Differentially Private Deep Learning for PII Detection and Redaction

Balancing Privacy Guarantees with Model Utility

Lanre Atoye | **Ekwelle Epalle Thomas Martial**

University of Guelph • Fall 2025

The Privacy Paradox in ML

THE CIRCULAR PROBLEM

Training PII detectors requires PII data → Models memorize sensitive information → Creates the very privacy violation we're trying to prevent

EVIDENCE OF RISK

Carlini et al. (2021): Verbatim training data extracted from GPT-2

Shokri et al. (2017): Membership inference attacks identify training records

$4.88M

Average data breach cost (2024) • $165 per record

REGULATORY PENALTIES

GDPR (EU)
€20M or 4% revenue

PIPEDA (Canada)
$100K CAD / violation

HIPAA (USA)
$1.5M per category

CCPA/CPRA (California)
$7,500 / violation

Sources: IBM Cost of a Data Breach Report 2024; GDPR Art. 83; PIPEDA s.28

Differential Privacy: The Solution

CORE GUARANTEE

Model outputs from datasets differing by one record can differ by at most e^ε

→ Individual records cannot be identified from model behavior

DP-SGD MECHANISM

1. **Gradient Clipping** — Bound individual contribution ($C=1.0$)
2. **Noise Injection** — Calibrated Gaussian noise ($\sigma=0.3\text{-}1.8$)
3. **Privacy Accounting** — Track cumulative budget via Rényi DP

PRIVACY BUDGET (E) SCALE

$\varepsilon < 1.0$ — Strong Privacy
Publication-grade protection

$\varepsilon = 1$–3 — Moderate Privacy
Suitable for most use cases

$\varepsilon = 3$–5 — Weak Privacy
Some protection remains

$\varepsilon > 5$ — Very Weak
Minimal privacy guarantees

Lower ε = More noise = Better privacy = Less accuracy

Technical Approach

MODEL ARCHITECTURE

DistilBERT-base-uncased

Token classification with BIO tagging scheme for 48 PII entity types

DATASET

PII-Masking-43K (Synthetic)

5,000 samples: 3,500 train / 750 val / 750 test • 90 unique labels

HARDWARE

CPU-Only Training

Intel i7, 32GB RAM • Democratizing privacy-preserving AI

TRAINING CONFIGURATION

Epochs	Batch Size	Clipping (C)	Noise (σ)
5	8 → 32	1.0	0.3–1.8

PRIVACY LEVELS TESTED

CRITICAL PIVOT

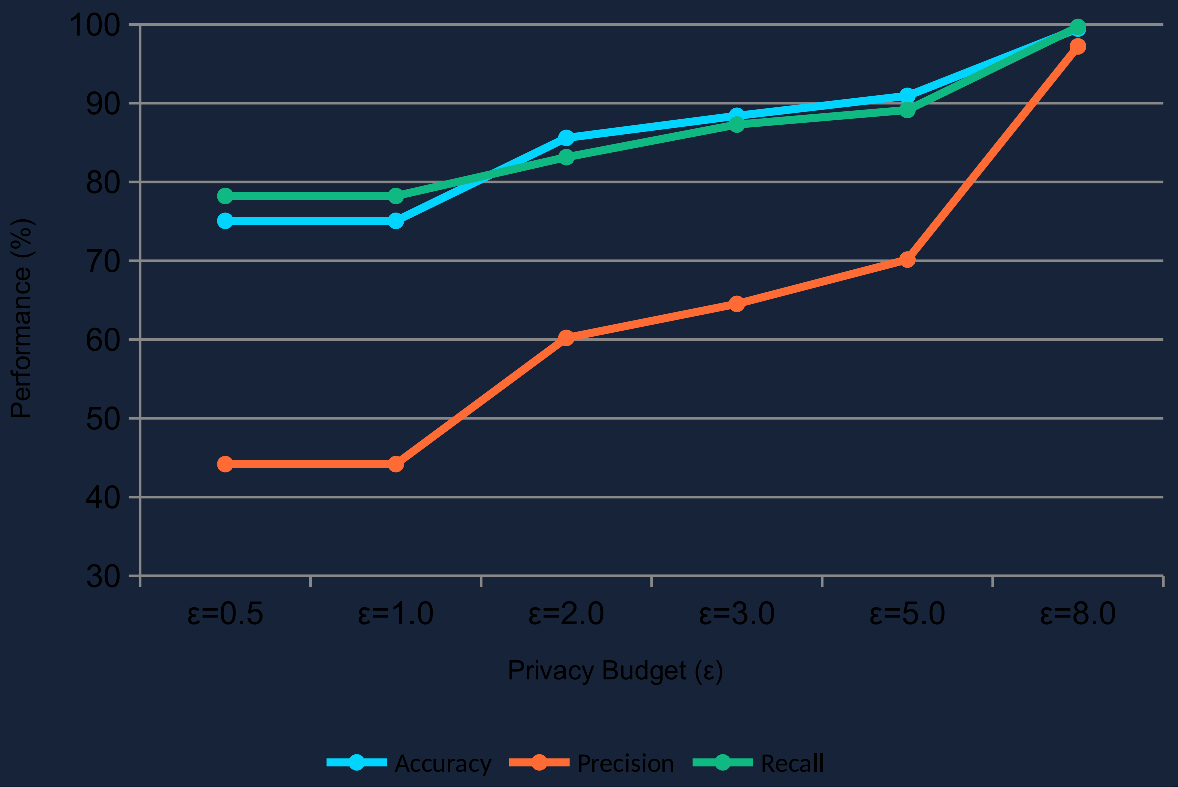
PyTorch/Opacus → JAX

Opacus compatibility issues with transformers (embedding padding, layer norm) made training impractical

8×
speedup

Days → 3-4 hrs
per model

Privacy-Utility Tradeoff



KEY INSIGHTS

+16.13%

ε=8.0 vs Regex baseline

Deep learning wins

Even with DP noise, models outperform pattern matching

Precision degrades faster

53 vs 21 pp drop — models become "trigger-happy"

⚠️ Anomaly Detected

ε=0.5 and ε=1.0 produce identical results — requires investigation

Baseline: Regex pattern matching (83.33% accuracy)

Performance by Privacy Level

Model	ε	Accuracy	Precision	Recall	F1	vs Baseline
Baseline (Regex)	∞	83.33%	80.12%	86.74%	—	—
DP Model	8.0	99.47%	97.21%	99.68%	0.984	+16.13%
DP Model	5.0	90.93%	70.15%	89.14%	0.786	+7.60%
DP Model	3.0	88.40%	64.53%	87.29%	0.742	+5.07%
DP Model	2.0	85.60%	60.21%	83.15%	0.698	+2.27%
DP Model	1.0	75.07%	44.18%	78.23%	0.562	-8.27%
DP Model	0.5	75.07%	44.18%	78.23%	0.562	-8.27%

SWEET SPOT

ε = 3.0–5.0

88–91% accuracy with meaningful privacy protection

CONFUSION MATRIX @ E=8.0

TN=621, FP=4, FN=0, TP=125

Near-perfect: only 4 false positives

CONFUSION MATRIX @ E=0.5

TN=443, FP=182, FN=5, TP=120

Heavy over-redaction: 182 false positives

ROC AUC remains >0.8 even at ε=0.5 — discriminative ability preserved, threshold/precision suffers

Redaction Pipeline: Real-World Testing

TEST CORPUS

50
Documents

- Emails
- Business reports
- Social media posts

OVERALL ACCURACY

79.8%

Stretch goal achieved
End-to-end pipeline delivered

PIPELINE FEATURES

ERROR BREAKDOWN

False Positives 12%
Non-PII text incorrectly redacted

Partial Matches 8%
Multi-word names partially captured

Format Variations 5%
Unusual PII formats missed

Challenges & Solutions

✗ Opacus-Transformers Incompatibility

Per-sample gradients, padding, layer norm issues

✓ Migrated to JAX framework

✗ Training Instability

NaN gradients, oscillation, potential collapse

✓ Gradient accumulation + careful LR tuning

✗ CPU Training Time

10-50x slower than GPU baseline

✓ DistilBERT + 5K samples + JAX = 3-4 hrs/model

✗ Token Classification Complexity

85-95% O labels, subword tokenization

✓ BIO tagging + entity-level evaluation

Key Lesson: Framework selection is as critical as algorithm design

Practical engineering decisions (PyTorch vs JAX, model size, batch strategy) determined project feasibility more than theoretical DP choices.

Conclusions & Future Work

KEY CONTRIBUTIONS

1. Privacy-Utility Analysis

Empirical analysis across 6 epsilon values for NER token classification

2. CPU-Only Training

Demonstrated feasibility — democratizing privacy-preserving AI

3. Framework Comparison

Opacus vs JAX analysis for practical deployment guidance

4. Redaction Pipeline

End-to-end system with 79.8% real-world accuracy

RECOMMENDATION

Use ε = 3.0–5.0 for most practical applications

Balances 88-91% accuracy with meaningful privacy protection

FUTURE DIRECTIONS

Scale to Full Dataset

5K → 43K samples for production readiness

Investigate ε=0.5/1.0 Anomaly

Verify training collapse vs accounting issue

Domain-Specific Testing

Legal documents, medical records, financial data

Per-Entity Performance

Break down metrics by PII type for targeted improvements

Rényi DP Mechanisms

Tighter accounting for stronger guarantees

Thank You

Questions & Discussion

PROJECT DELIVERABLES

7	14	Full
Trained Models	Page Report	Codebase

Code Repository

`github.com/Thommartial/privacy_project`

Lanre Atoye

Ekwelle Epalle Thomas Martial