Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee4159

RAPTOR: RISK-ADAPTIVE SECURITY AND PRIVACY-PRESERVING TRANSACTION PROCESSING FOR LARGE-SCALE MOBILE PAYMENT SYSTEMS

Wei Yao
University of Houston, Houston 77204, TX, USA.

Abstract: Mobile payment platforms must decide within milliseconds whether to approve, challenge or decline each transaction, while the most informative fraud signals are personal behavioural data that are fragmented across payment service providers (PSPs) and protected by data-protection law. This paper presents RAPTOR, a risk-adaptive and privacy-preserving transaction-processing framework that allocates both privacy exposure and authentication friction in proportion to the money at risk. RAPTOR (i) trains a two-stage risk model across PSPs with cross-silo federated learning, record-level differentially private SGD and pairwise-masking secure aggregation, with Rényi-DP accounting over both stages; (ii) scores every request with a Stage-1 model that uses only non-personal context and reads the customer profile through a Stage-2 model only when the prior-smoothed expected loss of the request exceeds a coverage-calibrated threshold; and (iii) maps the final calibrated risk to frictionless approval, step-up strong customer authentication or decline with an expected-loss rule that provably maximizes the number of exemptions under a fraud-loss budget. Experiments on two public payment benchmarks with 0.59 M and 1.75 M transactions distributed over five PSPs show that federated training stays within 0.012 AP of a centralized model, that the escalation gate routes 90.9% of the fraud value to profile-based verification while reading customer profiles for only 14.8% of the transactions, and that the expected-loss policy exempts 75.1% of the transactions at the reference loss budget, compared with 70.1% for a risk threshold and 35.2% for a low-value exemption rule. The online path sustains 3.25 M transactions per second on a single CPU core.
Keywords: Mobile payments; Fraud detection; Federated learning; Differential privacy; Secure aggregation; Risk-based authentication; Data minimization

1 INTRODUCTION

Mobile wallets, in-app payments and tokenized card-on-file transactions have turned payment authorization into a high-volume, latency-critical decision problem. For every request the processing chain must decide, typically within tens of milliseconds, whether the payment is approved silently, challenged with strong customer authentication (SCA), or declined. Two regulatory forces shape this decision. The revised Payment Services Directive and its regulatory technical standards require SCA by default and allow exemptions, such as transaction risk analysis or low-value payments, only while the provider keeps its fraud rate below reference levels; every unnecessary challenge adds friction and checkout abandonment [1]. The General Data Protection Regulation requires that personal data be adequate, relevant and limited to what is necessary for the purpose, which conflicts with the habit of evaluating every transaction against the complete behavioural profile of the customer [2].

Machine-learning fraud detection systems derive most of their accuracy from such profiles: aggregates of the customer's recent spending, its usual merchants and its deviation from habitual amounts [3-5]. These features are personal data held by the issuing bank or wallet provider, and in a multi-provider mobile-payment ecosystem they are fragmented across institutions that cannot pool raw records. Federated learning (FL) and differential privacy (DP) make collaborative training possible, and prior work has applied FL to card fraud detection and combined DP with secure multi-party computation for financial FL [6-13]. These studies protect the training phase, yet at inference time they still score every transaction with all features and leave the authentication decision to a fixed score threshold. Risk-based authentication, in turn, adapts the challenge to a risk estimate but ignores both the amount at stake and the privacy cost of computing the estimate [14-15].

This paper starts from the observation that privacy exposure and authentication friction are two budgets with the same natural currency: the money at risk. A €4 transit ticket with a low non-personal risk score neither justifies reading the customer's profile nor a step-up challenge, whereas a €400 transfer to an unusual merchant justifies both. We turn this observation into RAPTOR (Risk-Adaptive, Privacy-preserving Transaction ORchestration), a framework with the following contributions.

1) A two-tier privacy model of payment features and a cross-silo training pipeline in which K PSPs jointly train a non-personal Stage-1 model and a profile-based Stage-2 model with record-level DP-SGD, pairwise-masking secure aggregation and Rényi-DP accounting across both stages.

2) Risk-proportional data minimization: an escalation gate that reads a customer profile only when the prior-smoothed expected loss $m(\hat{p}_1 + \pi_0)$ of the request exceeds a threshold calibrated to cover a target share of the fraud value, with the profile access rate (PAR) as an auditable privacy indicator.

© By the Author(s) 2022, under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0).

3) An expected-loss step-up authentication policy that we prove to maximize the number of frictionless exemptions under a budget on executed fraud loss, evaluated against SCA-for-all, a €30 low-value exemption and risk thresholds with and without an exemption threshold value (ETV) cap.
4) An evaluation on two public payment benchmarks with five PSPs, three random seeds and five privacy levels, including single-core throughput and latency measurements of the online path.
Section 2 reviews related work, Section 3 states the system model, Section 4 presents RAPTOR, Sections 5 and 6 describe the experimental setup and results, and Section 7 concludes the paper.

2 RELATED WORK

2.1 Machine Learning for Payment Fraud Detection

Dal Pozzolo *et al.* formalized the operating conditions of real fraud-detection systems, including verification latency and alert-based precision metrics [3]. Correa Bahnsen *et al.* showed that transaction aggregation and periodic features describing the customer's habits are decisive, and Jurgovsky *et al.* modeled transaction sequences with LSTM networks [4-5]. Gradient-boosted trees such as XGBoost and LightGBM remain strong tabular baselines [16-17]. Because real transaction logs cannot be shared, agent-based simulators calibrated on aggregated real data have become standard benchmarks: BankSim for bank payments, PaySim for mobile money, and the reproducible simulator of the ULB/Worldline handbook for payment-card transactions with compromised terminals and customers [18-20].

2.2 Federated and Differentially Private Learning in Finance

FedAvg trains a shared model by averaging locally updated parameters; Yang *et al.* distinguished cross-silo and cross-device settings, and Kairouz *et al.* surveyed open problems including privacy and heterogeneity [6-8]. FFD trained a card fraud detector over bank silos, and federated meta-learning targeted few-shot fraud patterns [11-12]. Differential privacy bounds the influence of any single record; DP-SGD realizes it for deep models by per-example clipping and Gaussian noise, DP-FedAvg lifts it to federated training, and Rényi DP provides tight accounting for subsampled Gaussian mechanisms [9-10, 21-23]. Secure aggregation hides individual updates from the server [24]. Byrd and Polychroniadou combined DP with secure multi-party computation for a logistic fraud model [13]. RAPTOR builds on these components and extends privacy protection from training to inference-time data access and to the authentication decision.

2.3 Risk-Based Authentication and Regulation

Risk-based authentication (RBA) requests additional factors only when a login looks unusual: Freeman *et al.* proposed a statistical model of user authenticity, and Wiefling *et al.* measured RBA deployments in the wild [14-15]. In payments, the PSD2 technical standards exempt remote transactions from SCA under transaction risk analysis only below an ETV of €100, €250 or €500 that depends on the provider's fraud rate, and exempt low-value payments up to €30 [1]. Existing RBA designs threshold a risk score; RAPTOR instead ranks requests by expected monetary loss and enforces an explicit loss budget.

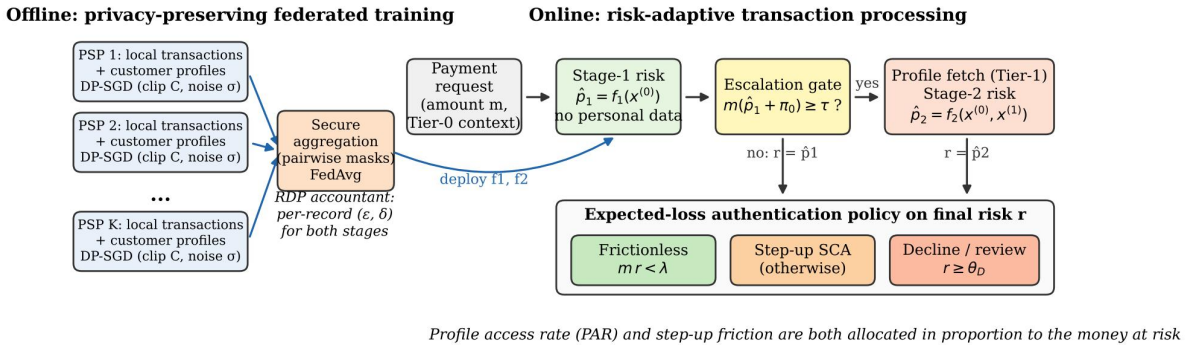


Figure 1 Architecture of RAPTOR

Note: Offline, PSPs train the Stage-1 and Stage-2 risk models with DP-SGD and secure aggregation. Online, every request is scored without personal data; the customer profile is fetched only when the prior-smoothed expected loss exceeds τ; the final calibrated risk drives the expected-loss authentication policy.

2.4 Cost-Sensitive and Cascaded Decisions

Cost-sensitive learning and example-dependent costs recognize that misclassification costs vary with the transaction amount, and probability calibration turns scores into decision-ready probabilities [4, 25-26]. Classifier cascades reduce test-time feature cost by evaluating expensive features only for uncertain inputs [27]. RAPTOR treats the customer

profile as the expensive feature, measures its cost in privacy exposure rather than computation, and routes requests by monetary rather than purely statistical uncertainty.

3 SYSTEM MODEL AND PROBLEM STATEMENT

3.1 Entities and Data

We consider K PSPs (issuers or wallet providers), each holding the transactions and behavioural profiles of its own customers, a risk-processing node that receives authorization requests, and an aggregation server used during training. A request carries its amount m and a Tier-0 vector $x^{(0)}$ of non-personal context: amount, time, and merchant- or terminal-level aggregates. The Tier-1 vector $x^{(1)}$ contains personal data: the customer's transaction counts and average amounts over sliding windows, its deviation from habitual amounts, its familiarity with the merchant, and demographic attributes. Tier-1 is released by the customer's PSP only on an explicit profile query.

3.2 Threat Model

The aggregation server and the risk node are honest-but-curious: they follow the protocol but may try to infer individual records from what they observe. Released models may be accessed by external adversaries mounting membership or attribute inference. PSPs do not share raw data with each other. Fraudsters control stolen credentials and pass a step-up challenge with probability p_b (e.g., through SIM swapping or social engineering); we use $p_b = 0.1$.

3.3 Objectives

RAPTOR pursues five objectives: (O1) high detection quality; (O2) record-level (ε, δ)-DP for every released model; (O3) inference-time data minimization, measured by the profile access rate PAR, the fraction of requests whose Tier-1 profile is read; (O4) maximal frictionless share under a budget ρ on executed fraud loss; and (O5) throughput and latency compatible with real-time authorization. Let $a_i \in \{E, S, D\}$ denote exemption, step-up SCA and decline for request i with label y_i. The executed fraud-loss rate is

$$L(a) = \frac{1}{M}\sum_i m_i y_i (\mathbb{1}[a_i = E] + p_b \mathbb{1}[a_i = S]) \quad (1)$$

where $M = \Sigma_i m_i$ is the processed volume; declined fraud is prevented, exempted fraud is executed, and challenged fraud is executed with probability p_b.

4 THE RAPTOR FRAMEWORK

Figure 1 summarizes the framework. Offline, the PSPs train two risk models: f_1 on Tier-0 features and f_2 on Tier-0 and Tier-1 features. Online, every request is scored by f_1; the profile is fetched and f_2 is evaluated only for requests that pass the escalation gate; and the final calibrated risk drives the authentication policy. Algorithm 1 lists the complete procedure.

4.1 Federated Differentially Private Training

Each PSP k runs DP-SGD on its n_k records with Poisson sampling at rate $q_k = B/n_k$ [10]. Every per-example gradient g_i is clipped to norm C,

$$\bar{g}_i = g_i \cdot min\left(1, \frac{C}{\|g_i\|}\right) \quad (2)$$

and the noisy batch gradient is

$$\tilde{g} = \frac{1}{B}\left(\sum_{i\in\mathcal{B}} \bar{g}_i + \mathcal{N}(0, \sigma^2 C^2 I)\right) \quad (3)$$

with the expected batch size B as normalizer. For a multilayer perceptron the per-example norms are available in closed form, since the gradient of a dense layer l is the outer product of the back-propagated error $\delta_i^{(l)}$ and the layer input $a_i^{(l-1)}$:

$$\|\nabla \ell_i\|^2 = \sum_l \|\delta_i^{(l)}\|^2 \left(\|a_i^{(l-1)}\|^2 + 1\right) \quad (4)$$

so clipping costs one extra row-wise dot product per layer instead of materializing per-example gradients. After one local epoch, the PSPs send weighted updates $\Delta_k = (n_k/n)(w_k - w_t)$ through secure aggregation [24]. Updates are encoded in 24-bit fixed point over the integers modulo 2^{64}, and PSP k adds pseudo-random masks derived from pairwise seeds s_{kj},

$$\tilde{u}_k = u_k + \sum_{j>k} PRG(s_{kj}) - \sum_{j<k} PRG(s_{jk}) \ mod\ 2^{64} \quad (5)$$

so that the masks cancel in the sum and the server learns only $w_{t+1} = w_t + \Sigma_k \Delta_k$. Each PSP chooses its own noise multiplier σ_k so that the $T_k = R\lceil n_k/B \rceil$ sampled-Gaussian steps it performs satisfy the target guarantee under RDP accounting [22-23]:

$$\varepsilon(\delta) = min_{\alpha>1}\left[T_k \varepsilon_\alpha(q_k, \sigma_k) + \frac{log(1/\delta)}{\alpha - 1}\right] \quad (6)$$

Because every record belongs to exactly one PSP, the per-record guarantee of the global model equals that of its PSP, and aggregation is post-processing. Training f_1 and f_2 on the same records composes in the RDP domain, which we report as the joint budget. Features are scaled with data-independent transforms (log1p with public upper bounds), so no

normalization statistics of private data are released. Isotonic calibration maps and decision thresholds are fitted on a separate validation period; these low-dimensional statistics lie outside the stated (ε, δ) guarantee, which covers the model parameters.

4.2 Risk-Proportional Escalation

Stage-1 scores are calibrated with isotonic regression, $\hat{p}_1 = g_1(f_1(x^{(0)}))$ [26]. Tier-0 features cannot observe customer-specific anomalies such as an account takeover that spends five times the customer's usual amount, so $\hat{p}_1$ systematically underestimates exactly the risk that the profile reveals. RAPTOR therefore escalates on a prior-smoothed expected loss,

$$e_i = \mathbb{1}\left[m_i\left(\hat{p}_{1,i} + \pi_0\right) \geq \tau\right], \; r_i = e_i\hat{p}_{2,i} + (1 - e_i)\hat{p}_{1,i} \tag{7}$$

where π_0 is the prior fraud rate and $\hat{p}_2 = g_2(f_2(x^{(0)}, x^{(1)}))$. For confidently risky requests the key behaves like the expected loss $m\ \hat{p}_1$; for requests that look benign without context it reduces to a stake-based rule $m\ \pi_0$, so large payments are not cleared on non-personal evidence alone. The threshold τ is the largest value whose escalated validation set still covers at least $1 - \eta$ of the fraud value ($\eta = 0.1$), which directly encodes a verification objective that compliance teams can audit.

4.3 Expected-Loss Step-Up Authentication

Given calibrated risk r_i, exempting request i instead of challenging it raises the expected executed loss by $(1 - p_b)\ m_i\ r_i/M$. The policy problem is

$$max_a \sum_i \mathbb{1}[a_i = E] \quad s.t. \;\; \mathbb{E}[L(a)] \leq \rho \tag{8}$$

Proposition 1. With calibrated risks and a fixed decline set, the rule that exempts requests in increasing order of $m_i r_i$ until the budget is exhausted, i.e., $a_i = E$ iff $m_i r_i < \lambda$, is optimal for (8).

Proof sketch. Problem (8) is a 0/1 knapsack in which every exemption has unit value and weight $w_i = (1 - p_b)m_i r_i$. For unit values, any feasible set containing an item heavier than an excluded one can swap them without losing value or feasibility, so the set of the lightest items is optimal; it is the sub-level set of $m_i r_i$. ∎

A threshold on r_i alone is optimal only when all amounts are equal. RAPTOR declines requests with $r_i \geq \theta_D$, where θ_D is the lowest threshold whose validation precision is at least 90%, and selects λ on the validation period by a single cumulative-sum pass over the sorted keys, which is exact because the loss is non-decreasing along the ordering. The resulting rule is

$$a_i = D \; if \; r_i \geq \theta_D; \;\; E \; if \; m_i r_i < \lambda; \;\; S \; otherwise. \tag{9}$$

Algorithm 1 RAPTOR Training and Online Processing

Input: PSP data $\{D_k\}$, rounds R, batch B, clip C, budget (ε, δ), coverage η, loss budget ρ
Offline (cross-silo, per stage $s \in \{1, 2\}$)
1: for each PSP k: $\sigma_k \leftarrow$ CalibrateRDP(ε, δ, $q_k = B/n_k$, $T_k = R\lceil n_k/B \rceil$)
2: for $t = 1, \ldots, R$ do
3: each PSP k: one local epoch of DP-SGD from w_t with (C, σ_k), Eqs. (1)–(3)
4: server: $w_{t+1} \leftarrow w_t + \text{SecAgg}(\{(n_k/n)\Delta_k\})$, Eqs. (4)–(5)
5: fit isotonic maps g_1, g_2; select τ (fraud-value coverage $\geq 1 - \eta$), θ_D, λ (loss $\leq \rho$)
Online, for each payment request ($x^{(0)}$, m)
6: $\hat{p}_1 \leftarrow g_1(f_1(x^{(0)}))$; if $m(\hat{p}_1 + \pi_0) \geq \tau$: fetch $x^{(1)}$, $r \leftarrow g_2(f_2(x^{(0)}, x^{(1)}))$; else $r \leftarrow \hat{p}_1$
7: if $r \geq \theta_D$: DECLINE; elif $m \cdot r < \lambda$: FRICTIONLESS; else: STEP-UP SCA

The online cost of RAPTOR is one small MLP and one calibration lookup per request, plus a profile query and a second MLP for the escalated fraction. Since τ and λ are scalars, they can be re-tuned whenever the loss budget or the fraud environment changes, without retraining the models.

5 EXPERIMENTAL SETUP

5.1 Datasets and Federated Partition

We use two public payment benchmarks released no later than 2022 (Table 1). BankSim simulates 180 days of payments by 4,112 customers at 50 merchants, calibrated on aggregated data of a Spanish bank, with customer age and gender [18]. FDH is the simulated transaction dataset of the ULB/Worldline fraud-detection handbook: 183 days of payments by 4,990 customers at 10,000 terminals, with three fraud scenarios (high amounts, compromised terminals, compromised customers whose spending is multiplied) [20]. Both streams follow the customer-to-merchant structure of wallet and card-on-file mobile payments. For BankSim, Tier-0 contains amount, merchant category, merchant activity and merchant fraud rates over 7- and 30-day windows computed with a 7-day label delay; Tier-1 contains customer activity windows, amount ratios, merchant familiarity, age and gender. For FDH we use the handbook's terminal features as Tier-0 and its customer features plus amount ratios as Tier-1. Splits are temporal with one-week gaps, and the first 30 days only warm up the sliding windows. Customers are assigned to $K = 5$ PSPs with Dirichlet(1) proportions, which yields strongly unequal silos (BankSim: 11 k–121 k training records; FDH: 18 k–350 k).

Table 1 Dataset Statistics

Property	BankSim [18]	FDH [20]
Transactions	594,643	1,754,155
Customers / merchants (terminals)	4,112 / 50	4,990 / 10,000
Fraud rate (count)	1.21%	0.84%
Fraud share of volume (value)	16.97%	2.05%
Train / val. / test transactions	294 k / 83 k / 85 k	882 k / 230 k / 221 k
Test frauds	920	1,948
Tier-0 / Tier-1 features	22 / 19	11 / 8
PSPs (Dirichlet $\alpha = 1$)	5	5

5.2 Models, Baselines and Metrics

Both stages use an MLP with hidden layers of 64 and 32 ReLU units (3.4 k–4.8 k parameters), trained for $R = 30$ rounds of one local epoch with expected batch 512, momentum 0.9, positive-class weight 3, learning rate 0.1 (non-private) or 0.3 (DP), clipping norm $C = 1$ and $\delta = 10^{-6}$. We compare centralized logistic regression (LR), XGBoost (300 trees, depth 6), LightGBM (300 trees, 31 leaves) and the MLP, a local-only setting in which each PSP trains alone and scores its own customers, non-private FedAvg with secure aggregation, and DP-FedAvg with $\varepsilon \in \{0.5, 1, 2, 4, 8\}$ per stage [6, 16-17]. MLP-based results are averaged over three seeds. We report ROC-AUC and average precision (AP, the primary metric under class imbalance); for data minimization the value-weighted AP ($AP_{\$}$), the fraud-value coverage (FVC, share of fraud value escalated) and PAR; and for authentication the frictionless share F and executed loss L from (1) with $p_b = 0.1$ [3]. Loss budgets are multiples κ of the reference loss $\rho_{ref} = p_b \times$ fraud-value share, i.e., the loss incurred if every transaction were challenged and none declined. All components are implemented in Python/NumPy and run on one core of a cloud virtual machine with 4 GB of memory.

Table 2 Test Performance of Centralized, Local and Federated Training (AUC / AP; Mean ± Std over Three Seeds for MLP Models)

Method	Privacy	BankSim S1 AUC	BankSim S1 AP	BankSim S2 AUC	BankSim S2 AP	FDH S1 AUC	FDH S1 AP	FDH S2 AUC	FDH S2 AP
LR	–	0.9975	0.884	0.9976	0.897	0.8709	0.529	0.8841	0.649
XGBoost [16]	–	0.9985	0.915	0.9991	0.949	0.8688	0.615	0.9023	0.758
LightGBM [17]	–	0.9985	0.913	0.9991	0.945	0.8707	0.625	0.9003	0.758
MLP (centralized)	–	0.9988	0.927 ± 0.000	0.9990	0.945 ± 0.002	0.8747	0.664 ± 0.005	0.9067	0.756 ± 0.005
MLP local-only	–	0.9985	0.912 ± 0.001	0.9986	0.923 ± 0.004	0.8704	0.645 ± 0.002	0.8986	0.726 ± 0.003
FedAvg + SecAgg	$\varepsilon = \infty$	0.9986	0.922 ± 0.001	0.9990	0.944 ± 0.001	0.8758	0.663 ± 0.001	0.9038	0.744 ± 0.002
DP-FedAvg (RAPTOR)	$\varepsilon = 8$	0.9972	0.892 ± 0.002	0.9973	0.904 ± 0.001	0.8690	0.590 ± 0.008	0.8893	0.690 ± 0.005
DP-FedAvg (RAPTOR)	$\varepsilon = 4$	0.9972	0.892 ± 0.003	0.9973	0.901 ± 0.002	0.8680	0.577 ± 0.020	0.8886	0.685 ± 0.007
DP-FedAvg (RAPTOR)	$\varepsilon = 1$	0.9964	0.886 ± 0.005	0.9969	0.873 ± 0.006	0.8511	0.520 ± 0.017	0.8865	0.645 ± 0.017

Note: S1: Stage-1 (Tier-0 features); S2: Stage-2 (Tier-0 + Tier-1). ε is the per-stage budget at $\delta = 10^{-6}$; LR, XGBoost and LightGBM are deterministic.

6 RESULTS

6.1 Detection Quality under Federation

Table 2 reports test performance. Federated training closes almost the entire gap to centralization: FedAvg with secure aggregation is within 0.001–0.012 AP of the centralized MLP on all four task variants and outperforms local-only training by 0.010–0.021 AP, with the largest gains on the profile-based Stage-2 models, where the small PSPs lack enough fraud examples on their own. On FDH, the centralized MLP and the gradient-boosted trees reach comparable AP (0.756 versus 0.758), so the federated MLP loses little against the strongest centralized baselines while never moving raw records. The tiering itself is informative: adding Tier-1 profile features lifts FDH AP from 0.663 to 0.744 under FedAvg and from 0.520 to 0.645 under DP at $\varepsilon = 1$, because compromised-customer fraud is visible only relative to the customer's own history. On BankSim, Tier-0 features already separate most fraud (AP 0.922), since fraud concentrates in a few merchant categories and high amounts.

6.2 Privacy–Utility Trade-Off

Figure 2 shows AP as a function of the per-stage budget. Utility degrades gracefully down to $\varepsilon = 1$ and saturates beyond $\varepsilon = 2$–4. On FDH at $\varepsilon = 1$, the private Stage-2 model (AP 0.645) is still close to the non-private Stage-1 model trained without any profile data (0.663), and at $\varepsilon = 8$ it reaches 0.690. On BankSim, the noise required at $\varepsilon = 1$ offsets the

benefit of the 19 additional Tier-1 inputs, so the private Stage-2 model (0.873) slightly trails the private Stage-1 model (0.886); this is precisely the regime in which RAPTOR's escalation gate, described next, reads almost no profiles. Table 3 details the accounting. Smaller PSPs need larger noise multipliers because their sampling rate q_k is higher, and the joint budget of the two stages under RDP composition is 1.44 for $\varepsilon = 1$ per stage, well below the 2.0 of basic composition. Figure 3 shows that DP-FedAvg converges within about 20 rounds on both datasets.

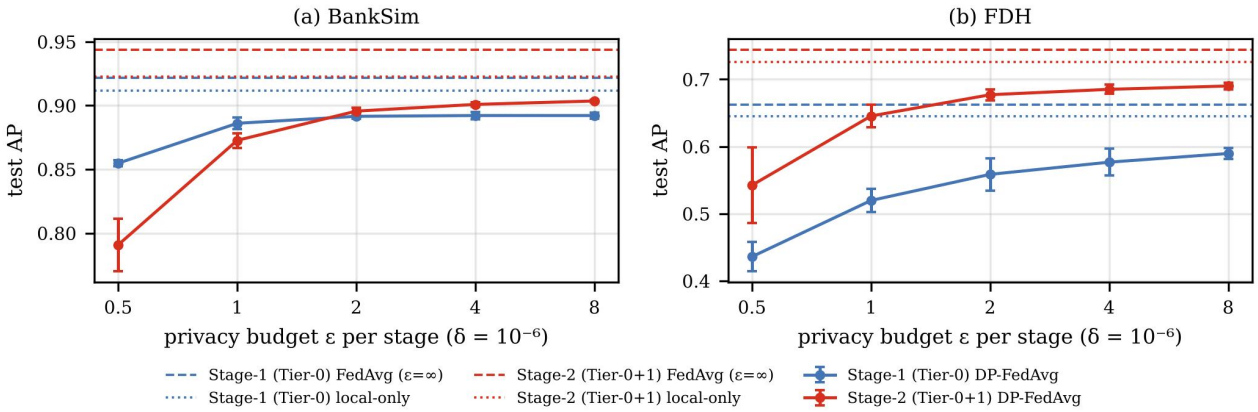


Figure 2 Privacy–Utility Trade-Off of DP-FedAvg (Mean ± Std over Three Seeds) on (a) BankSim and (b) FDH
Note: Dashed and dotted lines: non-private FedAvg and local-only training of the same stage.

Table 3 Privacy Accounting and Training Cost (Stage-2, ε = 1)

Quantity	BankSim	FDH
Noise multiplier σ_k (5 PSPs)	1.79–5.52	1.20–4.28
Sampling rate q_k	0.0042–0.046	0.0015–0.028
DP-SGD steps T_k	660–7,110	1,080–20,550
Joint ε, two stages (per-stage 1 / 2 / 4)	1.44 / 2.89 / 5.79	1.44 / 2.89 / 5.78
Model parameters	4,801	3,393
Traffic per round (K = 5, up + down)	375 KiB	265 KiB
Training time, FedAvg / DP (30 rounds)	5.7 s / 9.3 s	15.2 s / 26.0 s
Secure-aggregation time per round	1.4 ms	1.4 ms

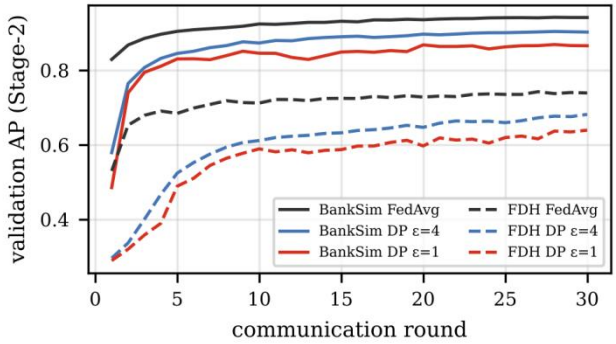


Figure 3 Validation AP of the Stage-2 Model across Communication Rounds

6.3 Risk-Proportional Data Minimization

Figure 4 and Table 4 compare escalation strategies for the private FDH models at equal PAR. Random escalation buys detection quality linearly with privacy exposure. Ranking by the Stage-1 risk score is the weakest informed rule, because it spends profile queries on low-value transactions. The pure expected loss $m\ \hat{p}_1$ improves FVC substantially, and the prior-smoothed key of RAPTOR achieves the highest FVC at every PAR: at 10% PAR it routes 88.8% of the fraud value to profile-based verification, against 80.0% for the risk score and 75.7% for an amount rule. The amount rule obtains a slightly higher $AP_\$$ at small PARs (0.765 versus 0.755 at 10% PAR), but it leaves 13–15 percentage points more fraud value unverified, which matters directly for the loss budget of Section 6.4. With the coverage target $\eta = 0.1$, the gate selects a PAR of 14.8% on FDH: 85.2% of the profile queries are avoided, FVC is 90.9% and $AP_\$$ reaches 0.763, i.e., 98.5% of the 0.775 obtained with full profile access. On BankSim, the gate selects a PAR of only 0.96% while still covering 96.6% of the fraud value; AP remains at 0.872, compared with 0.866 under full access, since the private profile model brings no gain there.

Table 4 Escalation Strategies on FDH (DP-FedAvg, ε = 1): Value-Weighted AP / FVC (%)

Strategy	PAR 5%	PAR 10%	PAR 20%
Random	0.520 / 4.3	0.541 / 9.2	0.582 / 20.8
Amount rule	0.755 / 71.6	0.765 / 75.7	0.772 / 82.4
Risk score $\hat{p}_1$	0.665 / 72.4	0.704 / 80.0	0.746 / 87.8
Expected loss m $\hat{p}_1$	0.730 / 82.7	0.749 / 87.1	0.763 / 92.0
RAPTOR m($\hat{p}_1+\pi_0$)	**0.745 / 85.7**	**0.755 / 88.8**	**0.767 / 92.5**
No access / full access	0.508 / 0	–	0.775 / 100

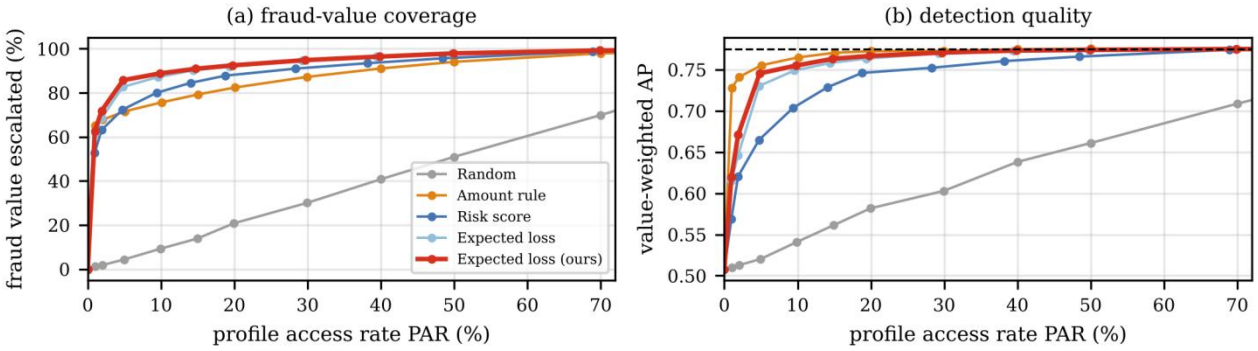


Figure 4 Risk-Proportional Escalation on FDH (DP-FedAvg, ε = 1 per Stage): (a) Share of the Fraud Value Whose Customer Profile Is Checked and (b) Value-Weighted AP of the Cascade, as Functions of the Profile Access Rate

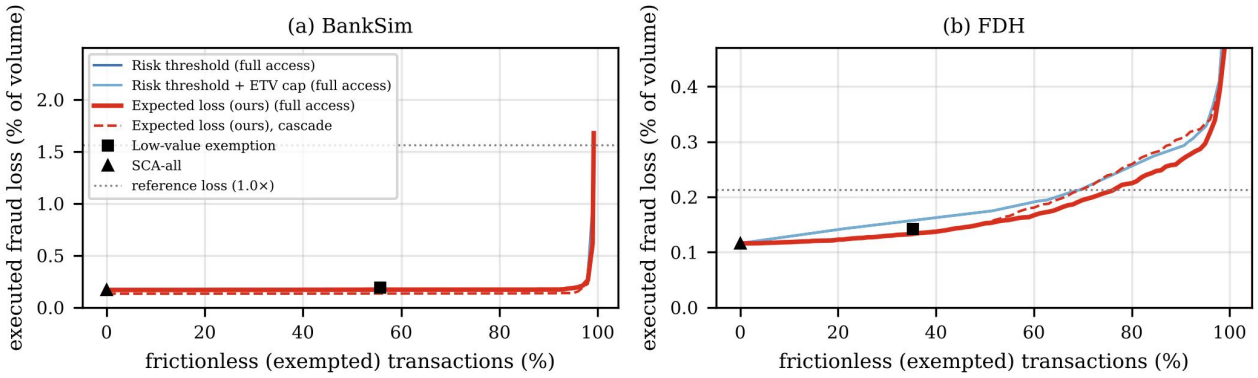


Figure 5 Frictionless Share versus Executed Fraud Loss of the Authentication Policies on (a) BankSim and (b) FDH (Test Period, DP-FedAvg ε = 1)

Note: Markers: SCA for every transaction and the €30 low-value exemption; dotted line: reference loss ρ_{ref}.

6.4 Risk-Adaptive Step-Up Authentication

Figure 5 traces the frictionless share against the executed loss for every threshold, and Table 5 reports operating points tuned on the validation period and measured on the test period. On FDH with full profile access, the expected-loss rule exempts 75.1% of the transactions at the reference budget ($\kappa = 1$, $\rho = 0.214\%$ of volume), compared with 70.1% for the risk threshold and 35.2% for the €30 low-value exemption. At the stricter budget $\kappa = 0.75$, the gap widens to 51.8% versus 21.3%: a risk-only rule must stop exempting once high-value transactions with moderate risk would exceed the budget, whereas the expected-loss rule continues to exempt small payments with the same risk. An ETV cap of €500 does not change the risk-threshold policy on FDH, where almost all payments are below €500. The decline rule blocks 44.7% of the fraud value while declining only 0.039% of genuine transactions. With the privacy-minimized cascade (PAR 14.8%), the expected-loss policy still exempts 66.3% at $\kappa = 1$, against 43.0% when no profile is read at all and 38.6% for a risk threshold on the same cascade scores; the remaining distance to full access is the price of reading 85.2% fewer profiles. On BankSim, all risk-based policies exempt more than 99% of the transactions within budget because declines alone stop 88.2% of the fraud value, while the low-value exemption covers only 55.8%; the expected-loss rule again yields the largest exempt share (99.18% versus 99.08% at $\kappa = 1$).

Table 5 Authentication Policies on FDH: Frictionless Share F (%) / Executed Loss L (% of Volume)

Policy (profile access)	κ = 0.75	κ = 1.0	κ = 1.25
Budget ρ	0.160	0.214	0.268
SCA for all	0.0 / 0.116	0.0 / 0.116	0.0 / 0.116
Low-value exemption ≤ €30	35.2 / 0.142	35.2 / 0.142	35.2 / 0.142
Risk threshold (full)	21.3 / 0.142	70.1 / 0.216	70.1 / 0.216
Risk thr. + €500 ETV cap (full)	21.3 / 0.142	70.1 / 0.216	70.1 / 0.216
Expected loss (full)	**51.8 / 0.154**	**75.1 / 0.210**	**88.9 / 0.258**

Policy (profile access)	κ = 0.75	κ = 1.0	κ = 1.25
Expected loss (none, PAR 0%)	0.0 / 0.174	43.0 / 0.199	65.0 / 0.252
Risk threshold (cascade)	35.6 / 0.153	38.6 / 0.160	73.6 / 0.252
Expected loss (cascade, PAR 14.8%)	**44.8 / 0.143**	**66.3 / 0.200**	**82.5 / 0.273**

6.5 Scalability and Overhead

Table 6 reports single-core scoring performance on the 220,887-transaction FDH test stream. With micro-batches of 8,192 requests, the full Stage-2 MLP scores 4.75 M transactions per second and the RAPTOR cascade 3.60 M, about 23 times faster than LightGBM on the same features. Per-request processing (batch size 1) stays below 150 µs at the 99th percentile for the cascade, far below typical authorization budgets. Replaying the test stream ten times (2.2 M requests) sustains 3.25 M transactions per second with profile queries for 14.75% of the requests. In a deployment where each profile query is a remote lookup at the customer's PSP, the cascade removes 85% of these round trips from the authorization path. On the training side, DP-SGD with closed-form clipping costs 1.6–1.7 times the time of non-private FedAvg, secure aggregation adds 1.4 ms per round for five PSPs, and the full federated training of a stage takes under 30 s for 0.88 M records.

Table 6 Online Scoring on One CPU Core (FDH Test Stream)

Scoring path	B = 1: tx/s (p99 µs)	B = 64: tx/s	B = 8192: tx/s
Stage-2 MLP on all requests	54.3 k (39.9)	2.03 M	4.75 M
RAPTOR cascade (PAR 14.8%)	**9.7 k (146.6)**	**0.42 M**	**3.60 M**
LightGBM, Stage-2 features	2.4 k (848.9)	0.07 M	0.16 M

7 DISCUSSION AND CONCLUSION

The experiments support a simple design principle for payment risk engines: privacy exposure and authentication friction should both be spent where the money at risk is high. The federated pipeline gives PSPs a shared model with record-level guarantees at a modest cost in AP; the escalation gate turns the profile access rate into a tunable, auditable quantity that operators can report alongside the fraud rate; and the expected-loss policy uses the resulting risk estimates more efficiently than score thresholds or flat low-value exemptions. The loss budget ρ plays the role of the fraud-rate ceilings that govern SCA exemptions in practice, so the same scalars τ and λ can be re-tuned when the regulatory tier or the fraud environment changes, without retraining.

RAPTOR is modular. Gradient-boosted trees with federated split finding, sequence encoders of the customer history or adaptive clipping can replace the MLPs without changing the gate or the policy, and amount-banded budgets can mirror the exemption tiers of the PSD2 technical standards more closely [1, 5]. Future work will extend the framework to streaming retraining under delayed labels, to differentially private release of the calibration maps, and to adversaries who adapt their amounts to the escalation key [3].

In summary, RAPTOR unifies cross-silo differentially private learning, risk-proportional data minimization and expected-loss step-up authentication into one transaction-processing framework. On two public payment benchmarks it matches centralized accuracy within 0.012 AP, reads customer profiles for 14.8% of the requests while covering 90.9% of the fraud value, raises the frictionless share at a fixed loss budget from 70.1% to 75.1% (from 21.3% to 51.8% under a tighter budget), and processes millions of transactions per second on a single core.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] European Commission. Commission Delegated Regulation (EU) 2018/389 of 27 November 2017 supplementing Directive (EU) 2015/2366 with regard to regulatory technical standards for strong customer authentication and common and secure open standards of communication. Official Journal of the European Union, 2018, L 69: 23-43.

[2] European Parliament and Council. Regulation (EU) 2016/679 (General Data Protection Regulation). Official Journal of the European Union, 2016, L 119: 1-88.

[3] Dal Pozzolo A, Boracchi G, Caelen O, et al. Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(8): 3784-3797.

[4] Correa Bahnsen A, Aouada D, Stojanovic A, et al. Feature engineering strategies for credit card fraud detection. Expert Systems with Applications, 2016, 51: 134-142.

[5] Jurgovsky J, Granitzer M, Ziegler K, et al. Sequence classification for credit-card fraud detection. Expert Systems with Applications, 2018, 100: 234-245.

[6] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017: 1273-1282.

[7] Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 2019, 10(2): 12:1-12:19.
[8] Kairouz P, McMahan H B, et al. Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 2021, 14(1-2): 1-210.
[9] Dwork C, Roth A. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 2014, 9(3-4): 211-407.
[10] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), 2016: 308-318.
[11] Yang W, Zhang Y, Ye K, et al. FFD: A federated learning based method for credit card fraud detection. In: Proceedings of the International Conference on Big Data (BigData 2019), LNCS 11514, Springer, 2019: 18-32.
[12] Zheng W, Yan L, Gou C, et al. Federated meta-learning for fraudulent credit card detection. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI), 2020: 4654-4660.
[13] Byrd D, Polychroniadou A. Differentially private secure multi-party computation for federated learning in financial applications. In: Proceedings of the 1st ACM International Conference on AI in Finance (ICAIF), New York, NY, USA, 2020.
[14] Freeman D, Jain S, Dürmuth M, et al. Who are you? A statistical approach to measuring user authenticity. In: Proceedings of the Network and Distributed System Security Symposium (NDSS), 2016.
[15] Wiefling S, Lo Iacono L, Dürmuth M. Is this really you? An empirical study on risk-based authentication applied in the wild. In: Proceedings of the 34th IFIP TC 11 International Conference on ICT Systems Security and Privacy Protection (SEC), IFIP AICT 562, 2019: 134-148.
[16] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 785-794.
[17] Ke G, Meng Q, Finley T, et al. LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems (NIPS), 2017: 3146-3154.
[18] Lopez-Rojas E A, Axelsson S. BankSim: A bank payments simulator for fraud detection research. In: Proceedings of the 26th European Modeling and Simulation Symposium (EMSS), Bordeaux, France, 2014: 144-152.
[19] Lopez-Rojas E A, Elmir A, Axelsson S. PaySim: A financial mobile money simulator for fraud detection. In: Proceedings of the 28th European Modeling and Simulation Symposium (EMSS), Larnaca, Cyprus, 2016: 249-255.
[20] Le Borgne Y A, Siblini W, Lebichot B, et al. Reproducible Machine Learning for Credit Card Fraud Detection - Practical Handbook. Brussels, Belgium: Université Libre de Bruxelles, 2022.
[21] McMahan H B, Ramage D, Talwar K, et al. Learning differentially private recurrent language models. In: Proceedings of the International Conference on Learning Representations (ICLR), 2018.
[22] Mironov I. Rényi differential privacy. In: Proceedings of the IEEE 30th Computer Security Foundations Symposium (CSF), 2017: 263-275.
[23] Mironov I, Talwar K, Zhang L. Rényi differential privacy of the sampled Gaussian mechanism. arXiv:1908.10530, 2019.
[24] Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017: 1175-1191.
[25] Elkan C. The foundations of cost-sensitive learning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), 2001: 973-978.
[26] Zadrozny B, Elkan C. Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002: 694-699.
[27] Xu Z, Kusner M J, Weinberger K Q, et al. Classifier cascades and trees for minimizing feature evaluation cost. Journal of Machine Learning Research, 2014, 15: 2113-2144.