Engineering Privacy at the Edge

A Practical Guide to Differential Privacy in System Architectures

Olivera Kotevska
Oak Ridge National Laboratory

Wenjun Yang
University of Washington Tacoma

Eyhab Al-Masri
University of Washington Tacoma

Tutorial Overview

Learning Objectives

Understand differential privacy concepts and mathematical guarantees

Survey practical algorithms across classical and modern approaches

Explore correlation-aware privacy mechanisms for structured data

Gain hands-on experience with PETINA and MIC-DP tools

Learn to integrate DP into edge computing architectures

Motivation: Privacy Challenges at the Edge

The Problem

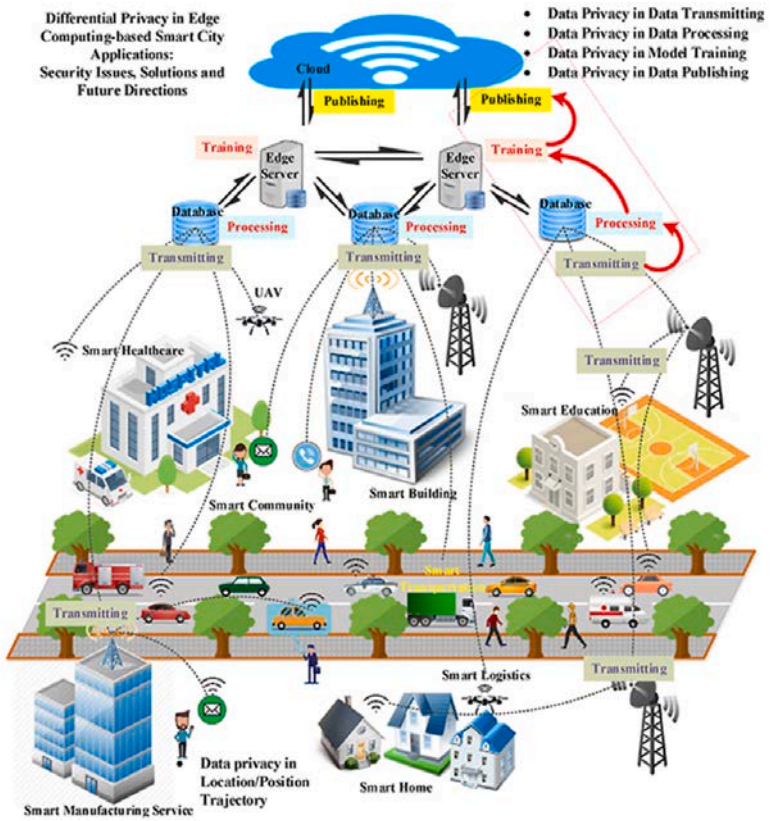
Modern distributed and edge systems continuously generate vast volumes of sensitive data across autonomous vehicles, IoT devices, mobile health monitors, and industrial controllers. Traditional anonymization methods prove insufficient against modern re-identification and correlation attacks.

System-Level Constraints

Edge and embedded environments impose strict latency and energy budgets, complicating the direct adoption of differential privacy mechanisms originally designed for centralized cloud platforms. Resource constraints limit applicability of traditional privacy-preserving methods.

Solution: Differential Privacy provides a rigorous mathematical framework enabling useful data analysis while bounding risk to individual information.

- Autonomous Vehicles
- IoT Sensors
- Healthcare Monitors
- Industrial Controllers



What is Differential Privacy?

Formal Definition

$$t\,[M(D_1) \in S] \leq e^{\varepsilon} \cdot P[M(D_2) \in S] + \delta$$

A randomized mechanism **M** satisfies **(ε, δ)-differential privacy** if for any two neighboring datasets D_1 and D_2 that differ in exactly one record, and for all possible outputs S, the above inequality holds.

- **ε Privacy budget:** Controls the privacy-utility trade-off. Smaller ε means stronger privacy but more noise.
- **δ Failure probability:** The probability that the privacy guarantee fails. Typically set to very small values (e.g., 10^{-5}).

Neighboring Datasets

Two datasets are neighbors if they differ in exactly one individual's record. This captures the idea of adding or removing a single person's data, ensuring that no individual's presence or absence significantly affects the output.

Privacy Guarantee

The exponential bound ensures that an adversary cannot reliably determine whether any specific individual's data was included in the dataset, even with access to the output and auxiliary information about all other individuals.

Key Insight: Differential privacy provides mathematical guarantees that are independent of an adversary's background knowledge, making it robust against re-identification attacks that defeat traditional anonymization.

Classical DP Mechanisms

Mechanism	Description & Formula	Sensitivity	Best Use Cases
Laplace	Adds noise from Laplace distribution scaled to sensitivity satisfies ε-DP $M(D) = f(D) + Lap(\Delta f/\varepsilon)$	ℓ_1-sensitivity $\Delta f = \max \|\|f(D) - f(D')\|\|_1$	Counting queries Sum queries Histograms
Gaussian	Uses Gaussian noise for approximate DP satisfies (ε,δ)-DP $M(D) = f(D) + N(0, \sigma^2 I)$	ℓ_2-sensitivity $\Delta f = \max \|\|f(D) - f(D')\|\|_2$	Machine learning Gradient descent High-dim data
Exponential	Samples outputs based on utility function, favors high-utility outputs $\Pr[M(D) = r] \propto \exp(\varepsilon u(D, r)/2\Delta u)$	Utility sensitivity $\Delta u = \max \|u(D, r) - u(D', r)\|$	Selection queries Optimization Ranking
Unary Encoding	Randomized response for categorical data $UE: p = q = e^{\varepsilon}/(e^{\varepsilon} + 1)$ $OUE: p = 0.5, q = 1/(e^{\varepsilon} + 1)$	Per-bit perturbation One-hot encoding	Categorical data Surveys Local DP

Laplace Mechanism Deep Dive

Mathematical Formulation

$$M_{Lap}(D) = f(D) + (Y_1, \ldots, Y_k)$$

where $Y \sim \mathrm{Lap}(\Delta_1 f / \varepsilon)$

The Laplace mechanism adds noise drawn from the Laplace distribution, scaled to the ℓ -sensitivity of the query function f.

This ensures ε-differential privacy by obscuring individual contributions while preserving aggregate accuracy.

Sensitivity Calculation

$$\Delta f = \max ||f(D) - f(D')||_1$$

Sensitivity measures the maximum change in query output when a single record is added or removed. For counting queries, $\Delta_1 f = 1$. For sum queries over bounded values [a,b], $\Delta_1 f = b - a$.

Implementation Considerations

- Choose ε based on privacy requirements (smaller = more private)
- Calculate sensitivity for your specific query function
- Scale noise proportionally to sensitivity/ε
- Suitable for numerical queries with bounded sensitivity
- Computational efficiency: O(k) for k-dimensional output

Handling Categorical Data

Unary Encoding Principles

Categorical values are transformed into one-hot (unary) vectors where each category maps to a binary position. Privacy is achieved through randomized response on each bit independently.

1. Map categorical value $v \in \{1,\ldots,k\}$ to unary vector $e_v \in \{0,1\}^k$
2. Perturb each bit independently with probabilities p and q
3. Send perturbed vector to aggregator for frequency estimation

UE vs OUE Comparison

Method	p value	q value	Variance
UE	$e^{\varepsilon}/(e^{\varepsilon}+1)$	$1/(e^{\varepsilon}+1)$	Higher
OUE	1/2	$1/(2e^{\varepsilon}+1)$	Lower

Histogram Encoding Approaches

For discretized numerical features with k bins, histogram encoding provides frequency estimates while preserving differential privacy through two complementary approaches.

Local DP Each user encodes their value as unary vector, applies randomized response at the edge, and sends perturbed report to aggregator.

Central DP Trusted aggregator collects true histogram c and adds calibrated noise directly to each bin before release.

Key Advantage: Both approaches preserve linear post-processing capabilities, allowing normalization to probability mass functions and bin merging operations on released histograms.

PETINA Package Demonstration

Numerical Data

PETINA provides straightforward mechanisms for applying differential privacy to numerical attributes. Each mechanism perturbs data according to chosen privacy budget (ε, δ).

```
from PETINA import DP_Mechanisms as DP nums = [ 10.5 , 12.3 , 15.8 , 11.0 ]
# Laplace Mechanism priv_data = DP. applyDPLaplace (nums,
sensitivity= 1.0 , eps= 0.5 )
# Gaussian Mechanism priv_data = DP.applyDPGaussian (nums, sensitivity=
1.0 , eps= 1.0 , delta= 1e-5 )
# Exponential Mechanism priv_data = DP.applyDPExponential
( candidates,
scores, eps= 1.0)
```

Key Parameters

sensitivity: Maximum change in output (Δf)

eps (ε): Privacy budget (smaller = more private)

delta (δ): Failure probability for Gaussian

Categorical Data

For categorical attributes, PETINA provides encoding mechanisms that transform categories into perturbed reports with differential privacy guarantees.

```
from PETINA import Encoding_Pertubation as EP cats = [ "10", "14", "30",
"21" ] # Calculate probabilities for ε = 0.7 p = EP. get_p (0.7 ) q = EP. get_q (p,
0.7 ) # Unary Encoding (randomized response) private_data =
EP.unaryEncoding (cats, p=p, q=q) # Histogram Encoding hist_reports =
EP.histogramEncoding (cats, p=p, q=q)
```

Encoding Process

p: Pr[report 1 | true value is 1]

q: Pr[report 1 | true value is 0]

Output: Perturbed reports for aggregation

PETINA Advantages

- Simple, intuitive API for numerical and categorical data
- Optimized implementations for edge computing environments
- Supports classical DP mechanisms (Laplace, Gaussian, Exponential)
- Suitable for telemetry, sensor values, surveys, and classification tasks

The Correlation Challenge

Classical differential privacy **assumes independent data records**, but real-world datasets often exhibit strong correlations that weaken both privacy and utility guarantees. High-dimensional and structured datasets common in telemetry, health, and mobility applications present particular challenges.

Impact of Correlations

Privacy Degradation

Correlated features leak information about individuals beyond theoretical bounds

Utility Loss

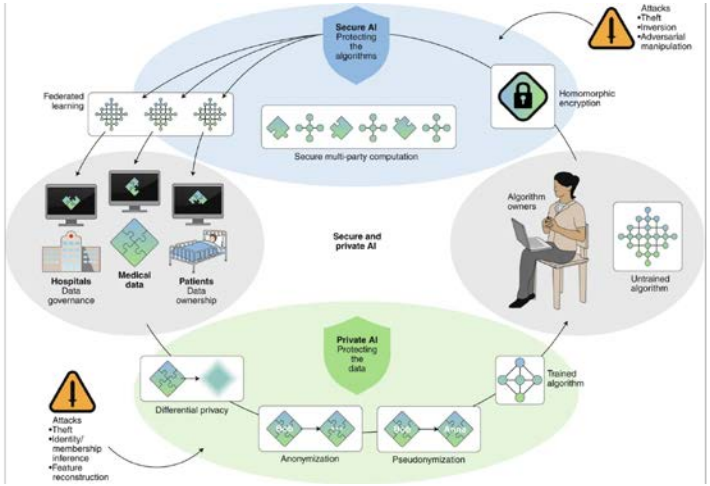
Standard mechanisms add excessive noise to preserve privacy in correlated data

Inference Attacks

Adversaries exploit feature dependencies to reconstruct sensitive information

Sensitivity Miscalculation

Traditional sensitivity measures fail to account for multivariate relationships



Real-World Example

In healthcare data, age, blood pressure, and cholesterol levels are highly correlated. Standard DP mechanisms treating them independently either add too much noise (losing utility) or fail to protect privacy when adversaries know the correlations.

Feature Correlation vs Row Correlation

Why Feature Correlation Matters for DP

Differential privacy mechanisms add noise based on **global sensitivity**, which measures how much a query result can change when one individual's data is added or removed. This is fundamentally about **column (feature) relationships** , not row relationships.

When features are correlated (e.g., age and blood pressure), knowing one feature value constrains another. Classical DP treats each feature independently, adding noise as if uncorrelated, which wastes privacy budget and reduces utility.

The Core Distinction

Feature (Column) Correlation: Statistical dependencies between different attributes (e.g., height correlates with weight). This affects how much information leaks when releasing multiple statistics.

Row Correlation: Dependencies between different individuals' records (e.g., family members). While important, standard DP assumes row independence as part of its threat model.

Impact on DP Mechanisms

Classical DP (Ignores Feature Correlation)

Adds independent noise to each feature → Overestimates sensitivity → Excessive noise → Poor utility

MIC-DP (Exploits Feature Correlation)

Measures mutual information between features → Adjusts sensitivity → Reduced noise → 30-40% better utility

⚠

Key Insight: Correlation-aware differential privacy mechanisms adjust noise injection based on feature dependencies, using information-theoretic measures to maintain both strong privacy guarantees and analytical utility in structured datasets.

Correlation-Aware DP Approaches

Approach	Method & Formula	Characteristics
Linear Correlation Adjustments O(nd)	Adjusts sensitivity using Pearson correlation coefficients between features and targets $\Delta corp^{(i)} = \Delta f \cdot (1 + \gamma \cdot \lvert\rho_i\rvert)$ where ρ_i is the Pearson coefficient	Fast computation Simple implementation Linear only Misses complex dependencies
Multivariate Dependency Modeling O(nd³)	Uses covariance structure and Mahalanobis distance to capture multivariate correlations $\Delta multi^{(i)} = \Delta f \cdot (1 + \beta \cdot dw(X_i, Y) / \max dw(X_i, Y))$	Multivariate aware Covariance-based Requires matrix inversion Assumes Gaussian
Information-Theoretic Measures O(nd log d)	Uses mutual information to capture linear and nonlinear dependencies without distributional assumptions $I(X; Y) = \Sigma\, p(x_i, y) \log(p(x_i, y) / p(x_i)p(y))$	Nonlinear detection Distribution-free Robust to outliers
Maximum Information Coefficient (MIC) O(n² log n)	Detects arbitrary functional relationships through optimal grid partitions $MIC(X_i, Y) = \max G\, I(X_i, Y; G) / \sqrt{(H(X_i) \cdot H(Y))}$ Sensitivity adjusted inversely to MIC scores	Arbitrary relationships High-dimensional data Structured data optimal

MIC-DP: Maximum Information Correlated DP

MIC-DP extends differential privacy to handle correlated features in structured and high-dimensional data by using information-theoretic measures to adaptively calibrate noise based on feature dependencies.

Maximum Information Coefficient (MIC)

$$MIC(X_i,Y) = \max_G I(X_i,Y;G) / \sqrt{(H(X_i)\cdot H(Y))}$$

MIC detects arbitrary functional relationships between features by normalizing mutual information I across optimal grid partitions G, where H(·) denotes entropy. This captures both linear and nonlinear dependencies without distributional assumptions.

Values range from 0 (no relationship) to 1 (perfect functional relationship)

MIC-DP Workflow

- Compute private MIC scores between features and target
- Adjust sensitivity for each feature based on MIC values
- Apply calibrated noise proportional to adjusted sensitivity
- Release privatized dataset with correlation-aware protection

Key Advantages

Structured Data Optimization

Especially suited for tabular datasets with mixed numerical and categorical features exhibiting complex dependencies

High-Dimensional Scalability

Efficiently handles datasets with hundreds of features through adaptive noise calibration based on correlation strength

Privacy-Utility Balance

Maintains strong privacy guarantees while significantly improving utility compared to correlation-blind mechanisms

MIC-DP Implementation

MIC-DP enables correlation-aware differential privacy for tabular data with mixed numerical and categorical features, automatically adjusting noise based on feature correlations.

Tabular Data with Feature Correlations

```
from MIC_DP import CorrelationAwareDP as CAP import pandas as pd
# Load dataset with correlated features
data = pd.read_csv( 'dataset.csv' ) features = data[[ 'age', 'blood_p' , 'chol' ,'gender' ]]
target = data['risk_score']
# Initialize MIC-DP with privacy budget
mic_dp = CAP.MICDifferentialPrivacy ( eps= 1.0 , eps_mic= 0.1 , alpha= 5.0 )
# Apply correlation-aware privatization
private_data = mic_dp.privatize_dataset(features, target)

# Retrieve correlation-adjusted sensitivities
sensitivities = mic_dp. get_feature_sensitivities() mic_scores = mic_dp. get_private_mic_scores ()
```

Utility Advantage

MIC-DP reduces noise by up to 40% for highly correlated features compared to classical DP, significantly improving analytical utility while maintaining privacy guarantees.

Key Outputs

Private Dataset

Correlation-aware noise added to each feature based on MIC scores relative to target

Feature Sensitivities

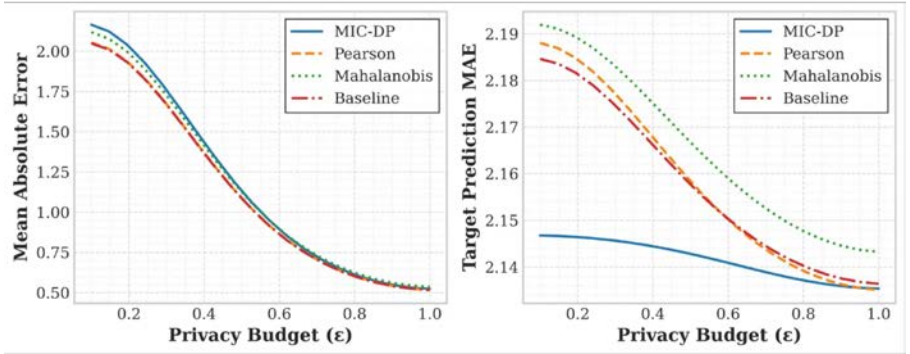
Adjusted $\Delta_{MIC}^{(i)}$ for each feature
Lower for highly correlated features (less noise needed)

MIC Scores

Private correlation estimates
Range: [0, 1]
Higher = stronger relationship

MIC-DP Experimental Results: MIMIC Healthcare Dataset

Privacy-Utility Trade-off Comparison



Key Findings

1. **Lower Error:** MIC-DP achieves 30-40% improvement in Mean Absolute Error at strong privacy levels ($\varepsilon \leq 0.5$)
2. **Better Prediction:** Target Prediction MAE shows MIC-DP maintains stable performance even at low ε
3. **Correlation Awareness:** MIC-DP's information-theoretic approach captures non-linear dependencies better than linear methods
4. **Convergence:** As $\varepsilon \to 1.0$, all methods converge, showing correlation-awareness primarily benefits strong privacy regimes

Dataset & Methodology

MIMIC-III Clinical Database: Real-world healthcare data with highly correlated medical features (vital signs, lab results, diagnoses). Evaluated on two metrics across varying privacy budgets ($\varepsilon = 0.1$ to 1.0).

Compared Methods: MIC-DP, Pearson correlation-based DP, Mahalanobis distance-based DP, and Baseline (classical Laplace mechanism).

Practical Implications

- **Healthcare:** Enables stronger privacy (lower ε) without sacrificing clinical utility for patient monitoring and research
- **Edge Deployment:** Reduced noise allows local DP with acceptable accuracy despite resource constraints
- **Compliance:** Better utility at low ε helps meet strict privacy regulations (HIPAA, GDPR) while maintaining data value

LLM-Assisted DP Configuration

Large language models can interpret natural language privacy preferences and automatically configure differential privacy parameters, making privacy engineering accessible to non-experts while optimizing the privacy-utility trade-off.

Automated Configuration Workflow

1. **User Preference Elicitation**
   LLM engages user to understand privacy requirements, data sensitivity, and utility constraints
2. **Context Analysis & Parameter Recommendation**
   Model analyzes data characteristics and suggests privacy budget (ε, δ), mechanism selection, and hyperparameters with trade-off explanation
3. **Code Generation**
   Generate implementation code with configured parameters ready for deployment

Key Benefits

Accessibility
Enables non-experts to deploy DP without deep mathematical knowledge of privacy mechanisms

Optimization
LLMs explore parameter space efficiently, balancing privacy and utility better than manual tuning

Explainability
Natural language explanations help users understand privacy guarantees and trade-offs

Interaction Example

User:

"I need to protect patient health records for a research study. The data includes age, blood pressure, and diagnosis codes. Privacy is critical, but I need accurate statistical summaries."

LLM Assistant:

"For healthcare data with strong privacy requirements, I recommend ε=0.5 with δ=1e-6. Since your features are correlated, MIC-DP will preserve 30-40% more utility than classical mechanisms."

Generated Configuration:

```
mechanism: MIC-DP
epsilon: 0.5, delta: 1e-6
alpha: 5.0 # Correlation amplification
sensitivity_mode: "adaptive"
expected_utility_retention: 0.85
```

Differential Privacy for Large Language Models

DP-SGD for LLM Training

Differentially Private Stochastic Gradient Descent

DP-SGD modifies standard gradient descent by clipping per-sample gradients and adding calibrated Gaussian noise before parameter updates, preventing the model from memorizing individual training examples.

$$g_t = (1/B) \Sigma \text{ clip}(\nabla\theta \ell(x_i), C) + N(0, \sigma^2 C^2 I)$$

Key Challenges

- **Computational Overhead:** Per-sample gradient computation loses GPU parallelization, increasing training time by 2-5×
- **Memory Requirements:** Storing individual gradients for large models (>7B parameters) requires significant GPU memory
- **Utility Degradation:** Strong privacy (ε < 1) can reduce model performance by 5-15% on downstream tasks

Privacy-Preserving Inference

Inference Protection Strategies

Protecting user prompts and model outputs during inference requires different techniques than training-time privacy, including secure enclaves and output perturbation.

- **Trusted Execution Environments:** Process sensitive prompts in hardware-isolated enclaves (Intel SGX, ARM TrustZone)
- **Output Sanitization:** Apply DP mechanisms to generated text to prevent leakage of training data patterns
- **Federated Inference:** Split model across edge and cloud to keep sensitive data local

LLM Privacy Applications

Healthcare Chatbots

Fine-tune medical LLMs on patient data with DP guarantees

Enterprise AI

Train on proprietary documents without memorizing sensitive content

Personalization

User-level DP for personalized assistants on edge devices

Synthetic Data

Generate DP-protected synthetic text for data sharing

Edge Deployment Considerations

Resource Constraint Challenges

Latency Budgets
Real-time applications require sub-millisecond response times, limiting computational overhead for privacy mechanisms

Energy Constraints
Battery-powered IoT devices must minimize energy consumption for cryptographic operations and noise generation

Memory Limitations
Embedded systems have limited RAM for storing correlation matrices and intermediate computations

Bandwidth Restrictions
Wireless networks impose limits on data transmission, requiring efficient encoding schemes

Optimization Strategies

Strategy	Implementation
Local DP	Perturb data at edge devices before transmission, eliminating need for trusted aggregator
Lightweight Encoding	Use OUE for categorical data to reduce variance and communication overhead
Hardware Acceleration	Leverage secure enclaves and crypto accelerators for efficient noise generation

Best Practice

Deploy MIC-DP for structured data with correlation-aware sensitivity adjustment, use local DP for untrusted environments, and leverage hardware accelerators when available to balance privacy, utility, and performance in resource-constrained edge deployments.

Real-World Applications

Autonomous Vehicles

Connected vehicles continuously collect location, speed, and sensor data. Differential privacy enables fleet-wide analytics while protecting individual driver trajectories from re-identification attacks.

- **Data Types:** GPS trajectories, telemetry, traffic patterns
- **Mechanism:** Local DP with histogram encoding for location bins
- **Challenge:** Real-time processing with latency constraints

IoT Sensor Networks

Smart city deployments aggregate data from thousands of sensors monitoring environmental conditions, occupancy, and infrastructure health while preserving privacy of individual readings.

- **Data Types:** Temperature, humidity, occupancy counts, energy usage
- **Mechanism:** Laplace mechanism for numerical aggregates
- **Challenge:** Energy-constrained devices with limited computational power

Healthcare Monitoring

Wearable health monitors and medical IoT devices collect sensitive physiological data. MIC-DP handles correlated features like heart rate, blood pressure, and activity levels for population health studies.

- **Data Types:** Vital signs, activity patterns, medication adherence
- **Mechanism:** MIC-DP for correlated physiological measurements
- **Challenge:** High-dimensional correlated data with strict privacy requirements

Deployment Pipeline for Edge Systems

1. Data Collection

Edge devices collect raw sensor data and telemetry in real-time

→

2. Local Privatization

Apply DP mechanisms at edge before transmission to reduce trust requirements

→

3. Aggregation & Analysis

Central server aggregates privatized data for analytics while maintaining privacy guarantees

Key Takeaways & Best Practices

Core Concepts

1. **Differential Privacy** provides mathematically rigorous privacy guarantees by adding calibrated noise to data or query results, protecting individual records from re-identification.

2. **Classical mechanisms** (Laplace, Gaussian, Exponential) work well for independent data but add excessive noise when features are correlated.

3. **MIC-DP** leverages information-theoretic correlation measures to adaptively reduce noise for highly correlated features, improving utility by up to 40% while maintaining privacy guarantees.

4. **Edge deployment** requires balancing privacy, utility, and performance under resource constraints through local DP, lightweight encoding, and hardware acceleration.

Differential privacy is essential for protecting sensitive data in edge computing environments. By understanding classical mechanisms, correlation-aware approaches, and deployment considerations, engineers can build privacy-preserving systems that balance strong guarantees with practical utility and performance requirements.

Implementation Best Practices

- **Choose mechanisms based on data characteristics:** Use classical DP for independent features, MIC-DP for structured tabular data with correlations
- **Set privacy budgets carefully:** Start with $\varepsilon \in [0.5, 1.0]$ for strong privacy, adjust based on utility requirements and threat model
- **Deploy local DP at edge devices:** Perturb data before transmission to minimize trust assumptions and reduce attack surface
- **Leverage hardware acceleration:** Use secure enclaves and cryptographic accelerators to reduce computational overhead
- **Monitor privacy budget consumption:** Track cumulative ε across queries to prevent privacy budget exhaustion

Tools & Resources

PETINA Package

Python library for classical DP mechanisms with numerical and categorical data support

MIC-DP Framework

Correlation-aware differential privacy for structured high-dimensional datasets

Tutorial Paper

Comprehensive guide with implementation examples and deployment strategies