potential in safe and efficient machine learning on TinyML devices. Federated learning is decentralized and benefits from reduced some privacy risks but requires additional techniques like differential privacy, homomorphic encryption, and secure multi-party computation to further mitigate threats such as gradient leakage and inference attacks. Techniques must be carefully adapted to practical deployment in very constrained TinyML devices with limited computational power, memory, and energy. The main challenge in federated learning in TinyML environments is high computational overhead, high energy consumption, and real-time performance. Research into the optimization of privacy-preserving methods for TinyML will continue to open up avenues of applications in scenarios that exploit these technologies in IoT networks, healthcare systems, and far beyond. That will ultimately unlock a future where edge-based intelligence is private and efficient.

REFERENCES

[1] Bonawitz, K., Eichner, H., Grieskamp, W., et al. (2019). Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046.

[2] McMahan, B., Moore, E., Ramage, D., & Hampson, S. (2017). Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629.

[3] Hitaj, B., Ateniese, G., & Pe´rez-Cruz, F. (2017). Deep models un- der the GAN: Information leakage from collaborative deep learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 603-618.

[4] Warden, P. (2019). TinyML: Machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers. O'Reilly Media.

[5] Reddi, V. J., Cheng, C., & Han, S. (2020). TinyML: Enabling ultra-low- power machine learning at the edge. IEEE Computer, 53(10), 48-57.

[6] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learn- ing: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50-60.

[7] Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V. (2019). Exploit- ing unintended feature leakage in collaborative learning. Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), 691-706.

[8] Kairouz, P., McMahan, H. B., & Avent, B. (2021). Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1), 1-210.

[9] Zhao, H., & Meng, X. (2021). Privacy-preserving federated learning: A survey. ACM Computing Surveys (CSUR), 54(4), 1-36.

[10] Wang, Z., & Gong, S. (2021). Preserving privacy in federated learning using differential privacy and secure multi-party computation. Journal of Cryptographic Engineering, 11(2), 125-134.

[11] Fung, C., Yoon, C. J. M., & Beschastnikh, I. (2020). Mitigating sybil attacks in federated learning: A weighted approach. arXiv preprint arXiv:1808.04866.

[12] Ruan, M., Wu, S., & Li, X. (2022). Efficient privacy-preserving federated learning in blockchain-based IoT systems. IEEE Internet of Things Journal, 9(10), 7445-7456.

[13] Dwork, C. (2006). Differential privacy. International Colloquium on Automata, Languages, and Programming, 1-12.

[14] Abadi, M., Chu, A., Goodfellow, I., et al. (2016). Deep learning with differential privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 308-318.

[15] Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. Proceedings of the 2015 ACM SIGSAC Conference on Computer and Communications Security, 1310-1321.

[16] Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. Foundations of Secure Computation, 4(11), 169-180.

[17] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. Proceedings of the 41st Annual ACM Symposium on Theory of Com- puting, 169-178.

[18] Bonawitz, K., Ivanov, V., Kreuter, B., et al. (2017). Practical secure aggregation for privacy-preserving machine learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1175-1191.

[19] Mohassel, P., & Zhang, Y. (2017). SecureML: A system for scalable privacy-preserving machine learning. Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), 19-38.

[20] Zhao, H., Liu, Y., Gong, X., & Cheng, X. (2021). Lightweight blockchain consensus mechanisms for IoT-based smart applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(6), 3755-3766.

[21] Jiang, W., Sivaraman, A., Hoang, D. T., et al. (2021). Federated learning in the sky: Joint power allocation and scheduling with UAV swarms. IEEE Transactions on Communications, 69(11), 7468-7483.

[22] Qian, L., Zhang, F., Li, H., & Zhang, Y. (2021). Energy-efficient blockchain-enabled federated learning for edge computing. IEEE Trans- actions on Green Communications and Networking, 5(3), 1401-1410.

[23] Luong, N. C., Hoang, D. T., Gong, S., et al. (2021). Applications of federated learning in smart cities: Recent advances, challenges, and future directions. IEEE Communications Magazine, 59(4), 70-77.

[24] Hu, Z., Li, X., & Gao, Q. (2022). Real-time federated learning for edge computing: Design and challenges. IEEE Network, 36(1), 45-51.

[25] Konecˇny´, J., McMahan, B., Yu, H., et al. (2016). Federated learn- ing: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.

[26] Xu, J., Wang, Z., & Zhang, L. (2021). Hybrid secure aggregation in federated learning for IoT edge networks. IEEE Transactions on Network Science and Engineering, 8(4), 3247-3258.