Can PPML Keep Your Data Private? Evaluating Privacy Techniques Against Inference Attacks

Noam Tarshish
Ben-Gurion University
Beer-Sheva, Israel
noamtars@post.bgu.ac.il

Daniel Hodisan
Ben-Gurion University
Beer-Sheva, Israel
hodisan@post.bgu.ac.il

Lavi Ben-Shimol
Ben-Gurion University
Beer-Sheva, Israel
laviben@post.bgu.ac.il

Abstract

As Privacy-Preserving Machine Learning (PPML) techniques like Federated Learning (FL) and Differential Privacy (DP) gain widespread adoption, it is critical to evaluate their effectiveness against practical privacy threats. In this study, we assess the resilience of FL and DP using Random Forest classifiers under three inference attacks: Membership Inference Attack (MIA), Model Inversion Attack, and Attribute Inference Attack (AIA). We evaluate three configurations: (1) a centralized Random Forest (Naive), (2) a federated ensemble, and (3) a federated model with differential privacy.

To further strengthen runtime security, we incorporate Intel SGX—a hardware-based Trusted Execution Environment (TEE)—and analyze its ability to prevent memory extraction attacks targeting sensitive model parameters and input data during inference. Our SGX-protected setup shows that even against privileged adversaries with root access, hardware isolation effectively thwarts process memory analysis and direct extraction of floating-point model coefficients.

Our results show that while FL alone moderately reduces vulnerability to inference attacks, the addition of DP significantly lowers attack success, dropping MIA AUC from 0.665 to 0.559 and AIA accuracy from 0.604 to 0.498, while preserving model accuracy above 90%. Even though SGX is not integrated into the federated pipeline, it serves as a complementary protection layer, against powerful system-level threats. This empirical study highlights the privacy-utility trade-offs of PPML and answers the central question: **Can PPML keep your data private under attack?**

1 Introduction

As machine learning systems are increasingly applied to sensitive domains like healthcare and finance, concerns about data privacy and model leakage have grown. Privacy-Preserving Machine Learning (PPML) seeks to address these risks by enabling learning over distributed or sensitive data while minimizing information exposure.

In this work, we evaluate two widely adopted PPML techniques—*Federated Learning (FL)* and *Differential Privacy (DP)*—and analyze their effectiveness in defending against inference-based privacy attacks. While FL decentralizes training by keeping data local, DP adds calibrated noise to model outputs or gradients to limit information leakage. Both methods offer software-level protection but introduce trade-offs between privacy and utility that must be empirically assessed.

To this end, we train Random Forest classifiers under three configurations: a centralized baseline (Naive RF), a federated ensemble, and a federated model with differential privacy. We simulate three inference attacks—Membership Inference, Model Inversion, and Attribute Inference—to evaluate the vulnerability of each configuration.

Beyond inference-time threats, we also examine the security of machine learning inference under a stronger adversary model with full system access. Specifically, we evaluate the use of Intel Software Guard Extensions (SGX)—a hardware-based Trusted Execution Environment (TEE)—to defend against memory extraction attacks that target sensitive model parameters and patient data during inference. While SGX is not integrated into the federated pipeline, it serves as a complementary runtime protection layer that can isolate model execution from untrusted infrastructure.

Our study provides a comprehensive empirical comparison of software-based privacy defenses (FL, DP) against inference attacks, and evaluates the added benefit of hardware-based protection (SGX) against privileged adversaries. This enables a broader understanding of how different PPML strategies address both data and runtime security in sensitive machine learning applications.

2 Related Work

This section reviews previous research on privacy and security in the machine learning lifecycle, covering the following: inference-time threats, software-based defenses, hardware-based isolation, and domain-specific use cases. We organize related work into six subsections to contextualize our approach.

1. Vulnerabilities in centralized learning paradigms (Section 2.1). 2. Security challenges and mitigations in federated learning (Section 2.2). 3. The role and impact of differential privacy mechanisms (Section 2.3). 4. Security considerations for our demo model class: tree-based learners (Section 2.4). 5. Domain-specific PPML applications in healthcare with multiple confidential data providers (Section 2.5). 6. Hardware-based confidentiality guarantees with Trusted Execution Environments such as SGX (Section 2.6).

Each subsection presents key attack types, defense mechanisms, and limitations addressed by our work, including inference leakage and runtime memory compromise.

2.1 Vulnerabilities in Centralized Machine Learning

Centralized ML—where all data reside on a single server—exposes models to membership inference, model inversion, and attribute inference attacks. Membership Inference Attacks (MIA) enable an adversary to infer whether a given data record was used during training by training shadow models to mimic the target model's decision boundary and analyzing output confidences or prediction losses to distinguish between members and non-members. Shokri et al. (Shokri et al., 2017) demonstrated membership inference attacks achieving up to 85% accuracy on image benchmarks (e.g., CIFAR-10) and precision above 0.85 for over 20 classes on a Texas hospital-stay dataset of 67,330 records using 10 shadow models.

2.2 Security Challenges in Federated Learning

Federated Learning (FL) enables multiple clients to collaboratively train a shared model by keeping raw data on local devices. The most common aggregation method is Federated Averaging (FedAvg) (McMahan et al., 2017), which computes a weighted average of client model updates. While FL prevents centralized data pooling, sharing gradients or model parameters can leak private information. Melis et al. (Melis et al., 2019) evaluated FL with a convolutional neural network on CIFAR-10 and CIFAR-100 using ten clients, and showed gradient-based membership inference attacks achieving up to 70% accuracy. Nasr et al. (Nasr et al., 2019) performed white-box reconstruction attacks on FL settings with a CNN on MNIST and logistic regression on Purchase-100, recovering images with structural similarity index (SSIM) between 0.5 and 0.7. These studies demonstrate that, despite mitigating direct data exposure, FL remains vulnerable to gradient-leakage and reconstruction attacks without additional privacy safeguards.

2.3 Differential Privacy Mechanisms

Differential Privacy (DP) provides a formal guarantee that the output of a computation reveals limited information about any single record.

DP-SGD is the most common instantiation in machine learning. Each per-example gradient is clipped to a maximum norm C to bound its sensitivity. Then, noise drawn from $\mathcal{N}(0,\, \sigma^2 C^2)$ is added to the sum of clipped gradients. By accounting for privacy loss with the moments accountant, DP-SGD yields an (ε, δ)-DP guarantee (Abadi et al., 2016). On image benchmarks such as CIFAR-10, DP-SGD at $\varepsilon = 1$ reduces membership inference accuracy from 85% to about 55% while maintaining over 70% top-1 accuracy (Abadi et al., 2016).

Dwork et al. (Dwork et al., 2016) introduced the Laplace mechanism, which achieves pure ε-DP by adding noise sampled from $\mathrm{Laplace}(\Delta f/\varepsilon)$, where Δf is the query sensitivity. Although optimal for low-dimensional queries, applying Laplace noise to high-dimensional model gradients (e.g. in a 10^6-parameter deep network) can degrade accuracy by over 20% on standard image tasks (Truex et al., 2020).

In federated learning, Geyer et al. (Geyer et al., 2017) integrated DP-SGD into FedAvg on two benchmarks: an LSTM next-character task using the Shakespeare dataset with 1 000 clients, and a CNN on CIFAR-10 with 600 clients. At $(\varepsilon = 8, \delta = 10^{-5})$, they observed test accuracy drops of 9.8% (from 77.2% to 67.4%) on CIFAR-10 and 11.3% (from 52.0% to 40.7%) on Shakespeare, demonstrating a practical utility–privacy trade-off.

Local DP (LDP) approaches apply Laplace noise at each client before any aggregation. Truex et

al. (Truex et al., 2020) evaluated LDP on logistic regression over the UCI Adult dataset (50 000 records, 14 features), finding accuracy declines from 84.3% to 69.2% at $\varepsilon = 0.5$. This shows pure-DP guarantees come at a higher cost in utility for tabular data.

Despite these advances, a systematic comparison of Gaussian versus Laplace mechanisms in federated, tabular healthcare scenarios—with mixed numerical and categorical features—remains an open research question.

2.4 Vulnerabilities of Tree-Based Models

Random Forests (RFs) are widely used for tabular classification due to their high accuracy and interpretability. Empirical studies report that RFs are vulnerable to membership inference attacks, reaching 60%–70% attack accuracy on benchmark tabular datasets such as UCI Adult (Deng and Liu, 2019).

Model inversion attacks attempt to reconstruct input features from model outputs or confidence scores (Fredrikson et al., 2015). Surrogate-based inversion techniques recover roughly 50% of decision paths in RFs by training substitute models on API outputs (Tram'er et al., 2020). Attribute inference attacks predict sensitive attributes given non-sensitive feature values and model access, achieving up to 65% recovery accuracy on census-style data (Zhang et al., 2020). These results show that tree ensembles expose a multifaceted attack surface, necessitating a holistic security evaluation alongside neural models.

2.5 Domain-Specific PPML in Healthcare

Real-world healthcare settings often involve federated collaborations among hospitals under regulations like HIPAA and GDPR. Yang et al. (Yang et al., 2021) ran FedAvg across five medical centers on aggregated EHR data for disease prediction, achieving within 5% of centralized model accuracy. Li et al. (Li et al., 2020) applied DP-SGD to logistic regression on EHR records, demonstrating that $(\varepsilon = 1, \delta = 10^{-5})$-DP can be enforced with less than 10% drop in AUC. However, these works typically isolate either FL or DP, and rarely evaluate combined defenses against multiple inference threats in a multi-institution context.

2.6 Hardware-Based Privacy with SGX

While most PPML strategies operate at the software level, Trusted Execution Environments (TEEs) such as Intel SGX offer hardware-enforced isolation to protect model execution and data during runtime. Costan and Devadas (Costan and Devadas, 2016) provide a foundational overview of SGX, detailing its memory encryption and enclave mechanisms. However, several studies (Van Bulck et al., 2018; Halderman et al., 2009) have shown that unprotected memory is vulnerable to physical and speculative execution attacks, motivating the need for enclave-based computation. Chen et al.(Chen et al., 2017) and Schwarz et al.(Schwarz et al., 2019) explored side-channel attacks even within SGX, while recent projects like Graphene (Tsai et al., 2017) and Gramine (Gramine Project, 2024) provide practical frameworks for deploying unmodified applications inside enclaves. Our work extends this line of research by empirically evaluating SGX's ability to protect Random Forest inference from memory extraction attacks, complementing FL and DP with strong runtime confidentiality.

2.7 Summary and Research Gap

While existing work has separately evaluated membership inference, model inversion, and attribute inference attacks on centralized models, and has applied Differential Privacy or Federated Learning in isolation, no prior study provides a unified, ablation-style comparison of all three defenses (Naive RF, FL, FL+DP) *and* all three inference attacks on healthcare-style tabular data.

Furthermore, while hardware-based isolation mechanisms such as Intel SGX have been proposed to protect model inference from memory-level threats, they are rarely evaluated in conjunction with PPML defenses or in practical machine learning pipelines.

By systematically quantifying utility–privacy trade-offs across classification accuracy, AUC, MSE, and inference success rates, and by complementing them with a hardware-backed SGX experiment under privileged adversaries, our work fills the gap for a comprehensive privacy evaluation of tree-based learners in sensitive federated settings.

3 Methodology

3.1 Overview

The methodology consists of three main phases: (1) preprocessing and sharding of the dataset to simulate decentralized client environments, (2) training of Random Forest models under three privacy configurations (Naïve, FL, FL+DP), and (3) simulation

of three categories of inference attacks.

In addition, we introduce a hardware-based evaluation using Intel SGX. This step aims to assess the effectiveness of Trusted Execution Environments (TEEs) in preventing memory-level attacks during inference time. While the SGX component is not part of the federated pipeline, it serves as a complementary protection mechanism against stronger adversaries with full system access. The full flow is illustrated in Figure 1.

3.2 Dataset and Problem Context

We use the *Healthcare Dataset* from Kaggle (Patil, 2020), a synthetic tabular dataset designed for multi-class classification analysis in healthcare settings. The dataset simulates patient records with features such as *age*, *gender*, *blood type*, *medical conditions*, *admission details*, and *billing information*.

It includes both numerical and categorical attributes, making it ideal for preprocessing steps such as normalization and categorical encoding. The data structure reflects sensitive healthcare domains, where inference threats—such as attribute inference—could compromise private medical or demographic details.

3.3 Data Preprocessing and Sharding

The original dataset is first subjected to a standard preprocessing pipeline. All numerical attributes are scaled using min-max normalization to bring values into a $[0, 1]$ range, which helps prevent certain features from dominating the learning process due to magnitude differences. Meanwhile, categorical attributes are transformed using one-hot encoding, which converts each category into a binary vector. This format is essential for Random Forest models, which do not natively handle categorical data.

Following preprocessing, the dataset is randomly partitioned into five equal, non-overlapping subsets. Each subset is assigned to a simulated client to reflect a federated learning environment where raw data remains locally stored and inaccessible to the central server or other clients. This form of data sharding emulates real-world privacy-sensitive settings, such as hospitals, banks, or regional agencies, where data sharing is restricted by design or by regulation.

3.4 Model Training Strategy

We define three training paradigms to explore the privacy-utility spectrum:

- **Naïve (Centralized) Model:** In this baseline setup, we train a single Random Forest model using the complete dataset. Since no privacy mechanism is applied, this serves as a control to measure attack susceptibility and baseline performance without protection.

- **Federated Learning (FL):** In this decentralized approach, each of the five clients independently trains a local Random Forest model using only its shard. Once trained, these models are aggregated into a global ensemble using majority voting. Since only predictions are shared—not raw data or model parameters—this setup provides partial privacy through decentralization alone.

- **Federated Learning with Differential Privacy (FL + DP):** To enhance privacy, this configuration adds Gaussian noise to each client's prediction vector before aggregation. This simulates output-level differential privacy without altering the internal training procedure of the Random Forests. The amount of noise (controlled by a tunable parameter ε) influences the balance between privacy protection and prediction accuracy.

This tiered training strategy enables us to assess how FL and DP individually and jointly impact model utility and vulnerability.

3.5 Attack Simulation

To quantify the effectiveness of each training configuration in defending against privacy threats, we simulate three well-established inference attacks:

- **Membership Inference Attack (MIA):** This attack aims to determine whether a given data sample was part of the training set. We implement several variations of MIA using different scoring signals—namely, model confidence, prediction entropy, and cross-entropy loss. These signals are commonly used in adversarial settings to detect overfitting or confidence gaps between seen and unseen data.

- **Model Inversion Attack:** This technique attempts to reconstruct sensitive input features by observing a model's outputs. We simulate this by training a shadow model that learns to invert the mapping from predicted labels or probabilities back to original input values.

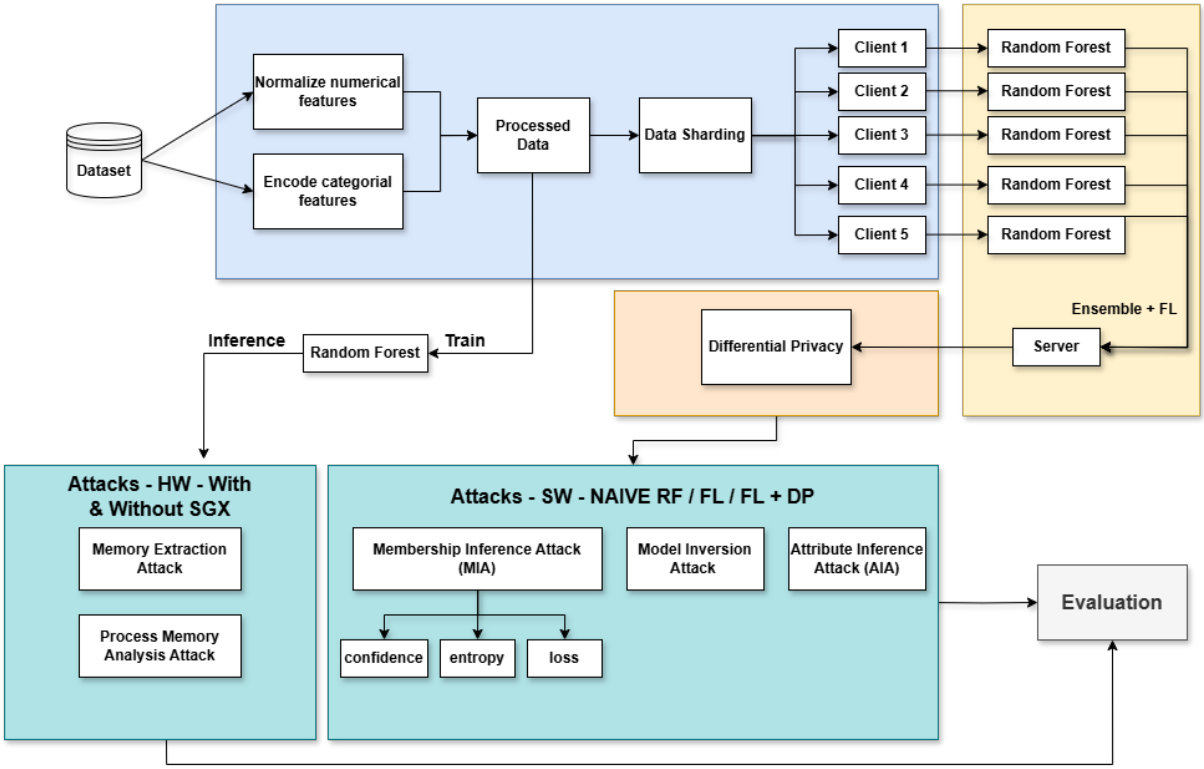

Figure 1: End-to-end PPML pipeline illustrating the full evaluation flow: from preprocessing and federated data sharding, through model training under three privacy configurations (Naive, FL, FL+DP), to attack simulation and evaluation. A parallel SGX track demonstrates hardware-based memory protection during inference, independent of the federated setup.

The risk here is that the model might "memorize" representative samples and leak attribute distributions through its behavior.

- **Attribute Inference Attack (AIA):** AIA targets sensitive attributes that were not explicitly included in the model's output. Given access to model predictions and non-sensitive features, the attacker attempts to infer a hidden attribute—such as medical condition, gender, or blood type. We remove one attribute from the dataset and train an adversarial classifier to predict it using the rest of the input and model responses.

Each of these attacks is executed against all three model configurations (Naïve, FL, FL+DP). Their effectiveness allows us to understand how vulnerable each setup is to different privacy threats. The evaluation and detailed results of these attacks are presented in Section 4.

3.6 SGX-Based Protection Against Memory Attacks

To evaluate defenses against privileged adversaries with full system access, we implemented a hardware-based protection setup using Intel Software Guard Extensions (SGX). SGX enables the execution of machine learning inference within a Trusted Execution Environment (TEE), where both code and data remain isolated from the host operating system, hypervisor, or any external process.

Unlike the FL and DP components, which focus on software-level defenses against inference-time attacks, the SGX component addresses a distinct class of threats: memory extraction and process introspection at runtime. These attacks are especially relevant in cloud or multi-tenant environments where adversaries may have root-level access.

3.6.1 Threat Model

We assume a powerful adversary capable of:

- Reading arbitrary regions of process memory via interfaces such as `/proc/{pid}/mem`
- Using introspection tools to extract floating-point model coefficients and patient inputs
- Inspecting heap and writable memory segments in real-time

3.6.2 SGX Workflow

Figure 1 (bottom left) illustrates the SGX evaluation path in parallel to the FL-DP pipeline. The SGX methodology includes the following steps:

1. **Model Wrapping:** A trained Random Forest model is selected for secure inference.
2. **Gramine Integration:** The Python-based inference script is ported into a secure SGX enclave using the Gramine LibOS framework.
3. **Manifest Configuration:** Security policies, file access rules, and enclave memory allocation are defined using a Gramine manifest file.
4. **Attestation and Launch:** The enclave is launched and remote attestation confirms its integrity and genuine SGX environment.
5. **Attack Simulation:** We attempt to extract model coefficients and patient inputs using direct memory read and process memory analysis, both with and without SGX.

3.6.3 Implementation Details

- **Libraries and Tools:** We used `psutil`, `struct`, and the `/proc` interface for memory inspection, and `Gramine v1.4+` to deploy the enclave.
- **Hardware Requirements:** Intel SGX-compatible CPU, Flexible Launch Control (FLC), and 8GB EPC memory.
- **Protected Files:** Both model parameters (`healthcare_model.pkl`) and test inputs were measured and protected by the enclave.

3.6.4 Verification Procedure

To validate SGX's protection, we compare:

- **Number of accessible memory regions** in SGX vs. non-SGX modes
- **Success rate of value extraction** (floating-point weights) across both settings
- **Attack surface reduction**, quantified by inaccessible regions and permission denials

This analysis enables a hardware-level validation of memory protection during ML inference and complements the software-based privacy guarantees of FL and DP.

4 Evaluation and Results

4.1 Model Utility Evaluation

We begin by evaluating the utility of each configuration using classification performance metrics. Table 1 summarizes the accuracy and AUC scores, while Figures 2, 3 visualize the model behavior across training settings.

Configuration	Accuracy	AUC
Naive RF	0.895	0.592
Federated	0.901	0.568
Federated + DP	0.901	0.489

Table 1: Utility comparison across privacy configurations.

Insights

From Table 1, we observe that:

- Federated training slightly improves accuracy compared to centralized training.
- However, AUC slightly drops in the federated setup, possibly due to ensemble averaging effects.
- Differential Privacy maintains accuracy but significantly affects model confidence and separability, as we'll show in the attack evaluation sections.

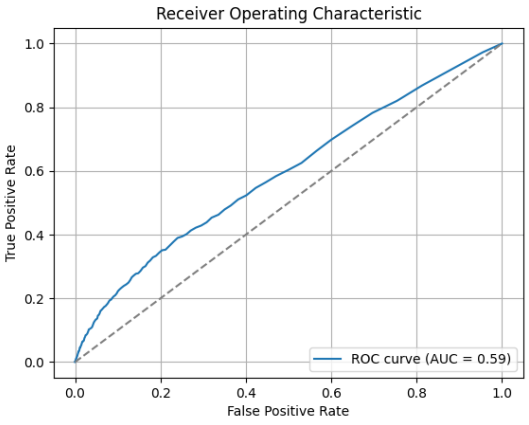

Figure 2: ROC Curve for Naive Random Forest (AUC = 0.592)

4.2 Membership Inference Attack (MIA)

We evaluate the vulnerability of each training configuration to membership inference attacks. This attack attempts to determine whether a given sample was part of the model's training set by analyzing output signals such as confidence, entropy, or per-sample loss.

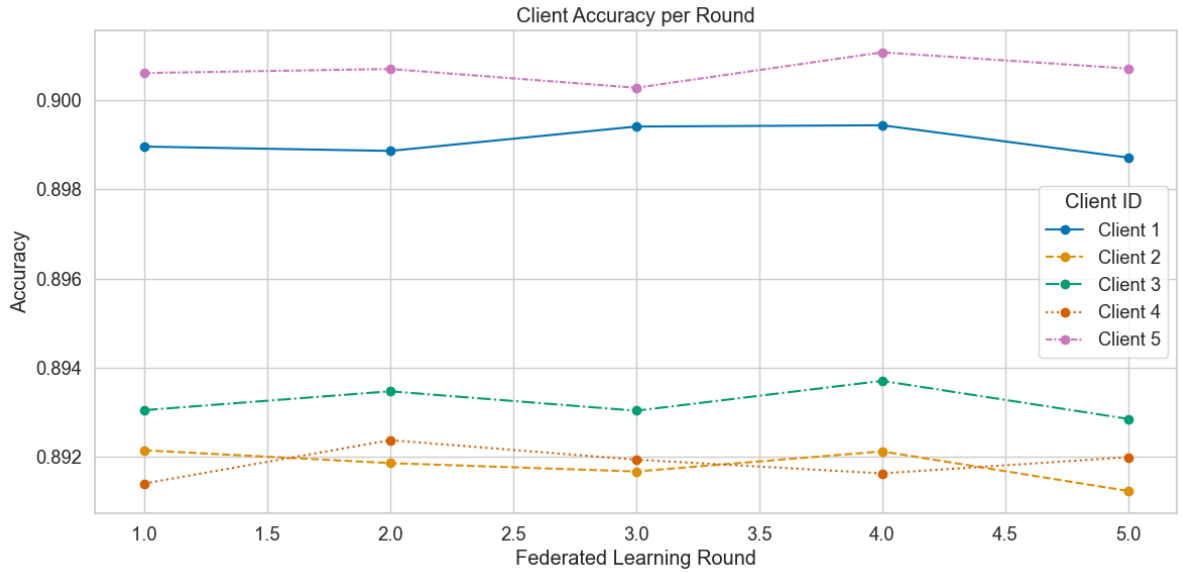

Figure 3: Client-wise accuracy during FL training (5 rounds)

Attack Setup

The attack model is a binary classifier trained to distinguish between member and non-member samples using outputs from the target model. For each configuration (Naive RF, FL, FL+DP), we compute ROC curves based on confidence, entropy, and loss. The AUC of these curves reflects how effectively the adversary can perform the attack.

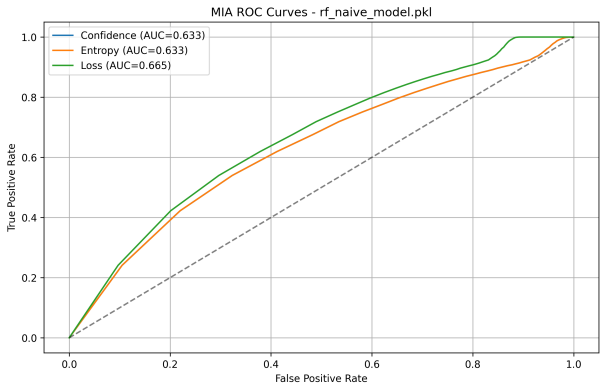

Figure 4: MIA ROC Curve – Naive RF (Best AUC = 0.665 using loss)

Membership Inference Attack (MIA) - Insights

- The Naive RF model is the most vulnerable to MIA, with an AUC of 0.665 using the loss signal, showcasing significant attack success compared to random chance (0.5), as shown in Figure 4.
- Federated training lowers the attack success rate slightly (AUC = 0.561), likely due to ensemble variance between clients, as shown in Figure 5.
- The addition of Differential Privacy further decreases the adversary's ability to infer membership. However, the AUC still remains above 0.5, indicating partial leakage, as shown in Figure 6.
- As shown in Figure 7, the loss metric consistently provides the highest leakage signal across all settings, while confidence and entropy provide weaker but still non-trivial signals.

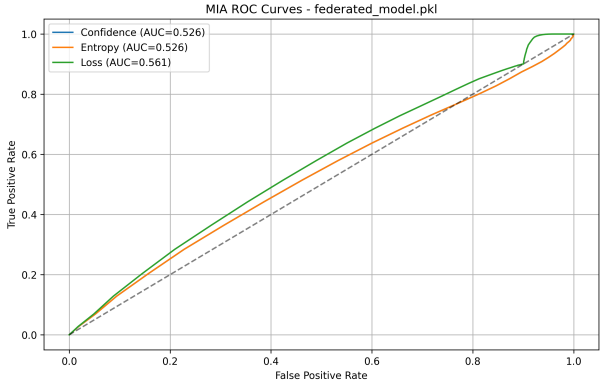

Figure 5: MIA ROC Curve – Federated (Best AUC = 0.561 using loss)

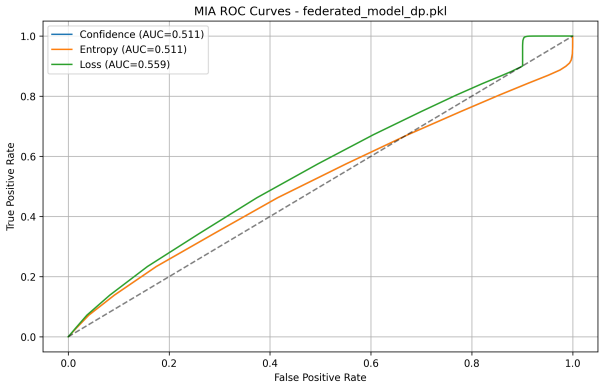

Figure 6: MIA ROC Curve – Federated + DP (Best AUC = 0.559 using loss)

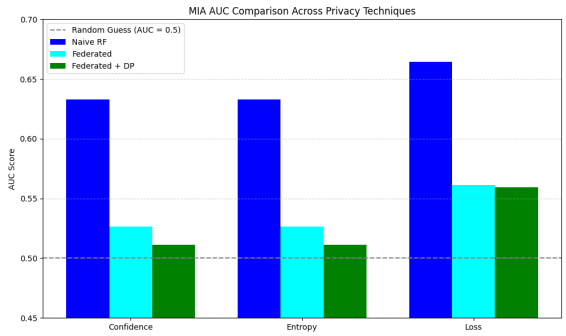

Figure 7: AUC Comparison of MIA across Training Configurations and Signal Types

4.3 Model Inversion Attack

The goal of model inversion is to reconstruct input features (or their distributions) by leveraging the model's output. An attacker may use access to soft predictions or confidence values to approximate the original feature vectors.

Attack Setup

we focus on reconstructing a binary-sensitive label using a regression-based inversion model. The success of the attack is evaluated using the Mean Squared Error (MSE) between the predicted and true feature values.

Model Inversion Attack - Insights

- The Naive RF model is highly vulnerable to inversion attacks, with an extremely low MSE (0.0001), meaning the predicted values almost perfectly match the true labels, as shown in Figure 8.
- Federated Learning significantly reduces the attack's effectiveness (MSE increases to 0.0992), likely due to client-level distribution noise and model variance, as shown in Figure 9.
- Interestingly, adding Differential Privacy to the federated setup does not further increase the MSE, as shown in Figure 10.
- This suggests that Federated Learning alone disrupts the gradient or prediction signal enough to reduce the attacker's reconstruction capabilities.

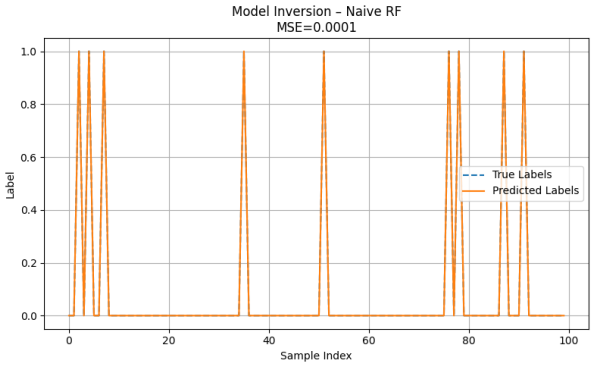

Figure 8: Model Inversion – Naive RF (MSE = 0.0001)

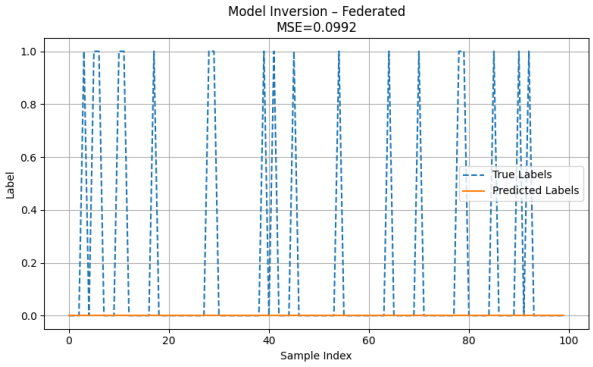

Figure 9: Model Inversion – Federated (MSE = 0.0992)

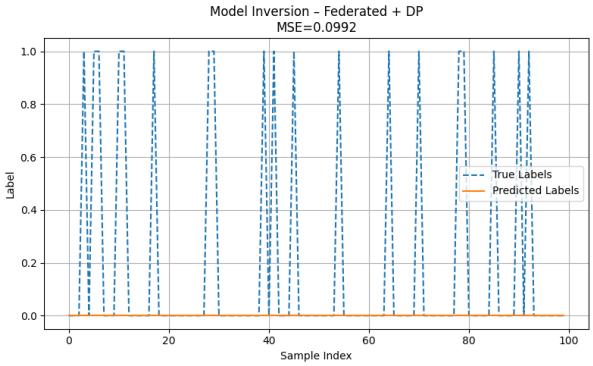

Figure 10: Model Inversion – Federated + DP (MSE = 0.0992)

4.4 Attribute Inference Attack (AIA)

The goal of Attribute Inference Attacks is to predict a sensitive attribute (e.g., blood type) that was not included as a direct target during model training. The attacker uses model outputs and known non-sensitive features to infer the hidden attribute.

Attack Setup

we train a separate attack model on model outputs and known features to predict a binary-sensitive feature. We report the attack's classification accuracy and analyze feature importance.

Attribute Inference Attack (AIA) - Insights

- Surprisingly, the federated setup shows higher attribute inference accuracy (0.604) than the Naive RF (0.546). This could be due to overfitting on local distributions, unintentionally exposing feature correlations, as shown in Figure 11 and Figure 12.

- When Differential Privacy is applied in the federated setup, the attack's accuracy drops to 0.498 — nearly indistinguishable from random guessing, as shown in Figure 13. This supports the effectiveness of DP in obscuring fine-grained correlations between features and sensitive attributes.

- Feature importance analysis (right side of each figure) shows that in Naive RF and FL, the model's own confidence is often among the top predictors — again reinforcing the role of prediction strength in privacy leakage.

- Under DP, feature importances flatten and shift to low-signal features, illustrating noise injection's success in neutralizing inference.

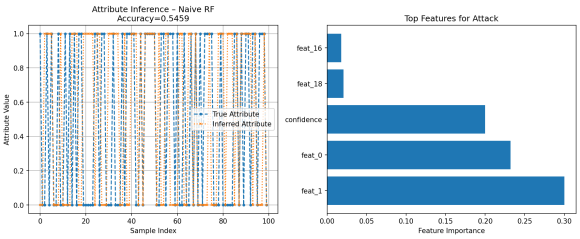

Figure 11: Attribute Inference – Naive RF (Accuracy = 0.546)

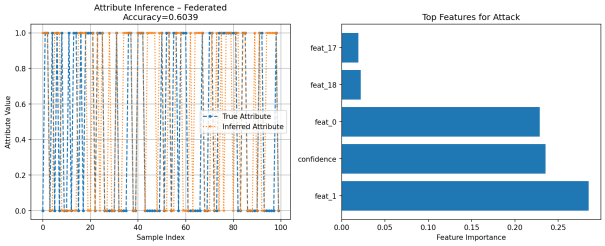

Figure 12: Attribute Inference – Federated (Accuracy = 0.604)

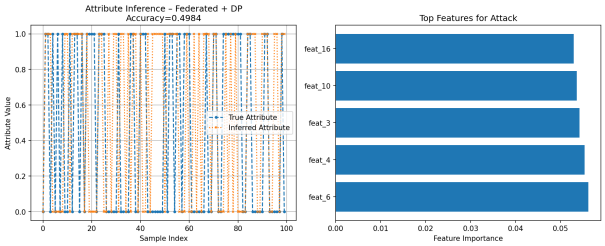

Figure 13: Attribute Inference – Federated + DP (Accuracy = 0.498)

Summary of Findings

To consolidate the results presented across utility and attack evaluations, Table 2 provides a comparative overview of all configurations. It highlights the trade-offs between model performance and privacy leakage across Naive RF, Federated Learning, and Federated Learning with Differential Privacy.

4.5 SGX-Based Memory Attack Evaluation

In contrast to inference-time threats such as MIA, inversion, or AIA, this section focuses on runtime memory-level attacks under a powerful adversary model. Specifically, we evaluate the ability of Intel SGX enclaves to protect sensitive inference workloads against direct memory extraction and process introspection.

Attack Setup

We trained a Random Forest model on the same synthetic healthcare dataset (55,500 records, 20 features), then deployed it inside an SGX enclave using the Gramine LibOS framework. Two attack vectors were tested:

- **Direct Memory Extraction:** Using the Linux `/proc/[pid]/mem` interface to scan process memory and search for floating-point values representing model coefficients.

- **Process Mapping Inspection:** Parsing `/proc/[pid]/maps` alongside runtime intro-

Metric / Attack Type	Naive RF	Federated	Federated + DP
Accuracy	0.895	0.901	0.901
AUC (Utility)	0.592	0.568	0.489
MIA AUC (Best Signal)	0.665 (loss)	0.561 (loss)	0.559 (loss)
Model Inversion MSE	0.0001	0.0992	0.0992
AIA Accuracy	0.546	0.604	0.498
Privacy Leakage Level	High	Medium	Low
Utility Trade-off	Baseline	Slight drop in AUC	Significant AUC drop

Table 2: Summary of utility and privacy metrics across training configurations.

spection tools such as `psutil` to locate memory regions and attempt data reconstruction.

Both attacks targeted runtime-accessible parameters (model weights) and input data (patient features), under the assumption of a privileged adversary with root-level access.

Results

The attacks were executed in both non-enclave and SGX-enclave modes. Table 3 summarizes the observed outcomes:

Insights

As illustrated in Figure 14, SGX enclaves provide hardware-level memory encryption and process isolation, eliminating the attack surface available in traditional memory layouts.

- In the unprotected mode, all attacks succeeded in recovering 10 floating-point model weights and accessing the full process memory layout.
- In SGX-protected mode, enclave memory remained encrypted and isolated: all memory access attempts failed, with zero sensitive values extracted.
- The results confirm that SGX provides robust runtime confidentiality, complementing inference-time defenses like FL and DP.

5 Conclusions and Future Work

In this work, we implemented and evaluated a privacy-preserving machine learning pipeline using Random Forest models under three core configurations: a centralized (Naive) model, Federated Learning (FL), and Federated Learning with Differential Privacy (FL + DP). We tested these models against three major inference attack vectors, Membership Inference Attack (MIA), Model Inversion, and Attribute Inference Attack (AIA), to assess both utility and privacy robustness.

Beyond inference-time privacy, we extended our evaluation to runtime memory security by deploying a trained logistic regression model inside an Intel SGX enclave. This allowed us to evaluate whether Trusted Execution Environments (TEEs) can prevent memory-based attacks targeting sensitive model parameters and patient data during inference.

Conclusions

- **Utility:** Federated Learning slightly improved accuracy over the centralized baseline while maintaining similar AUC. Adding Differential Privacy preserved accuracy but reduced AUC due to injected noise.
- **MIA Resistance:** Naive RF was highly vulnerable to MIA (AUC = 0.665), while FL reduced this to 0.561. The addition of DP slightly lowered attack performance to 0.559, suggesting moderate added value.
- **Inversion Robustness:** The centralized model was highly exposed (MSE = 0.0001), while both FL and FL + DP increased MSE to 0.0992, showing clear mitigation effects even without DP.
- **AIA Leakage:** FL led to the highest attribute inference accuracy (0.604), likely due to local overfitting. Applying DP reduced inference accuracy to near-random levels (0.498), successfully mitigating the threat.
- **SGX Protection Effectiveness:** In our hardware-based experiment, SGX successfully blocked memory extraction and process analysis attacks, reducing data exposure from 10 extracted model coefficients to 0. The enclave protected both model and patient data from access even by privileged OS-level adversaries, as summarized in table 3.
- **Defense Layers:** While FL and DP offer statistical and software-level privacy, SGX

Attack Vector	Unprotected	SGX-Protected
Memory Extraction	10 sensitive float values extracted	0 values extracted (blocked)
Process Mapping Access	Full process memory accessible via `/proc`	Restricted; enclave memory hidden
Sensitive Data Leaked	Model weights exposed	None
Protection Effectiveness	—	100.0% reduction in exposure

Table 3: Comparison of attack success in unprotected vs. SGX-protected inference.

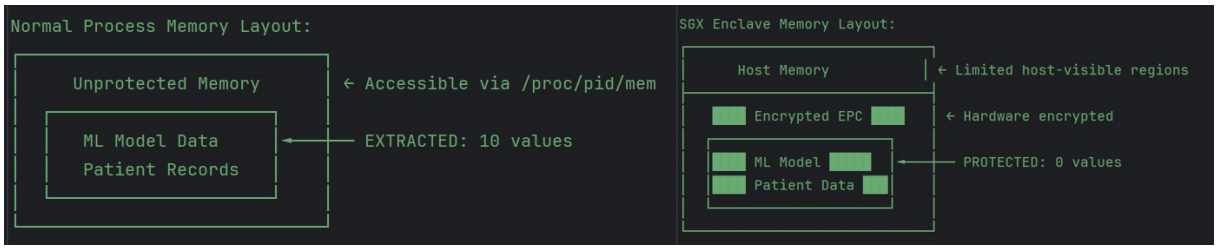

Figure 14: Memory layout comparison: in unprotected mode, model and patient data reside in accessible process memory. SGX enclaves encrypt and isolate runtime memory, preventing extraction even under privileged access.

complements these defenses by addressing runtime memory threats. Together, these approaches provide multi-layered protection across both inference and execution phases.

Future Work

Our study focused on tree-based models and basic enclave inference. Future extensions may include:

- **Gradient-based Federated Learning:** Implementing and comparing deep neural networks trained with FedAvg and DP-SGD.
- **Adaptive DP mechanisms:** Dynamically tuning the noise scale (ε) during training to optimize the privacy-utility trade-off.
- **SGX Integration with FL:** Investigating how SGX enclaves can be integrated into client-side training in federated settings to protect data *during* model updates—not just inference.
- **Advanced Threat Evaluation:** Simulating stronger adversarial models, including side-channel attacks and enclave bypass attempts, to assess SGX limitations.
- **Compliance and Deployment:** Exploring how hardware-based protections like SGX can align with HIPAA/GDPR requirements in real-world healthcare ML systems.
- **Performance Optimization:** Investigating the performance overhead of SGX in production environments and optimizing enclave memory usage and I/O patterns.

This work demonstrates that combining FL, DP, and SGX enables a more complete security posture for privacy-critical machine learning. Each layer addresses a distinct class of threats—from inference leakage to runtime memory compromise—and together, they help answer the central question: **Can PPML keep your data private under attack?**

References

Martın Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Liwei Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM.

Shangyu Chen, Yinqian Chen, and Binyu Zang. 2017. Detecting privileged side-channel attacks in shielded execution. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (AsiaCCS)*, pages 261–272.

Victor Costan and Srinivas Devadas. 2016. Intel sgx explained. Cryptology ePrint Archive. Report 2016/086.

Yilin Deng and Shu Liu. 2019. Model inversion attacks against decision trees and random forests. In *NeurIPS Workshop on Privacy-Preserving Machine Learning*.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2016. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In

Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pages 1322–1333. ACM.

Robin C Geyer, Tassilo Klein, and Mutaz Nabi. 2017. Differentially private federated learning: A client level perspective. In *NIPS Workshop on Machine Learning on the Phone and other Consumer Devices*.

Gramine Project. 2024. Gramine documentation. https://gramine.readthedocs.io/.

J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. 2009. Lest we remember: Cold boot attacks on encryption keys. *Communications of the ACM*, 52(5):91–98.

Xiang Li, Keke Huang, Wen Yang, and Xiaofeng Zhang. 2020. Differentially private data analysis for healthcare. *IEEE Transactions on Information Technology in Biomedicine*, 24(12):2821–2832.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Ag"uera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE.

Milad Nasr, Reza Shokri, and Arvind Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753. IEEE.

Prasad Patil. 2020. Healthcare dataset. https://www.kaggle.com/datasets/prasad22/healthcare-dataset. Accessed: 2025-07-30.

Michael Schwarz, Stefan Weiser, Sebastian Schwarzl, Daniel Gruss, and Stefan Mangard. 2019. Practical enclave malware with intel sgx. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, pages 177–196. Springer.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

Florian Tram'er, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2020. Stealing machine learning models via prediction apis. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 601–619.

Stacey Truex, Nicolas Baracaldo, Arslan Anwar, Thomas Steinke, Hans Ludwig, Rui Zhang, and Ramarathnam Sekar. 2020. Ldp-fed: Practical and rigorous local differential privacy in federated learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2893–2910.

Chia-Che Tsai, Donald Porter, and Mona Vij. 2017. Graphene-sgx: A practical library os for unmodified applications on sgx. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pages 645–658.

Jo Van Bulck, Otto Weisse, Frank Piessens, Wolfgang Ortner, Dmitry Evtyushkin, and Daniel Gruss. 2018. Foreshadow: Extracting the keys to the intel sgx kingdom with transient out-of-order execution. In *27th USENIX Security Symposium (USENIX Security 18)*.

Qiang Yang, Yang Liu, Tianjian Chen, and Yong Tong. 2021. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19.

Xinran Zhang, Haifeng Zhang, and Yuxin Zhang. 2020. Attribute inference attacks and defenses in machine learning. *Journal of Privacy and Confidentiality*, 10(1):1–23.