INTERNATIONAL JOURNAL OF BUSINESS AND ECONOMICS INSIGHTS

Volume: 13; **Issue:** 6
Pages: 01– 34
Accepted: 04 May 2023
Published: 02 June 2023

Privacy Leakage Through Repeated AI Explanations in Collaborative Financial Cybersecurity Systems

Maishar Mostari Zenia[1]; Kynatun Mahin[2];

[1]. *Computer Science & Cybersecurity Researcher;*
Bachelor of Science in Computer Science and Engineering;
International University of Business Agriculture and Technology, Dhaka, Bangladesh;
Email: mzenia1s@semo.edu

[2]. *Junior Software Developer, Pirthe, Bangladesh;*
Computer Science & Engineering Department, American International University-Bangladesh, Dhaka, Bangladesh; Email: kynat.mahin@gmail.com

Doi: 10.63125/azdhg571
Received: *09 March 2023;* ***Revised:*** *13 April 2023;* ***Accepted:*** *02 May 2023;* ***Published:*** *02 June 2023;*

Abstract

Banks that share a fraud model and its explanations gain detection power, but each explanation can carry information from other members that the asker was never meant to see. This article measures that leakage on a transparent synthetic consortium of six banks, 24 features, a boosted-tree model with TreeSHAP attributions, and a confidential attribute z (the counterparty is flagged by another member), and it compares three attackers: imputation from own features, the prediction score, and the explanation. Imputation reached an AUC of 0.724 and the score 0.918; one unprotected explanation reached 1.000. Fresh Gaussian noise of 0.4 log-odds lowered one-query AUC to 0.955 but repeated queries with perturbed inputs averaged it away (0.998 at four queries, 1.000 at 16), as a Gaussian theory for the z attribution anticipated (within 0.02 of a direct simulation). Entity-keyed (sticky) noise did not erode (0.957 at every query count) but depends on a trusted entity identifier; per-account query budgets were defeated by colluding accounts (AUC 0.84 rising to 0.975 with six accounts at noise 0.8). Redacting context attributions gave 0.962 and kept own-feature attributions intact, whereas the noise that pushed leakage near the imputation baseline kept only a fifth of own-feature top-3 agreement. Membership inference was modest (AUC 0.63 with score and explanation against 0.58 for the score), while the explanation cut the queries needed to extract the own-feature contribution from 151 to 25. Formal accounting gave epsilon between 3 and 990 for the noise levels studied, loose against measured leakage. In a 2,000-draw Monte Carlo analysis the mean AUC was 0.873 and 48% of draws exceeded 0.90, with noise level and the use of entity-keyed noise dominating the Sobol indices. All figures describe a synthetic testbed with a large, clean attribution for z and are probably upper bounds for rarer attributes. A measurement protocol, release settings and a governance roadmap are proposed.

Keywords

Explanation Privacy, Attribute Inference, Treeshap, Repeated Queries, Differential Privacy Accounting, Collaborative Fraud Detection, Financial Cybersecurity, Model Risk Governance

Introduction

Financial crime does not respect the boundaries of a single bank. A mule account opens at one institution, receives funds from a second and pays out through a third, and the signals that expose it are scattered across all three. Banks, payment operators and supervisors therefore build ways to pool what they know: threat-intelligence exchanges, shared watchlists, joint models, and public-private partnerships that pass information within legal limits (Skopik et al., 2016; Wagner et al., 2019; Financial Action Task Force, 2022). The economics of sharing are mixed. Members gain from a larger view, but each also bears the cost of exposing its own customers, and the incentives to share are fragile (Gal-Or & Ghose, 2005). Recent pilots of collaborative analytics for money-laundering detection and technical work on federated learning for financial graphs show that the modelling is feasible (BIS Innovation Hub, 2023; Suzumura et al., 2019).

When a shared model scores a transaction, the member that asked is usually shown more than a number. Supervisors and model-risk practice expect decisions based on models to be understood and challenged (Board of Governors of the Federal Reserve System & Office of the Comptroller of the Currency, 2011; Monetary Authority of Singapore, 2018), analysts want to see why a payment was flagged, and Shapley-based attributions computed exactly for tree ensembles have become a common way to show it (Lundberg & Lee, 2017; Lundberg et al., 2020). In fraud and anti-money-laundering analytics, explanations are now a normal part of the toolkit (Psychoula et al., 2021; Bussmann et al., 2021; Bracke et al., 2019). In a consortium, though, the explanation is computed from a model that was trained on, and takes as input, information from other members. The attribution for a feature that the asking bank cannot see is, in effect, a message about that feature.

That message can be dangerous. Suppose one input to the shared model is whether the counterparty has been flagged by another member's investigation. That flag is confidential, and in many jurisdictions disclosing that a suspicious activity report has been filed, or is being considered, can amount to tipping off (Financial Action Task Force, 2012). If the explanation of a score includes a contribution for that input, the asker learns the flag by reading the contribution(Asheke & Maksudul, 2022; Asheke & Badrul, 2022). An obvious remedy is to add noise to the explanation, to round it, or to show only the largest attributions. Whether these remedies work depends on something that single-query thinking misses: the asker can ask again (Akter & Sumaiya, 2022; Imran & Jannatul, 2022). A bank can query the same counterparty many times, perturb the inputs it controls, and average away noise that was calibrated for one release. It can also ask from several accounts, or through colluding members, if the limit on queries is per account (Maksudul & Islam, 2022; Morshedul & Islam, 2022). This is the same arithmetic that makes reconstruction possible when many noisy answers are allowed (Dinur & Nissim, 2003; Dwork et al., 2010), applied to an explanation service.

The privacy literature has studied the prediction channel for years: membership inference, model inversion, property inference and model extraction against trained models and their APIs (Shokri et al., 2017; Fredrikson et al., 2015; Tramèr et al., 2016; Ganju et al., 2018; Rigaki & Garcia, 2023; Hu et al., 2022). A smaller body of work asks what explanations add (Shokri et al., 2021; Milli et al., 2019; Aïvodji et al., 2020; Luo et al., 2022; Duddu & Boutet, 2022; Pawelczyk et al., 2023). Two methodological cautions from that work shape this study. First, an attribute inference attack must be compared with a baseline that imputes the attribute from other features, because a great deal of apparent leakage is statistical inference that any observer could make (Jayaraman & Evans, 2022). Second, formal guarantees such as differential privacy are stated for a sensitivity and a number of queries, and they are often far from what an actual attacker achieves (Dwork & Roth, 2014; Dong et al., 2022; Jayaraman & Evans, 2019). What is missing, as far as I could establish, is a controlled comparison of release controls for an explanation service in a consortium setting that includes repeated and coordinated queries, reports the utility cost of each control for the analyst, and checks the results against a formal accounting. I may have missed relevant work, and the literature on this subject is growing quickly.

This article provides that comparison on a synthetic testbed whose every assumption is stated. Four research questions guide it.

RQ1. What does each release channel (the score, the attributions of one query, and repeated queries) reveal about a confidential attribute beyond what imputation from the asker's own features reveals?

RQ2. How do repeated queries, noise that is fresh or keyed to the entity, query budgets and colluding

accounts change that leakage?
RQ3. What does each defence cost in explanation utility, and how do formal privacy accounting and measured leakage compare?
RQ4. What do other leakage types (membership inference and extraction) and differences between member banks add, and what follows for governance and analytics practice?
The contributions are fourfold. A three-attacker protocol separates what the explanation adds from what the score and the asker's own features already give. A study of fresh and entity-keyed noise, redaction, rounding, truncation and budgets quantifies where repeated and coordinated queries defeat each control, with a Gaussian theory that predicts the erosion of fresh noise. A utility and accounting analysis sets the cost to the analyst and the formal bounds beside measured leakage. A Monte Carlo and Sobol analysis over uncertain defence and attacker settings shows which choices matter, and a governance framework turns the findings into records and monitors. The study uses synthetic data and cannot say how large leakage would be in a real consortium; it can show which mechanisms matter and in what order.

Background and Related Work

Collaborative defence in financial crime and cybersecurity

Information sharing has long been advocated for security and financial crime, and the survey literature describes its dimensions: what is shared, with whom, under which trust model and with what legal and economic limits (Skopik et al., 2016; Wagner et al., 2019). In financial crime the international standard-setter asks countries and institutions to share information within and across groups while protecting confidentiality, including the confidentiality of suspicious transaction reporting (Financial Action Task Force, 2012; Financial Action Task Force, 2022). The economic analysis of sharing finds that members weigh the benefit of a pooled view against the competitive and liability costs of exposure (Gal-Or & Ghose, 2005). Pilot work on cross-institution analytics for money laundering, and federated approaches to learning on financial graphs, show that members can contribute without moving raw data (BIS Innovation Hub, 2023; Suzumura et al., 2019; McMahan et al., 2017). The contribution of the other members then enters the model as features such as counterparty risk, device alerts or cross-bank velocity, which is exactly the situation studied here.

Fraud and money-laundering analytics and their explanations

Reviews of fraud detection describe supervised and unsupervised models, imbalance, concept drift and the cost of review (Ngai et al., 2011; West & Bhattacharya, 2016; Abdallah et al., 2016; Bhattacharyya et al., 2011; Dal Pozzolo et al., 2018; Carcillo et al., 2021). Money-laundering detection adds graph methods and sparse labels (Chen et al., 2018; Jensen & Iosifidis, 2023; Weber et al., 2019; Pourhabibi et al., 2020). Gradient-boosted trees remain a strong baseline for tabular transaction data (Ke et al., 2017), and their calibrated probabilities matter when rare classes are resampled (Dal Pozzolo et al., 2015). Explanations are added for analysts and for supervisors. Studies of explainable models in credit and fraud argue that Shapley-based attributions can make models acceptable to supervisors (Bussmann et al., 2021; Bracke et al., 2019; Psychoula et al., 2021), and model-risk guidance in the United States asks for effective challenge of models (Board of Governors of the Federal Reserve System & Office of the Comptroller of the Currency, 2011), as do principles issued by the Monetary Authority of Singapore (Monetary Authority of Singapore, 2018). Little of this literature asks what an explanation reveals to the party that receives it.

Properties and fragility of Shapley-based explanations

TreeSHAP computes exact Shapley values for tree ensembles in polynomial time and has a consistency property that makes the attributions additive in the log-odds of the model (Lundberg & Lee, 2017; Lundberg et al., 2018; Lundberg et al., 2020). Attribution methods differ in axioms and in behaviour, and explanations can be unstable under small perturbations of the input (Sundararajan & Najmi, 2020; Alvarez-Melis & Jaakkola, 2018). They can also be manipulated: an adversary can build models whose explanations hide their dependence on a feature (Slack et al., 2020; Lakkaraju & Bastani, 2020), and black-box attacks on explanation methods in security settings have been demonstrated (Kuppa & Le-Khac, 2020). Whether a Shapley value is an appropriate summary for a particular question is itself debated (Chen et al., 2020; Sundararajan & Najmi, 2020). These properties matter here in two ways. Instability is the reason repeated queries with small perturbations return different attributions, and the

additive structure is the reason an attribution for one feature can be read off the vector.

Privacy attacks on machine learning

Four families of attack are relevant. Membership inference asks whether a record was in the training data; the attack of Shokri and colleagues trains shadow models to mimic the target (Shokri et al., 2017), later work links vulnerability to overfitting and sharpens the attack (Yeom et al., 2018; Salem et al., 2019; Sablayrolles et al., 2019; Song & Mittal, 2021; Carlini et al., 2022; Ye et al., 2022), and label-only variants need no scores (Choquette-Choo et al., 2021). Model inversion and attribute inference recover input features from outputs (Fredrikson et al., 2015; Wu et al., 2016), and property inference recovers properties of the training distribution (Ganju et al., 2018). Model extraction rebuilds a model from queries (Tramèr et al., 2016; Papernot et al., 2017; Jagielski et al., 2020). Collaborative training can leak through the updates and shared outputs (Shokri & Shmatikov, 2015; Bonawitz et al., 2017; Hitaj et al., 2017; Melis et al., 2019; Nasr et al., 2019; Zhu et al., 2019). Surveys organise these attacks and the defences (Rigaki & Garcia, 2023; Liu et al., 2022; Hu et al., 2022). Attribute inference has a long history in other settings, for example from social-network behaviour (Gong & Liu, 2016), and its evaluation has a central difficulty: an attacker that merely imputes a sensitive attribute from correlated public features is not exploiting the model, and a defence cannot be asked to prevent that. Jayaraman and Evans make this point and recommend that attribute inference be evaluated against imputation baselines (Jayaraman & Evans, 2022). This study adopts the recommendation.

Privacy risks of explanations

Explanations add an output channel with its own risks. Gradient-based explanations can leak membership beyond what predictions do (Shokri et al., 2021) and can be used to reconstruct a model from far fewer queries than labels require (Milli et al., 2019). Counterfactual explanations can be used to extract models (Aïvodji et al., 2020) and recourse can create privacy risks of its own (Pawelczyk et al., 2023). Explanations used for model inversion have been shown to improve reconstruction of inputs (Zhao et al., 2021b). For Shapley values specifically, attacks have been shown that infer private features of a party from the attributions served to another (Luo et al., 2022), and sensitive attributes can be inferred from explanations of tabular models (Duddu & Boutet, 2022). Interpretable models can also permit probabilistic reconstruction of their training data (Ferry et al., 2023). Studies in federated learning examine how Shapley values can be interpreted across parties (Wang, 2019). The present study differs in two ways. It includes the baselines that separate the explanation from the score and from imputation, and it adds repeated queries and coordinated accounts, which interact with noise-based defences.

Differential privacy, accounting and private explanations

Differential privacy calibrates noise to sensitivity to bound what any output reveals about one individual or attribute (Dwork et al., 2006; Dwork, 2006; Dwork & Roth, 2014). Guarantees degrade with the number of releases, and the accounting has been tightened by Rényi and zero-concentrated formulations (Mironov, 2017; Bun & Steinke, 2016), analytic calibration of the Gaussian mechanism (Balle & Wang, 2018) and Gaussian differential privacy (Dong et al., 2022). Pufferfish and related frameworks let the secret and the adversary's prior be specified (Kifer & Machanavajjhala, 2014), which suits a single confidential attribute. Practice shows that formal and empirical privacy can differ widely (Jayaraman & Evans, 2019), and recent guidance on evaluating differential-privacy guarantees stresses stating the unit of privacy and the assumptions (Near et al., 2025). Differentially private explanations have been proposed for feature attributions and for interpretable predictions (Patel et al., 2022; Harder et al., 2020), at a cost in explanation quality. A practical question that this literature leaves open is how much noise a given explanation service needs once repeated and coordinated queries are allowed, and what that noise costs the analyst.

Repeated queries, stateful monitors and identity

Averaging over repeated noisy answers is the basis of classical reconstruction results (Dinur & Nissim, 2003), and boosting and composition arguments show how quickly guarantees degrade (Dwork et al., 2010). Defences against model stealing have therefore monitored query streams for patterns (Juuti et al., 2019). Per-account limits are only as strong as the identity system beneath them (Islam & Ahmed, 2023; Sadia & Mosheur, 2023), and multiplying identities defeats limits that are applied per identity (Douceur, 2002). Noise that is deterministic in the entity, so that every repeat receives the same offset,

does not average out, but it requires that queries can be tied to an entity. These considerations motivate the comparison of fresh and entity-keyed noise and of per-account and per-entity budgets.

Explanations, regulation and governance

Law and standards touch explanation from several directions. The General Data Protection Regulation and the debate over a right to explanation concern what individuals can ask of automated decisions (European Parliament and Council, 2016; Wachter et al., 2017; Selbst & Powles, 2017). The EU AI Act sets duties for providers and deployers of AI systems (European Parliament and Council, 2024b), and its application dates and the treatment of fraud-detection systems should be checked against current texts, since the timetable has been the subject of change and amendment proposals that I could not confirm. Operational resilience for financial entities in the European Union is addressed by DORA (Stabak &Shahid, 2022; Feroz & Pervaz, 2022), security of network and information systems by NIS2 (Islam & Md, 2023; Feroz & Shabbir, 2022), and anti-money-laundering rules by the 2024 regulation (European Parliament and Council, 2024c) and the FATF Recommendations (Financial Action Task Force, 2012). Privacy management is addressed by the NIST Privacy Framework (National Institute of Standards and Technology, 2020) and ISO/IEC 27701 (International Organization for Standardization, 2019), and adversarial machine learning, including privacy attacks, is organised in a NIST taxonomy and in the MITRE ATLAS knowledge base (Vassilev et al., 2024; MITRE, n.d.). Model-risk guidance asks for validation and governance of models (Board of Governors of the Federal Reserve System & Office of the Comptroller of the Currency, 2011). These texts have been revised or are being revised at different rates, and the reader should verify the current version and the dates of application before relying on any statement here.

Requirements and the framework

Table 1 organises the literature into eleven strands and the gap each leaves. Six requirements follow (Table 2). R1, know what each channel reveals: a control is only meaningful relative to what the score and the asker's own features already give away, so three attackers are compared. R2, account for repeated and coordinated queries: noise that protects one release can be averaged away, and accounts can be multiplied. R3, preserve the explanation for the analyst: a control that destroys the explanation defeats its purpose, so utility is measured alongside leakage. R4, quantify with bounds and with measurements: formal bounds can be vacuous and measured leakage depends on the attacker, so both are reported. R5, cover other leakage types: membership and extraction risks can move independently of attribute leakage. R6, make assumptions and residual risk governable: regulators and members need records, monitors and a roadmap rather than a single number.

From these the framework (Figure 1) is designed in five layers: the member banks and data layer, the shared model and explanation service, the leakage channels, the release controls, and the governance and assurance layer. Feedback flows from measured leakage to the release controls, and from the query log to the budgets. Figure 2 follows a single query from the asker to the attacker model and marks where each control acts and where a repeat loop returns. The Methods give the testbed, the attackers and the controls.

Table 1: Literature strands, what they establish, representative sources and the gap addressed

Strand	What the literature establishes	Representative sources	Gap addressed here
Privacy attacks on machine learning	Trained models and their prediction APIs leak whether a record was in training, sensitive attributes of a record, parts of the training data and the model itself; surveys organise the attacks	Shokri et al. (2017); Fredrikson et al. (2015); Tramèr et al. (2016); Carlini et al. (2022); Rigaki & Garcia (2023); Hu et al. (2022)	Studied mostly on predictions; the added channel of explanations, and repeated queries in a consortium, are not isolated
Privacy risks of explanations	Gradient, counterfactual and Shapley-based explanations can reveal membership, model parameters and sensitive attributes	Shokri et al. (2021); Milli et al. (2019); Aïvodji et al. (2020); Luo et al. (2022); Duddu & Boutet (2022); Pawelczyk et al. (2023)	Leakage is not compared with what the prediction or an imputation already reveals; effects of repeats on noise defences are not measured
Baselines for attribute inference	Much apparent attribute inference can be reproduced by imputing the attribute from other features, so a baseline is needed	Jayaraman & Evans (2022); Wu et al. (2016); Gong & Liu (2016); Ganju et al. (2018)	Three attackers are compared: imputation, prediction only and explanation
Differential privacy and accounting	Noise calibrated to sensitivity gives formal guarantees that degrade with composition; zero-concentrated and Gaussian accountants tighten the bounds	Dwork et al. (2006); Dwork & Roth (2014); Mironov (2017); Bun & Steinke (2016); Balle & Wang (2018); Dong et al. (2022)	Attribute-level accounting for explanation releases, set against measured leakage
Private explanations	Explanations can be released with differential privacy or from private models, at some cost in quality	Patel et al. (2022); Harder et al. (2020); Kifer & Machanavajjhala (2014)	Utility is measured with faithfulness and agreement, and noise is tested against repeated queries
Repeated queries, reconstruction and abuse	Answering many queries with bounded noise permits reconstruction; stateful monitors and identity multiplication affect query limits	Dinur & Nissim (2003); Dwork et al. (2010); Juuti et al. (2019); Douceur (2002)	Fresh and entity-keyed noise, per-account and per-entity budgets, and collusion are compared
Explanation methods and their properties	Shapley-based attributions, their variants and their instability; explanations can be manipulated	Lundberg & Lee (2017); Lundberg et al. (2020); Sundararajan & Najmi (2020); Chen et al. (2020); Slack et al. (2020); Alvarez-Melis & Jaakkola (2018)	Leakage is treated as a measurable property of an explanation service
Fraud and money-laundering detection with explanations	Fraud models and AML detection use boosted trees and graphs; explanations are expected by model-risk and supervisory practice	Ngai et al. (2011); West & Bhattacharya (2016); Dal Pozzolo et al. (2018); Psychoula et al. (2021); Bussmann et al. (2021); Board of Governors of the Federal Reserve System & Office of the Comptroller of the Currency (2011)	Privacy of the explanation channel is rarely examined
Collaborative detection and information sharing	Banks and operators share intelligence and train models jointly, with economic incentives and legal limits	Skopik et al. (2016); Wagner et al. (2019); Gal-Or & Ghose (2005); Suzumura et al. (2019); BIS Innovation Hub (2023); Financial Action Task Force (2022)	Confidential contributions of other members flow into explanations returned to a member
Privacy in federated and shared learning	Federated training reduces data movement but leaks through updates and shared outputs	McMahan et al. (2017); Bonawitz et al. (2017); Melis et al. (2019); Nasr et al. (2019); Zhu et al. (2019)	Training protections do not cover the explanations served afterwards
Law, standards and governance	Explanation, confidentiality and risk-management duties apply to financial and AI systems	Wachter et al. (2017); Selbst & Powles (2017); European Parliament and Council (2024b); National Institute of Standards and Technology (2020); Vassilev et al. (2024)	Mapping from measured leakage to documented controls

Table 2: Requirements for governing explanation release: rationale, operationalisation in the study and where each is evidenced

Requirement	Why it matters	How it is operationalised here	Evidence in the testbed
R1 Know what each channel reveals	A control is only meaningful relative to what the score and the other features already give away	AUC for inferring z from own features (imputation), from the score, and from the explanation	Table 6, Figure 4
R2 Account for repeated and coordinated queries	Noise that protects one query can be averaged away over many; accounts can be multiplied	Repeated queries with perturbed own features (m up to 16); budgets B and colluding accounts c	Tables 8 and 9, Figure 6
R3 Preserve the explanation for the analyst	A control that destroys the explanation defeats its purpose	Own-feature top-3 agreement, rank correlation, deletion faithfulness and context information retained	Table 7, Figure 7
R4 Quantify with bounds and with measurements	Formal bounds can be vacuous and measured leakage depends on the attacker	Gaussian-mechanism accounting set against measured AUC, with a theory check	Table 11, Figure 11
R5 Cover the other leakage types	Membership and extraction risks can move independently of attribute leakage	Shadow-model membership inference and extraction of the own-feature contribution	Tables 12 and 13, Figure 8
R6 Make assumptions and residual risk governable	Regulators and members need records and monitors, not a single number	Per-bank attackers, Monte Carlo and Sobol analysis, governance roadmap	Tables 10 and 14, Figures 9, 10 and 12

Figure 1: Explanation-release governance framework for a collaborative financial cybersecurity consortium: member banks and data, shared model and explanation service, leakage channels, release controls, and governance and assurance layers, with feedback from measured leakage to controls

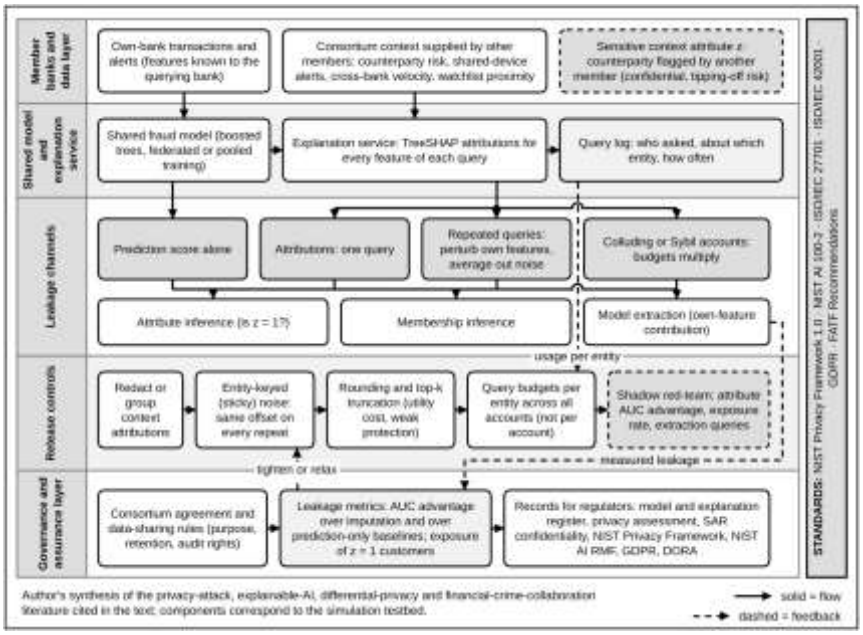
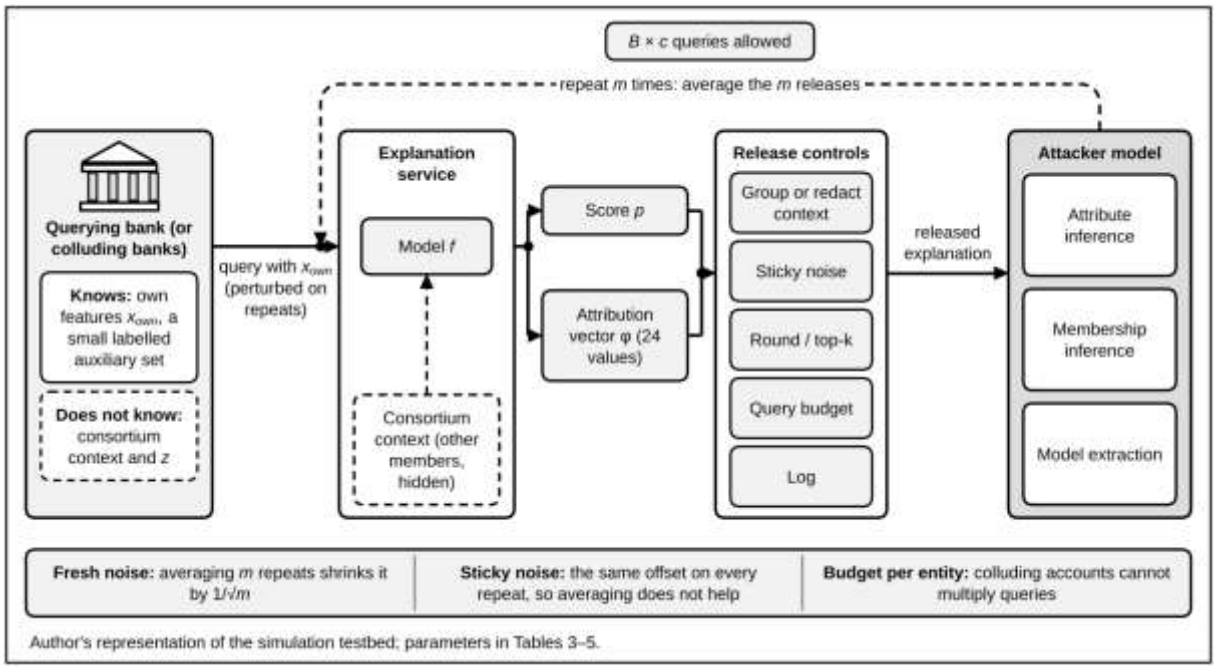


Figure 2: From a query to a leaked attribute: the asking bank, the explanation service with hidden consortium context, the release controls, the attacker model, and the repeat loop that averages m releases

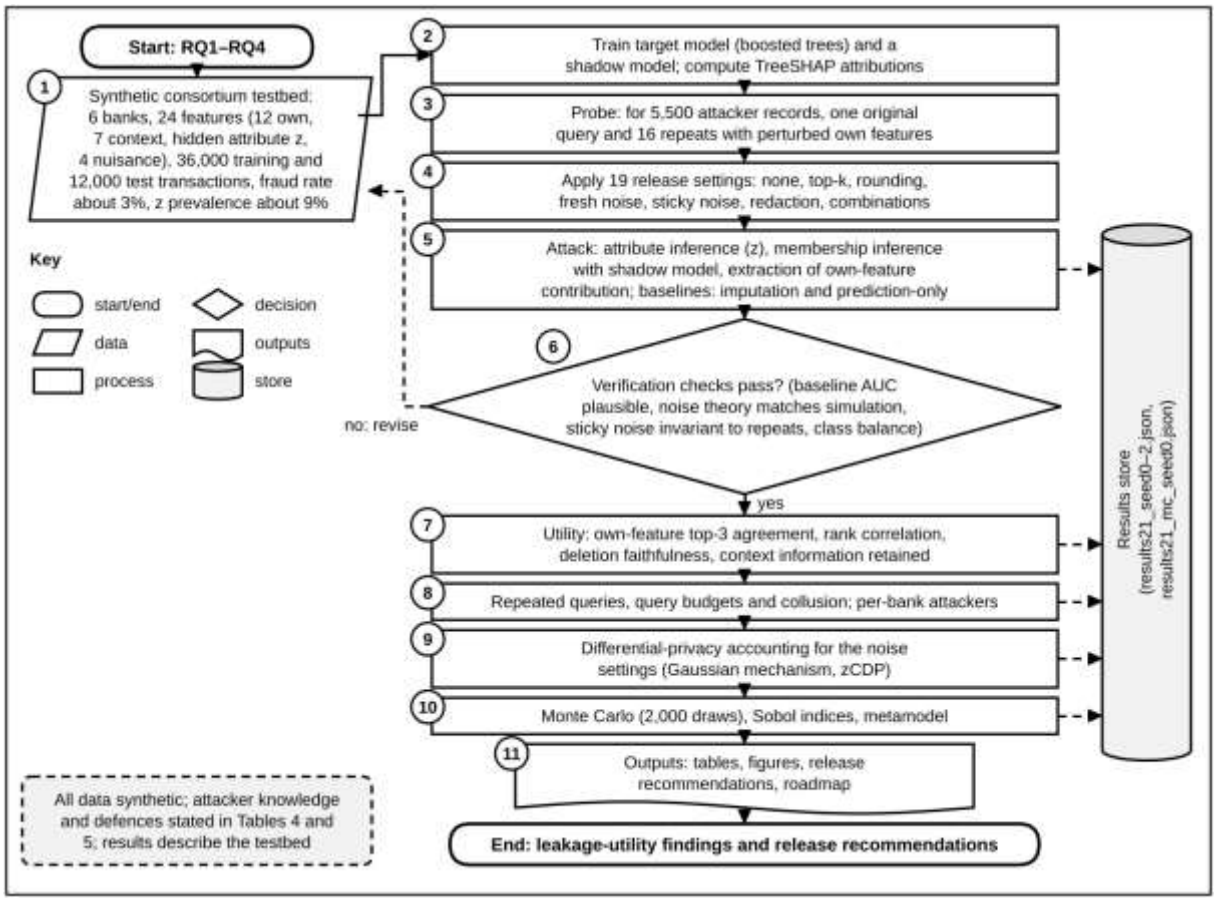


Methods

Study design

The study is a controlled simulation of a consortium of six banks that share one fraud model and its explanations (Figure 3). A synthetic consortium is generated, a target model and a shadow model are trained, and the target's TreeSHAP attributions are computed for a pool of attacker records. For each record the probe consists of one original query and 16 repeats in which the asker's own features are perturbed. Nineteen release settings are then applied to the stored attributions, and three attackers try to infer a confidential attribute, with two further attackers attempting membership inference and extraction. Utility measures, repeated and coordinated queries, per-bank attackers, differential-privacy accounting, and a Monte Carlo and Sobol analysis follow. A verification step checks that the baselines are plausible, that a Gaussian theory matches the simulation, that entity-keyed noise is invariant to repeats and that classes are balanced where required. Every number is computed from three independent seeds of the whole pipeline, except where a table states that one seed was used, and standard deviations over seeds are reported as the measure of simulation variation, not as a sampling interval for any real population.

Figure 3: Study design as a flowchart: synthetic consortium, target and shadow models, probe with repeated queries, release settings, attackers and baselines, verification gate, utility, repeated and coordinated queries, accounting, Monte Carlo and Sobol analysis, and outputs



Every dataset, label and attribute in the study is synthetic. No real transaction, customer, bank or investigation was used. The values in the testbed were chosen to make mechanisms visible and to be plausible in order of magnitude; they were not estimated from any consortium, and I do not claim they match one. Where a statement about practice goes beyond the simulation, it is flagged as inference.

Testbed: consortium, features and the confidential attribute

The consortium has six member banks whose shares of transactions are 26, 22, 18, 14, 12 and 8 per cent. A transaction has 24 features (Figure 4 and Table 3): 12 own-bank features that the querying bank knows, 7 consortium context features supplied by other members (counterparty risk, shared-device alerts, cross-bank velocity, a mule-network score, watchlist proximity, beneficiary age and corridor risk), one binary hidden attribute z, and four nuisance features. Own features are drawn from t distributions with six degrees of freedom and have bank-specific means, so the banks are not identically distributed. The hidden attribute z is that the counterparty is flagged by another member's investigation. Its log-odds depend on a bank-specific intercept and on a linear combination of the own features, so z is correlated with what the asker can see, as a flag would be in practice (a flagged counterparty tends to behave in recognisable ways). The context features shift with z. The prevalence of z is 8.7% overall and varies between 4.6% and 13.2% by bank.

The fraud label follows a logistic model that includes interactions, a threshold effect, context effects and a direct effect of z; the intercept is calibrated so that the fraud rate is about 3%. The target model is a gradient-boosted tree ensemble with 250 trees, 15 leaves and a maximum depth of four (Ke et al., 2017). It is trained on 36,000 transactions and evaluated on 12,000. The AUROC is 0.854 on the test data and 0.943 on the training data; this gap matters later because it is what makes membership inference possible at all. Attributions are exact TreeSHAP values in log-odds (Lundberg et al., 2020). The attribution for z is large and clean: its mean is 0.86 for flagged counterparties and about −0.08 for others,

with standard deviations of 0.15 and 0.013, and its within-record standard deviation across repeats is only 0.012. This is an important design choice and a limitation. A feature that enters the model with a large effect and is not diluted by correlated features produces an attribution that separates the classes almost perfectly. Real attributes may be weaker, and the undefended leakage reported here is therefore probably an upper bound for them.

Figure 4: The testbed: member banks (size, prevalence of z and fraud rate), global importance of the 24 features, the released attribution of z by class, and the three attackers without release control

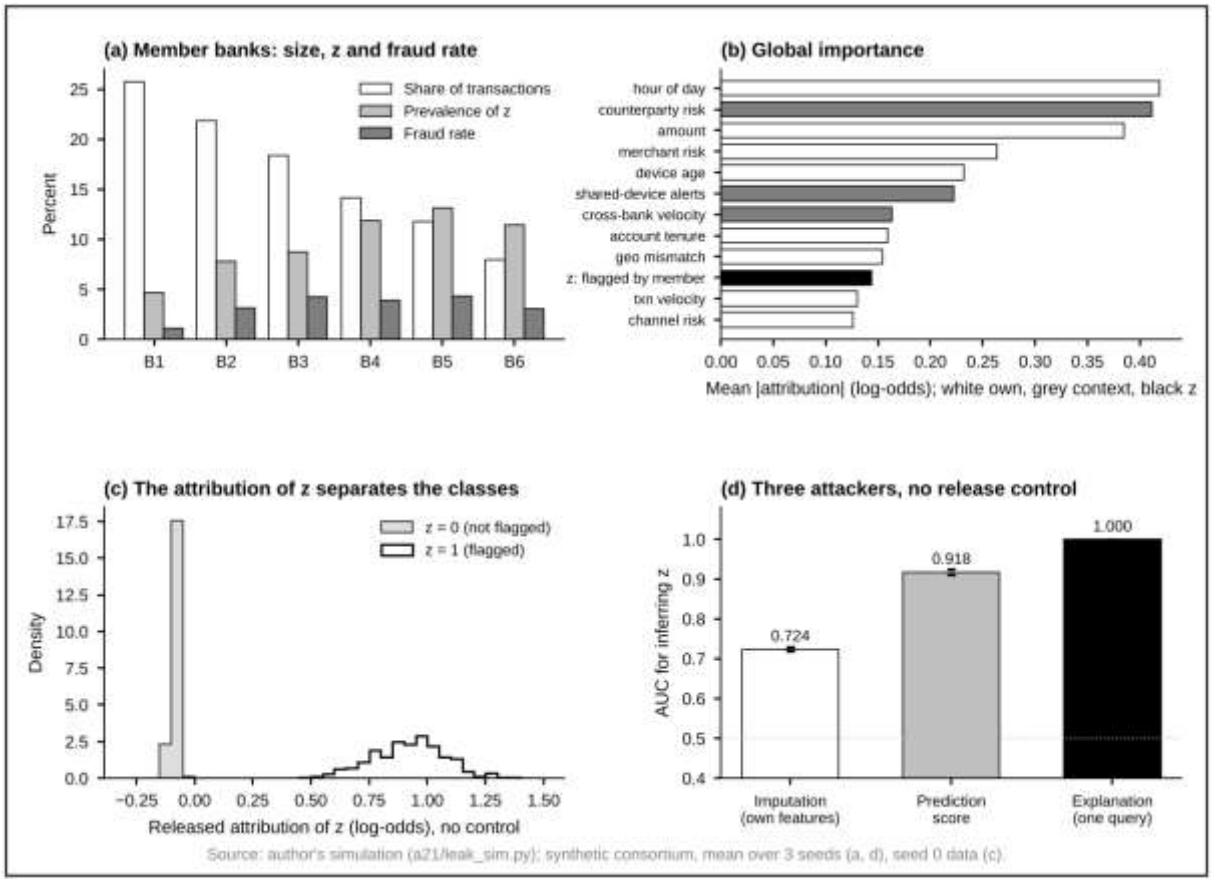


Table 3: Testbed: elements, model, values and basis

Element	Model	Values	Basis
Consortium	Six member banks that share one scoring-and-explanation service; transactions are non-identically distributed across banks	Shares 26, 22, 18, 14, 12, 8 per cent; bank-specific feature means and fraud intercepts	Stylised
Features	24 per transaction: 12 own-bank, 7 consortium context supplied by other members, one hidden binary attribute z, 4 nuisance	Own features are t-distributed (6 degrees of freedom); context features shift with z	Stylised
Hidden attribute z	The counterparty is flagged by another member's investigation; it enters the model and is correlated with own features	Prevalence 8.7% overall, 4.6% to 13.2% by bank	Stylised; confidential in practice
Fraud label	Logistic model with interactions, a threshold effect, context effects and a direct effect of z	Rate 3.1% in test data	Stylised
Target model	Gradient-boosted trees (250 trees, depth 4); TreeSHAP attributions in log-odds	AUROC 0.854 on test and 0.943 on training data	Ke et al. (2017); Lundberg et al. (2020)
Data sizes	36,000 training and 12,000 test transactions; 1,500 attacker auxiliary and 4,000 target records; shadow model on 36,000	Three seeds; mean and standard deviation over seeds	Stylised
Repeated queries	One original query and 16 repeats whose own features are jittered by 0.3 standard deviations; context features fixed	Attacker uses the first m of 17 releases	Assumption
Drift and labels	No drift; the attacker's auxiliary labels for z are correct	Not simulated	Limitation

Attackers and baselines

The asking bank is the attacker. It knows its own features for a query and can obtain the model's score and attributions for that query. It does not know the context features or z. To learn what the attributions mean, it holds an auxiliary set of labelled records (1,500 by default; 50 to 1,500 in a sweep) for which it has observed the released explanations under the same release setting and the true value of z. That is a

strong assumption, since labels for z are exactly what the asker does not have, and it is stated as a limitation. Three attribute-inference attackers are compared, each a regularised logistic regression on standardised features, and each scored by the area under the ROC curve (AUC) on 4,000 held-out target records (Table 4). The imputation attacker uses the own features only. The prediction attacker adds the mean raw score over its queries, and the score is never altered by a release control. The explanation attacker adds the mean of the released 24-dimensional attribution vector over its m queries (and, when top-k truncation is used, the share of attributions that are non-zero). A boosted-tree attacker (120 trees, seven leaves) tests whether a stronger learner changes the findings. A flag is the attacker's posterior of at least 0.5 that the counterparty is flagged, and exposure is reported as the share of flagged counterparties found (recall) and the precision of the flags. Following the guidance on attribute inference (Jayaraman & Evans, 2022), the claim of interest is the gain of the explanation over the imputation and prediction attackers, not the absolute AUC.

Membership inference follows the shadow-model approach (Shokri et al., 2017): a shadow model is trained on 36,000 records from the same process, its members and non-members are used to train a boosted classifier, and the classifier is applied to the target's members and non-members. The features are the score and the loss of the true label for the prediction attacker, statistics of the released attribution (total size, entropy, concentration, the size of the z attribution and the spread over repeats) for the explanation attacker, and both for the combined attacker. The sets are balanced by fraud label (1,000 fraud and 1,000 non-fraud members against equal numbers of non-members), because the fraud class is rare and unbalanced evaluation would mix a change in class mix with a change in leakage. Extraction asks how well an attacker can reconstruct the own-feature contribution of the model (the sum of own-feature attributions) from q queries (25 to 2,000): from scores with a small boosted regression, or from released attributions with per-feature boosted fits or a boosted fit of the summed own-feature attributions. Fidelity is the correlation with the true own-feature contribution on 2,000 held-out records. Extraction of the context part is not attempted because the asker cannot set the context features.

Table 4: Attackers and baselines: goal, knowledge and capability, and output measured

Attack or baseline	Attacker goal	Attacker knows and can do	Output measured
Attribute inference with explanation	Infer whether the counterparty behind a query is flagged by another member (z)	Own features of its transactions; a labelled auxiliary set of 50 to 1,500 records with released explanations under the same controls; up to m repeats with perturbed own features, B queries per account and c colluding accounts	AUC; flagged customers found and precision at a posterior of 0.5
Imputation baseline	Infer z from own features alone	Same auxiliary set; no explanation, no score	AUC
Prediction-only baseline	Infer z from own features and the released score	Same auxiliary set; the score is not altered by any control	AUC
Membership inference	Decide whether a record was in the model's training data	A shadow model trained on data from the same distribution; the label of the record; the score and the explanation (statistics of attribution size, entropy, concentration and spread over repeats)	AUC over 1,000 fraud and 1,000 non-fraud members against equal numbers of non-members
Model extraction	Reconstruct the own-feature contribution of the model	q queries with own features; released scores or released attributions; a boosted regression of the score, or additive fits of the attributions	Correlation with the true own-feature contribution on 2,000 held-out records

Release controls

Seven controls and their combinations are studied (Table 5), applied in this order to the stored attributions: redaction, noise, top-k truncation and rounding. Redaction replaces the eight context attributions, including that of z, by their sum, so that the asker learns the net effect of the context but not its parts. Noise adds independent Gaussian noise of 0.1 to 1.6 log-odds to every attribution. With fresh noise the draw is independent in every release; with sticky noise one offset per entity is drawn and reused in every release for that entity, which is a way of making repeated queries return the same answer. Sticky noise needs an entity identifier that the service trusts, and I treat that as an assumption, not as a property of the method. Top-k truncation keeps the k largest attributions in magnitude (k = 3, 5, 8), and rounding rounds each attribution to a multiple of 0.1, 0.25 or 0.5. Query budgets are modelled as B queries per account, with c colluding accounts pooling B × c queries, capped at 16 queries.

Table 5: Release controls: definition and parameters, the assumption each relies on, and the expected weakness

Release control	Definition and parameters	Assumption it relies on	Expected weakness
Top-k truncation	Keep the k attributions of largest magnitude and set the rest to zero (k = 3, 5, 8)	Sensitive attributions are small	A large attribution of z is kept; which features appear also leaks
Rounding	Round each attribution to a multiple of 0.1, 0.25 or 0.5 log-odds	The signal is fine-grained	A large attribution survives rounding
Fresh noise	Add independent Gaussian noise (standard deviation 0.1 to 1.6) to every attribution in every release	Noise cannot be averaged out	Repeated queries average it away
Sticky noise	Add one Gaussian offset per entity, reused in every release for that entity	Queries are tied to an entity identifier	A fixed offset limits repeats but one release already leaks
Redaction of context attributions	Replace the eight context attributions (including z) by their sum	The analyst needs only the net context effect	The sum and the score still carry z
Query budget	Answer at most B queries per account; c colluding accounts pool B × c queries	One identity per member	Collusion and Sybil accounts multiply the budget
Utility measures	Own-feature top-3 agreement (Jaccard), rank correlation of own and context attributions with the true ones, deletion faithfulness ratio (top-3 removal against random-3 removal)	Analysts act mainly on own-bank features	Does not measure analyst performance

Repeated queries are generated by jittering each own feature with independent Gaussian noise of 0.3 standard deviations, which is the kind of input perturbation that an attacker with control of its own inputs can do. The context features, which belong to other members, stay fixed. The attacker averages the m releases it obtains. Averaging is the simplest attack on noise; stronger estimators (for example, robust or model-based filtering of the noise) are not tried and could do better.

Utility of the released explanation

A control is only useful if it leaves an explanation that an analyst can use. Four measures are computed on 800 test queries, each against the unmodified attributions. Own-feature top-3 agreement is the Jaccard index between the three own-feature attributions of largest magnitude in the released and true explanations. Rank correlation is the mean Spearman correlation between the released and true own-feature attributions, and separately for the context attributions (set to zero under redaction). Deletion faithfulness is the ratio of the mean change in the model score when the three features the released explanation ranks highest are replaced by their training means, to the same change for three randomly chosen own features; it is about 2.05 for the true explanation here and 1 for an explanation no better than chance. These are proxies. They measure agreement with the model's own attributions and not

whether analysts decide better, and studies of explanations in human decision-making find that the two can diverge (Bansal et al., 2021; Buçinca et al., 2021).

Privacy accounting

For the noise settings, privacy is accounted for under the Gaussian mechanism with zero-concentrated differential privacy (Bun & Steinke, 2016; Mironov, 2017). For a release with noise of standard deviation σ on an attribution with sensitivity Δ, the zero-concentrated parameter after m releases is $\rho = m\Delta^2/(2\sigma^2)$, which converts to (ε, δ)-differential privacy with $\varepsilon = \rho + 2\sqrt{(\rho \ln(1/\delta))}$ and $\delta = 10^{-5}$. Two sensitivities are used. The observed one is the separation between the class means of the z attribution (about 0.94); it is not a worst case over all pairs of inputs and it is only meaningful for this model. The conservative one clips the attribution to a magnitude C = 1 and takes Δ = 2C, which is a valid bound for the clipped release (about one per cent of the z attributions exceed the clip). Both treat the attribution of z as the protected quantity. Fresh-noise repeats compose; sticky noise does not compose across repeats on the same entity because the same output is returned, but a single release has the same ε as a fresh release of the same σ. These calculations are intended to show how formal bounds compare with measured leakage and are not a certification of any release setting (Near et al., 2025; Dong et al., 2022).

A Gaussian theory of fresh noise

A simple model predicts how fresh noise erodes under repeats. Let Δμ be the separation of the class means of the z attribution, s the pooled within-class standard deviation of the noiseless attribution, and σ the standard deviation of fresh noise. Averaging m releases leaves noise of standard deviation $\sigma/\sqrt{m}$, so the effective separation is $d' = \Delta\mu/\sqrt{(s^2 + \sigma^2/m)}$ and the AUC of a threshold on the average is approximately $\Phi(d'/\sqrt{2})$, where Φ is the standard normal distribution function. This is the equal-variance Gaussian result for the z attribution alone. The class variances here differ, so the formula is an approximation, and it ignores the own features and other attributions that the full attacker also uses, so I treat it as a check and as a lower bound on the full attacker at high noise, not as an exact law. The observed separation Δμ is 0.94 and s is 0.043.

Monte Carlo, Sobol indices and metamodel

To understand how defence and attacker choices interact, nine settings are varied jointly and uniformly (2,000 draws on seed 0): noise standard deviation (0 to 1.6), the use of entity-keyed noise (a uniform input thresholded at 0.5), queries the attacker wants (1 to 16), budget per account (1 to 16), colluding accounts (1 to 6), top-k (3 to 24, where 24 means none), rounding step (0 to 0.5), redaction (a uniform input thresholded at 0.5) and the attacker's labelled records (50 to 1,500). The effective number of queries is the smaller of the number wanted, the budget times the colluders, and 16. The output is the AUC of the explanation attacker. First-order and total Sobol indices are estimated with a Saltelli design of 512 base samples for nine inputs, a total of 5,632 evaluations (Sobol', 2001; Saltelli et al., 2010; Herman & Usher, 2017). A boosted regression metamodel and a depth-three decision tree are fitted to the draws, and the former is cross-validated. The ranges are stylised, since no consortium data inform them, and the Sobol indices describe sensitivity within them. Monte Carlo and Sobol results use one seed of the simulation, so they show variation over settings, not over simulation replications.

Verification

Five checks were run before the results were read, and the fifth is partial. First, the undefended explanation attacker and the imputation baseline were inspected for plausibility: an AUC of 1.000 for the undefended explanation is a consequence of the design choice discussed above and is reported as such, while the imputation AUC of 0.724 with 1,500 labels falls with fewer labels (0.589 with 50), as expected. Second, the Gaussian theory was compared with a direct simulation of the z attribution under fresh noise at five noise levels and three query counts (the comparison is reported in the Results). Third, sticky noise was verified to be invariant to repeats (the AUC does not move with m beyond seed variation). Fourth, class balance in the membership evaluation was confirmed. Fifth, the target model's AUROC on its members and non-members (0.943 and 0.853) was recorded, since the gap between them bounds what membership inference can exploit; the shadow model's own performance was not logged separately, which is a gap in the verification. An earlier version of the membership features diluted the original query with perturbed repeats and gave AUCs near chance; it was corrected before the final runs by using the original query plus the mean and spread over repeats, and the final numbers reported here use the corrected version only. No check can establish that the testbed resembles a real consortium.

Results

All results come from the synthetic testbed. Values are means over three seeds with standard deviations over seeds where shown; Monte Carlo and Sobol results use one seed. An AUC of 0.5 is chance and 1.0 is perfect separation of flagged from unflagged counterparties.

What each channel reveals

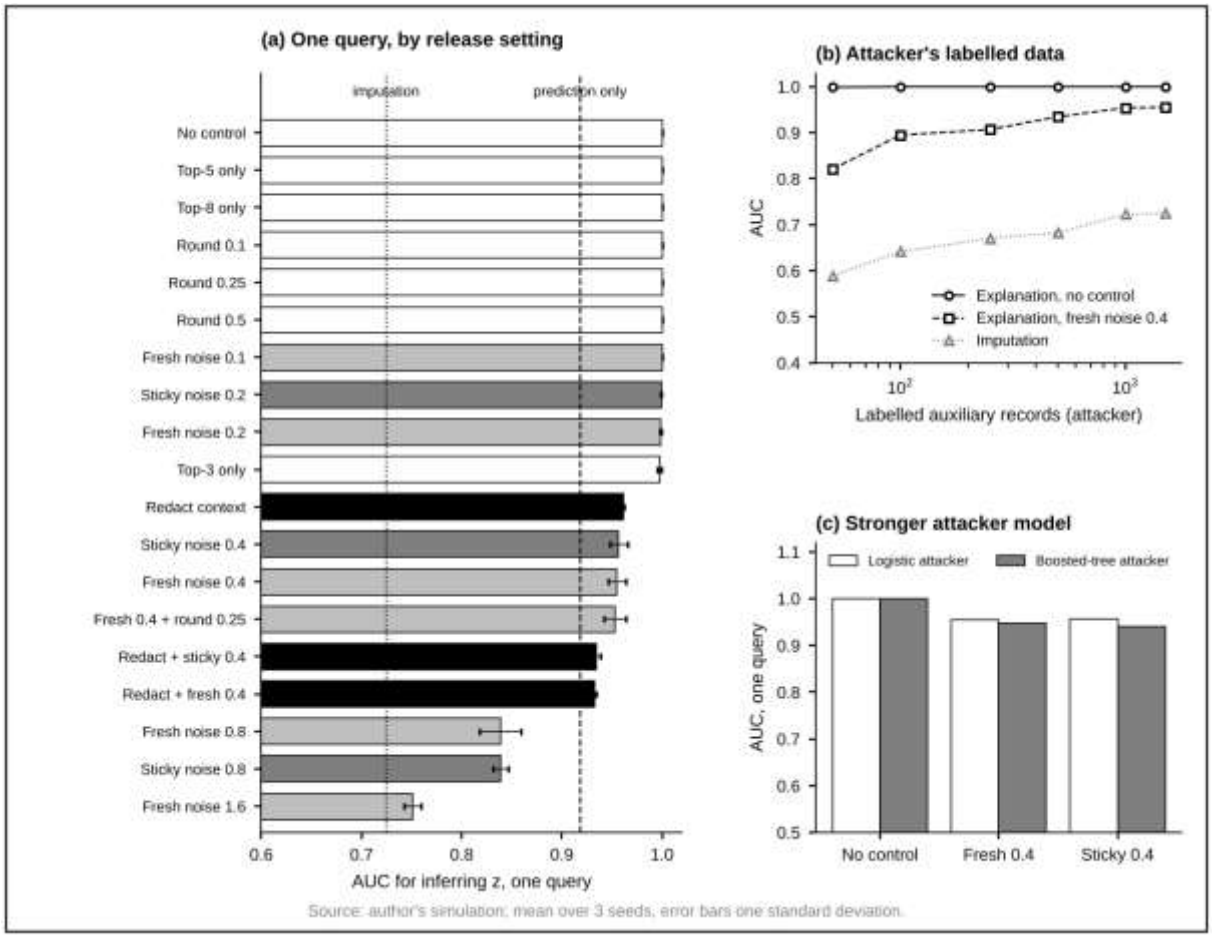
Imputation from the asker's own features gave an AUC of 0.724 with 1,500 labelled records (Table 6). Adding the score raised it to 0.918, so the prediction channel already reveals most of z in this testbed, because z enters the fraud model with a direct effect and the score therefore carries it. One unprotected explanation gave 1.000. The gain of the explanation over the score is thus 0.08 in AUC, and over imputation 0.28. The undefended number should be read with the design choice in mind: the attribution for z has a mean of 0.86 for flagged counterparties against about −0.08 for others, with small spread, so almost any classifier separates them.

Table 6: Attribute-inference AUC by attacker, number of queries m, noise setting and amount of labelled data (mean ± standard deviation over three seeds)

Attacker (AUC for inferring z)	m = 1	m = 4	m = 16	Note
Imputation (own features), 1,500 labelled records	0.724 ± 0.006			Baseline
Prediction score and own features	0.918 ± 0.008			Score not altered by controls
Explanation, no control (logistic attacker)	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	Main attacker
Explanation, fresh noise 0.4 (logistic attacker)	0.955 ± 0.009	0.998 ± 0.001	1.000 ± 0.000	Main attacker
Explanation, sticky noise 0.4 (logistic attacker)	0.957 ± 0.009	0.954 ± 0.006	0.957 ± 0.008	Main attacker
Explanation, no control (boosted-tree attacker)	1.000 ± 0.000		1.000 ± 0.000	Stronger attacker
Explanation, fresh noise 0.4 (boosted-tree attacker)	0.947 ± 0.006		1.000 ± 0.000	Stronger attacker
Explanation, sticky noise 0.4 (boosted-tree attacker)	0.941 ± 0.012		0.948 ± 0.012	Stronger attacker
Imputation, 50 labelled records	0.589 ± 0.073			Labelled data
Imputation, 250 labelled records	0.670 ± 0.029			Labelled data
Imputation, 1,500 labelled records	0.724 ± 0.006			Labelled data
Explanation without control, 50 labelled records	0.999 ± 0.000		0.999 ± 0.000	Labelled data
Explanation without control, 250 labelled records	1.000 ± 0.000		1.000 ± 0.000	Labelled data

The ordering of attackers did not depend on the attacker's learner. A boosted-tree attacker gave 1.000 without control and 0.947 under fresh noise of 0.4 for one query, against 0.955 for the logistic attacker, and it returned to 1.000 at 16 queries (Table 6). The amount of labelled data mattered for the baselines but hardly for the undefended explanation (Figure 5, panel b): imputation fell to 0.589 with 50 labels and 0.670 with 250, whereas the undefended explanation attacker reached 0.999 with 50 labels. In other words, an attacker with a handful of labelled examples loses almost nothing by having few labels when attributions are released without control, a point that matters for the argument that a rare confidential attribute cannot be learned.

Figure 5: Attribute-inference AUC for one query under each release setting, with the imputation and prediction-only baselines; effect of the attacker's labelled data; and a boosted-tree attacker



One query: what each control changes

Truncation and rounding did nothing useful against attribute inference (Table 7, Figure 5, panel a). Keeping only the top five or eight attributions, or rounding to 0.1, 0.25 or 0.5, left the AUC at 1.000; keeping only the top three reduced it to 0.997. The reason is mechanical: the attribution of z is large when z is 1, so it survives both operations, and which features appear in a truncated explanation is itself informative. Fresh noise of 0.1 or 0.2 log-odds left the AUC at 1.000 and 0.999. Noise of 0.4 lowered it to 0.955, at which point the attacker found 58% of flagged counterparties with 84% precision at a posterior of 0.5, and noise of 0.8 lowered it to 0.839 (21% found, 64% precision). Noise of 1.6 gave 0.751, close to the imputation baseline of 0.724. Sticky noise gave the same single-query numbers as fresh noise (0.957 at 0.4, 0.839 at 0.8), as it should, since for one release the two are statistically identical.

Redacting the context attributions into their sum gave 0.962, with 61% of flagged counterparties found. This is above the score-only value of 0.918 because the sum of the context attributions still contains the effect of z, which the score cannot show in isolation. Adding fresh or sticky noise of 0.4 to a redacted release gave 0.933 and 0.934. Combining fresh noise of 0.4 with rounding of 0.25 gave 0.953, no better than noise alone.

A caution applies to every number below 0.918. The score is available to the attacker whatever is done to the explanation, so an AUC below 0.918 does not mean that the explanation protects z: a rational attacker would ignore an uninformative explanation and use the score. The leakage that can be attributed to the explanation channel is the amount by which the explanation attacker exceeds 0.918, and the floor for the whole service is the score unless the score is protected as well. In this testbed noise of 0.8 or more puts the explanation below that floor for a single query; noise of 0.4 and redaction leave it 0.04 above.

Table 7: Release settings for one query: attribute-inference AUC, share of flagged counterparties found and precision, and utility measures (AUC as mean ± standard deviation over three seeds; n/a means exposure was not computed for that setting)

Release setting (one query)	AUC for z	Flagged found (%)	Precision of flags (%)	Top-3 agreement	Rank corr., own	Faith-fulness	Rank corr., context
No control	1.000 ± 0.000	100	100	1.00	1.00	2.05	1.00
Top-3 only	0.997 ± 0.002	n/a	n/a	0.58	0.62	1.99	0.51
Top-5 only	1.000 ± 0.000	100	100	0.91	0.77	2.09	0.70
Top-8 only	1.000 ± 0.000	n/a	n/a	1.00	0.89	2.08	0.84
Rounding 0.1	1.000 ± 0.000	n/a	n/a	0.84	0.96	2.06	0.94
Rounding 0.25	1.000 ± 0.000	100	100	0.72	0.84	1.98	0.79
Rounding 0.5	1.000 ± 0.000	n/a	n/a	0.62	0.66	1.97	0.58
Fresh noise 0.1	1.000 ± 0.000	n/a	n/a	0.68	0.79	2.06	0.74
Fresh noise 0.2	0.999 ± 0.001	n/a	n/a	0.48	0.62	1.94	0.56
Fresh noise 0.4	0.955 ± 0.009	58	84	0.30	0.42	1.69	0.36
Fresh noise 0.8	0.839 ± 0.021	21	64	0.22	0.25	1.37	0.21
Fresh noise 1.6	0.751 ± 0.009	n/a	n/a	0.18	0.13	1.11	0.11
Sticky noise 0.2	0.999 ± 0.001	n/a	n/a	0.48	0.63	1.97	0.57
Sticky noise 0.4	0.957 ± 0.009	61	81	0.31	0.43	1.77	0.38
Sticky noise 0.8	0.839 ± 0.008	19	66	0.21	0.23	1.32	0.21
Redact context attributions	0.962 ± 0.001	61	84	1.00	1.00	2.07	0.00
Redact + fresh noise 0.4	0.933 ± 0.002	49	74	0.31	0.43	1.72	0.00
Redact + sticky noise 0.4	0.934 ± 0.004	51	78	0.30	0.41	1.67	0.00
Fresh noise 0.4 + rounding 0.25	0.953 ± 0.011	n/a	n/a	0.30	0.42	1.70	0.39

Repeated queries

Repeating the query with perturbed own features eroded fresh noise as the Gaussian theory predicts (Table 8, Figure 6, panel a). With noise of 0.4, the AUC rose from 0.955 at one query to 0.988 at two, 0.998 at four and 1.000 at eight. With noise of 0.8 it rose from 0.839 to 0.902, 0.952, 0.989 and 0.998 at 16 queries. Even noise of 1.6, which brought one query close to the baseline, rose to 0.907 at eight queries and 0.955 at 16. The Gaussian theory describes the noise on the z attribution alone, and a direct simulation of that attribution with noise of standard deviation $\sigma/\sqrt{m}$ agrees with it: for noise of 0.4 at one, four and 16 queries the theory gives 0.950, 0.999 and 1.000 against 0.941, 0.999 and 1.000 simulated; for noise of 0.8 it gives 0.796, 0.950 and 0.999 against 0.779, 0.942 and 0.999; for noise of 1.6 it gives 0.661, 0.796 and 0.949 against 0.657, 0.781 and 0.945. The simulated values are lower than predicted by at most 0.017, which is consistent with the class variances being unequal and the formula being an equal-variance approximation. The end-to-end attacker of Table 8 does better than the theory at large noise (for example 0.751 against 0.661 at noise of 1.6 and one query, and 0.831 against 0.796 at four), because it also uses the own features and the other attributions, which carry the imputation signal that the theory ignores. The theory is therefore a good description of what the z attribution alone gives and a lower bound on what a combining attacker achieves. Rounding on top of fresh noise did not help (0.986 at two queries and 0.999 at four, against 0.988 and 0.998 without it).

Entity-keyed noise behaved differently. With sticky noise of 0.4 the AUC was 0.957, 0.955, 0.954, 0.955 and 0.957 at 1, 2, 4, 8 and 16 queries; with sticky noise of 0.8 it was 0.839, 0.843, 0.837, 0.844 and 0.846 (Table 8, Figure 6, panel b). The variation is within the seed variation of about 0.01. Repeating a query for a sticky entity returns the same offset, so there is nothing to average. Redaction alone was constant at 0.962, because it does not depend on the noise, and redaction with fresh noise rose slowly from 0.933

to 0.959 over 16 queries, while redaction with sticky noise stayed between 0.930 and 0.938. The best-performing combinations at 16 queries were therefore those that used sticky noise or redaction, and the fresh-noise settings that looked adequate at one query were no better than no control once the attacker repeated.

Table 8: Attribute-inference AUC by release setting and number of repeated queries, with Gaussian-theory predictions for fresh noise (mean over three seeds; theory rows from the equal-variance approximation, n/a where not computed)

Release setting (AUC for z, explanation attacker)	m = 1	m = 2	m = 4	m = 8	m = 16
No control	1.000	1.000	1.000	1.000	1.000
Fresh noise 0.1	1.000	1.000	1.000	1.000	1.000
Fresh noise 0.2	0.999	1.000	1.000	1.000	1.000
Fresh noise 0.4	0.955	0.988	0.998	1.000	1.000
Fresh noise 0.8	0.839	0.902	0.952	0.989	0.998
Fresh noise 1.6	0.751	0.789	0.831	0.907	0.955
Sticky noise 0.2	0.999	0.998	0.999	0.998	0.998
Sticky noise 0.4	0.957	0.955	0.954	0.955	0.957
Sticky noise 0.8	0.839	0.843	0.837	0.844	0.846
Redact context attributions	0.962	0.962	0.962	0.962	0.962
Redact + fresh noise 0.4	0.933	0.950	0.953	0.958	0.959
Redact + sticky noise 0.4	0.934	0.933	0.930	0.931	0.938
Fresh noise 0.4 + rounding 0.25	0.953	0.986	0.999	1.000	1.000
Theory, fresh noise 0.4	0.950	n/a	0.999	n/a	1.000
Theory, fresh noise 0.8	0.796	n/a	0.950	n/a	0.999
Theory, fresh noise 1.6	0.661	n/a	0.796	n/a	0.949

Budgets and collusion

A query budget per account is a control on m, and its protection depends on how many accounts the attacker can use (Table 9, Figure 6, panels c and d). Under fresh noise of 0.8, a budget of one query per account gave an AUC of 0.840 for a single account, 0.903 for two, 0.934 for three and 0.975 for six colluding accounts. A budget of four with one account gave 0.952, the same as four queries from one account without limit, and six accounts with a budget of two gave 0.997. Under fresh noise of 0.4, a budget of one query gave 0.947 with one account and 0.988 with two, and four queries or more gave 0.999 or higher. In other words, a per-account budget is equivalent to a per-entity budget only if each member has exactly one identity, and the experiment treats colluding accounts as exact pooling of their queries, which is a best case for the attacker because pooled queries are assumed to be free of coordination costs. Sticky noise of 0.4 stayed at about 0.95 in every cell of the budget grid, as it did for the repeat sweep (Table 9).

Figure 6: Repeated and coordinated queries: erosion of fresh noise against theory, entity-keyed noise against repeats, collusion across accounts, and the AUC under budgets per account and colluding accounts

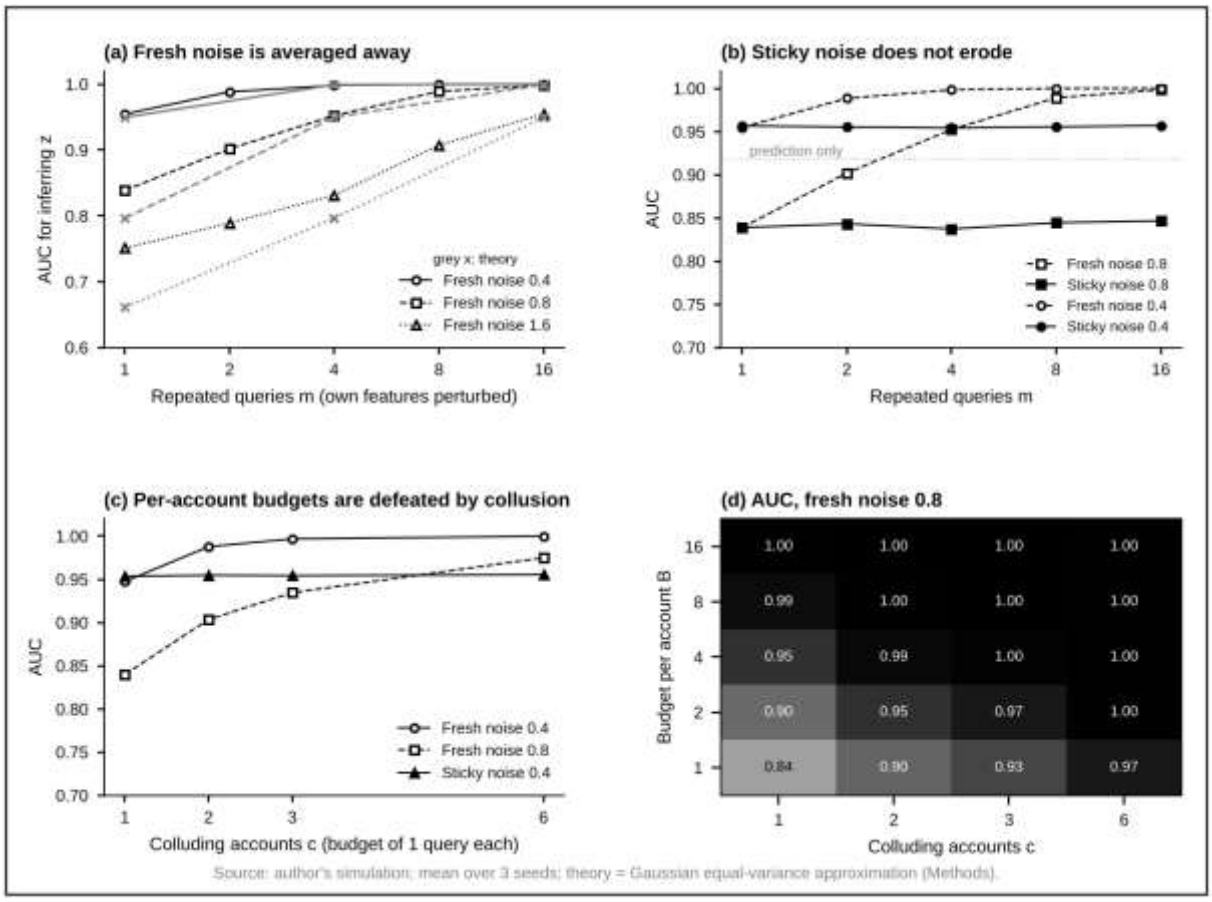


Table 9: Attribute-inference AUC under per-account query budgets B and colluding accounts c (mean over three seeds)

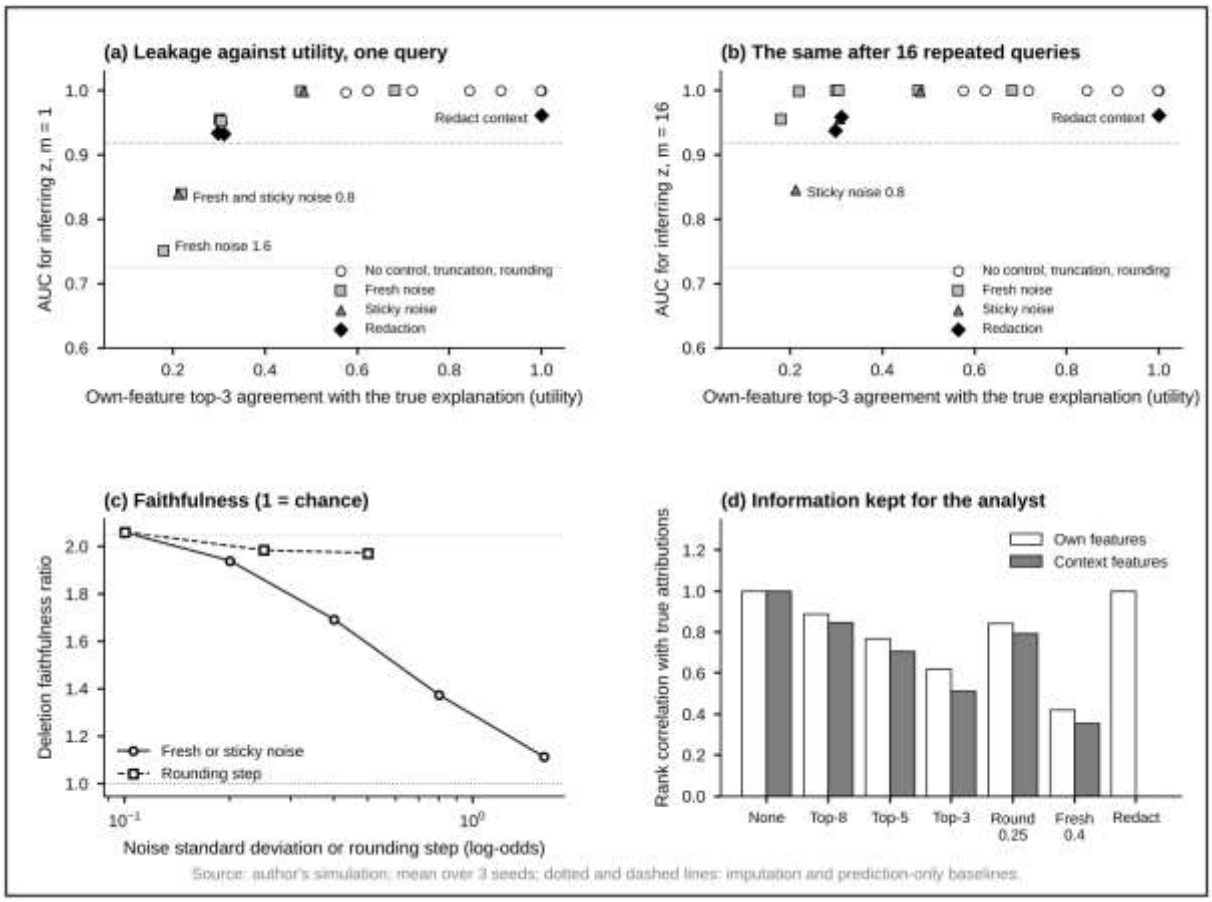
Release setting	Budget per account B	c = 1	c = 2	c = 3	c = 6
Fresh noise 0.4	1	0.947	0.988	0.997	1.000
Fresh noise 0.4	2	0.985	0.999	1.000	1.000
Fresh noise 0.4	4	0.999	1.000	1.000	1.000
Fresh noise 0.4	8	1.000	1.000	1.000	1.000
Fresh noise 0.4	16	1.000	1.000	1.000	1.000
Fresh noise 0.8	1	0.840	0.903	0.934	0.975
Fresh noise 0.8	2	0.898	0.952	0.974	0.997
Fresh noise 0.8	4	0.952	0.990	0.996	0.999
Fresh noise 0.8	8	0.987	0.999	0.999	0.999
Fresh noise 0.8	16	0.999	0.999	0.999	0.999
Sticky noise 0.4	1	0.953	0.955	0.954	0.955
Sticky noise 0.4	2	0.955	0.951	0.957	0.951
Sticky noise 0.4	4	0.949	0.952	0.957	0.950
Sticky noise 0.4	8	0.950	0.955	0.950	0.958
Sticky noise 0.4	16	0.953	0.957	0.956	0.954

What the controls cost the analyst

Controls that leave leakage near the score floor cost much of the explanation (Figure 7). In Table 7, fresh or sticky noise of 0.4 left an own-feature top-3 agreement of 0.30, a rank correlation of 0.42 for own features and 0.36 for context features, and a deletion faithfulness of 1.69 against 2.05 for the true explanation. Noise of 0.8 left 0.22 top-3 agreement and a faithfulness of 1.37, and noise of 1.6 left 0.18 and 1.11, which is close to chance. Rounding and truncation preserved more of the explanation (top-5 retained 0.91 top-3 agreement and 0.77 rank correlation; rounding to 0.25 retained 0.72 and 0.84) and gave no protection. Redaction preserved the own-feature part completely (top-3 agreement 1.00, faithfulness 2.07) and removed the context part (rank correlation 0.00), while lowering the one-query AUC only to 0.962. Redaction combined with noise of 0.4 gave an AUC of 0.933 with the same own-feature utility as noise alone.

The trade-off is therefore not smooth. Rounding, truncation and light noise are cheap and protect nothing here. Strong noise protects a single query and costs most of the explanation. Redaction is cheap for an analyst who needs own-feature attributions and a net context effect, and it protects against reading z directly but not against the net effect (panels a and b of Figure 7, and panel d for the information retained). I could not measure whether an analyst would reach a worse decision with the net context effect than with its parts, and I flag this as the main limitation of the utility results.

Figure 7: Leakage against utility for one query and for 16 repeated queries, deletion faithfulness by noise or rounding step, and rank correlation retained under each control



Member banks

Per-bank attackers, each with only its own bank's labelled records, behaved similarly (Table 10). The imputation AUC ranged from 0.645 at bank 1 (384 labelled records, z prevalence 4.6%) to 0.686 at bank 5 (175 records, prevalence 13.2%), a spread of 0.04 that is of the order of the variation to be expected from sample size, and I do not interpret it. Without control the explanation attacker reached 1.000 at

every bank. Under fresh noise of 0.8 and one query, AUC ranged from 0.723 to 0.787, and with eight queries from 0.959 to 0.979. Under sticky noise of 0.8 with eight queries, it ranged from 0.740 to 0.784. The banks differ in the prevalence of z and in the number of labels, but those differences did not translate into material differences in the protection that noise gave. This result does not rule out larger differences in a real consortium whose members differ more in size, data and customer mix, and a consortium should test its own members separately.

Table 10: Per-bank attackers: share of transactions, prevalence of z, fraud rate, labelled records, and AUC under selected release settings (mean over three seeds)

Attacker	Share (%)	z prevalence (%)	Fraud rate (%)	Aux. records	Imputation AUC	Expl., no control	Fresh 0.8, m = 1	Fresh 0.8, m = 8	Sticky 0.8, m = 8
Bank 1	25.8	4.6	1.1	384	0.645	1.000	0.747	0.968	0.783
Bank 2	21.9	7.8	3.1	317	0.648	1.000	0.787	0.977	0.740
Bank 3	18.4	8.7	4.2	270	0.671	1.000	0.744	0.979	0.772
Bank 4	14.1	11.9	3.9	225	0.677	1.000	0.752	0.973	0.784
Bank 5	11.8	13.2	4.3	175	0.686	1.000	0.723	0.966	0.771
Bank 6	8.0	11.5	3.1	130	0.666	1.000	0.742	0.959	0.760

Formal accounting against measured leakage

Gaussian accounting under zero-concentrated differential privacy gave epsilon values that depended strongly on sensitivity and on the number of releases (Table 11, $\delta = 10^{-5}$). With the observed sensitivity of 0.94, noise of 0.8 gave $\varepsilon = 6.3$ for one release, 14.0 for four and 33.6 for 16; with sensitivity 2 under clipping, the same noise gave 15.1, 36.5 and 98.0. Noise of 0.4 gave ε of 14.0, 33.6 and 89.3 with the observed sensitivity, and noise of 0.2 gave 33.6, 89.3 and 267.1, up to 991.9 at 16 releases under the conservative sensitivity. These are very large values for a guarantee. Bringing ε down to 1.2 for one release took noise of 4 with the observed sensitivity, and 8 with the conservative sensitivity, and the corresponding measured AUC was not computed because noise of that size removes the explanation entirely. The accounting is therefore loose against the measured leakage: noise of 0.8 gave a measured one-query AUC of 0.839 (below the score floor), while the accounting still reported ε above 6 for one release. This is expected for a worst-case bound applied to a particular attribute, and it does not mean the bound is wrong, only that it is not informative here about the risk an attacker poses. Conversely, the accounting for 16 releases at noise of 0.8 ($\varepsilon = 33.6$) describes a weak guarantee, which agrees in direction with the AUC of 0.998 that the attacker reached.

Table 11: Gaussian-mechanism accounting (zero-concentrated differential privacy, $\delta = 10^{-5}$) for the noise settings, with observed and conservative sensitivity, and measured AUC (n/a where not computed)

Noise sd (log-odds)	ε, m = 1 (Δ obs.)	ε, m = 4 (Δ obs.)	ε, m = 16 (Δ obs.)	ε, m = 1 (Δ = 2)	ε, m = 4 (Δ = 2)	ε, m = 16 (Δ = 2)	Measured AUC, m = 1	Measured AUC, m = 16
0.2	33.6	89.3	267.1	98.0	296.0	991.9	0.999	1.000
0.4	14.0	33.6	89.3	36.5	98.0	296.0	0.955	1.000
0.8	6.3	14.0	33.6	15.1	36.5	98.0	0.839	0.998
1.6	3.0	6.3	14.0	6.8	15.1	36.5	0.751	0.955
4	1.2	2.4	5.0	2.5	5.3	11.6	n/a	n/a
8	0.6	1.2	2.4	1.2	2.5	5.3	n/a	n/a

Membership inference and extraction

Membership inference was weak to moderate (Table 12, Figure 8, panels a and b). Using the score and the loss only, the attacker's AUC was 0.584. The explanation only gave 0.542 without control. Both together gave 0.633 for one query and 0.641 for 16. Fresh noise of 0.4 lowered the explanation-only AUC to chance (0.501) and the combined AUC to 0.598 at one query, but at 16 queries the combined AUC returned to 0.631. Restricted to fraud records, which are more vulnerable because the model fits them more closely, the combined AUC was 0.730 without control, 0.665 under noise of 0.4, 0.682 under redaction and 0.717 under top-five truncation. The gap in the target's AUROC between members (0.943) and non-members (0.853) is what the attacker exploits, and a model that generalised better would leak less through this channel. Membership inference in this testbed is thus a smaller risk than attribute inference, and the explanation adds about 0.05 to the score-only attacker in AUC.

Extraction was the clearest case in which the explanation changed the economics of the attack (Table 13, Figure 8, panels c and d). With 25 queries, a score-only surrogate reached a fidelity of 0.57 with the true own-feature contribution, whereas additive fits to released attributions reached 0.88. The number of queries needed to reach a fidelity of 0.8 was 151 from scores and 25 from unprotected attributions; it was also 25 under rounding of 0.25 and under top-five truncation, 84 under fresh noise of 0.4 and 1,388 under fresh noise of 1.6. With 2,000 queries every setting except noise of 1.6 reached 0.92 to 0.95 and noise of 1.6 reached 0.84. Explanations therefore make extraction of the own-feature part cheaper by a factor of six in this testbed, and noise delays but does not prevent it. This is consistent with earlier results for explanation-based reconstruction of models (Milli et al., 2019; Aïvodji et al., 2020), though the quantities are not comparable.

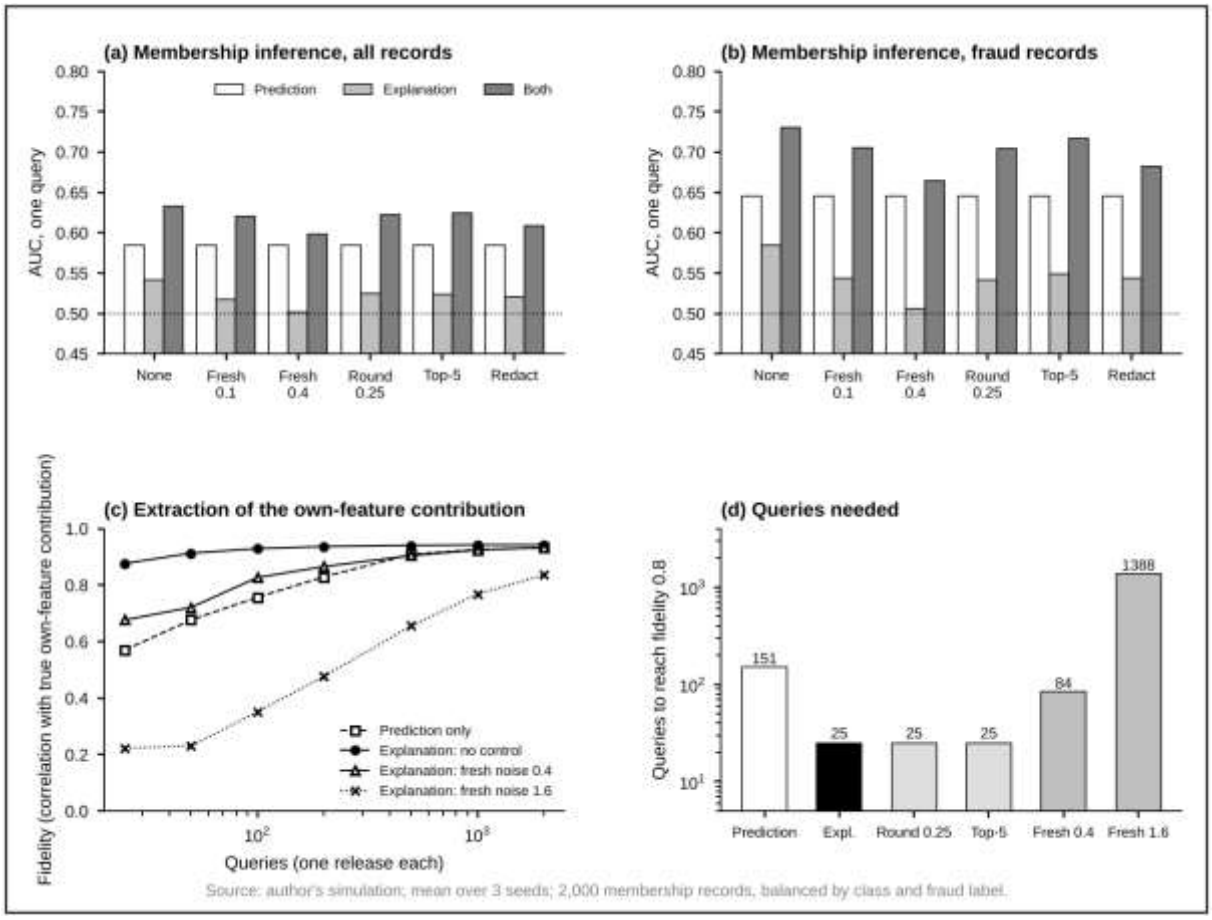
Table 12: Membership-inference AUC by release setting for score only, explanation only and both (mean ± standard deviation over three seeds; target AUROC on members and non-members shown)

Release setting (membership AUC)	Score only, m = 1	Explanation only, m = 1	Both, m = 1	Both, m = 16	Both, fraud records, m = 1
No control	0.584 ± 0.005	0.542 ± 0.003	0.633 ± 0.002	0.641 ± 0.002	0.730 ± 0.006
Fresh noise 0.1	0.584 ± 0.005	0.517 ± 0.001	0.620 ± 0.003	0.634 ± 0.004	0.705 ± 0.007
Fresh noise 0.4	0.584 ± 0.005	0.501 ± 0.009	0.598 ± 0.008	0.631 ± 0.005	0.665 ± 0.012
Rounding 0.25	0.584 ± 0.005	0.525 ± 0.005	0.622 ± 0.006	0.631 ± 0.007	0.705 ± 0.013
Top-5 only	0.584 ± 0.005	0.523 ± 0.015	0.624 ± 0.001	0.639 ± 0.006	0.717 ± 0.009
Redact context attributions	0.584 ± 0.005	0.521 ± 0.014	0.608 ± 0.004	0.628 ± 0.004	0.682 ± 0.006
Target model AUROC: members; non-members	0.943	0.853			

Table 13: Extraction of the own-feature contribution: fidelity (correlation with the true contribution) by number of queries and release setting (mean ± standard deviation over three seeds)

Queries q	Score only	Expl., additive	Expl., summed	Rounding 0.25	Top-5	Fresh noise 0.4	Fresh noise 1.6
25	0.57 ± 0.04	0.88 ± 0.02	0.72 ± 0.01	0.87 ± 0.02	0.85 ± 0.01	0.68 ± 0.04	0.22 ± 0.07
50	0.68 ± 0.04	0.91 ± 0.01	0.84 ± 0.01	0.90 ± 0.01	0.89 ± 0.00	0.72 ± 0.01	0.23 ± 0.05
100	0.76 ± 0.04	0.93 ± 0.00	0.87 ± 0.01	0.92 ± 0.00	0.91 ± 0.00	0.83 ± 0.01	0.35 ± 0.04
200	0.83 ± 0.02	0.94 ± 0.00	0.91 ± 0.00	0.93 ± 0.00	0.92 ± 0.00	0.87 ± 0.01	0.48 ± 0.03
500	0.91 ± 0.01	0.94 ± 0.00	0.93 ± 0.00	0.93 ± 0.00	0.92 ± 0.00	0.90 ± 0.00	0.66 ± 0.02
1,000	0.93 ± 0.00	0.94 ± 0.00	0.94 ± 0.00	0.94 ± 0.00	0.92 ± 0.00	0.92 ± 0.00	0.77 ± 0.02
2,000	0.93 ± 0.00	0.94 ± 0.00	0.95 ± 0.00	0.94 ± 0.00	0.92 ± 0.00	0.93 ± 0.00	0.84 ± 0.01

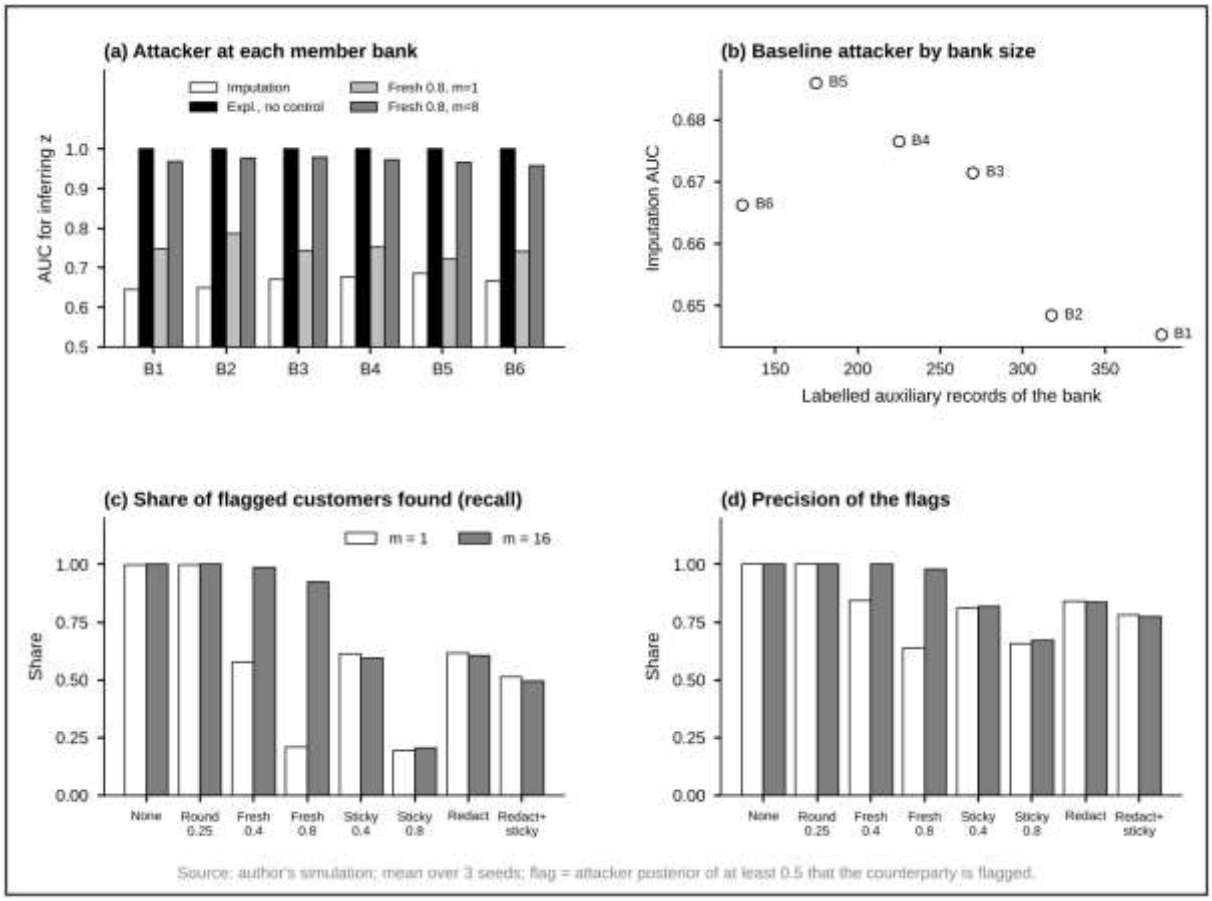
Figure 8: Membership inference by release setting for all records and fraud records, extraction fidelity by number of queries, and queries needed to reach a fidelity of 0.8



Exposure of flagged customers

An AUC does not say how many flagged counterparties are identified. At a posterior of 0.5, an attacker without control found essentially all flagged counterparties with precision 1.0 (Figure 9, panels c and d). Under fresh noise of 0.4 it found 58% of them with precision 0.84 at one query, and 98.8% with precision near 1 at 16. Under fresh noise of 0.8 it found 21% at one query and 93% at 16. Under sticky noise of 0.4 it found 61% at one query and 59% at 16, and under sticky noise of 0.8 it found 19% and 20%. Redaction alone found 62% and 61%, and redaction with sticky noise found 51% and 50% at one and 16 queries. In practical terms, even the best settings here still disclosed the flag of about half of the flagged counterparties to the attacker with a precision of 0.78, since the attacker in this testbed starts with a score that already carries the signal. Tipping-off concerns do not require perfect inference; an attacker that is right four times out of five about a customer being under investigation may already have learned something that the confidentiality rules were designed to withhold, though whether that is so in a given legal setting is a question for counsel.

Figure 9: Per-bank attackers under selected release settings, baseline attacker by bank size, share of flagged customers found and precision of the flags after one and 16 queries



Monte Carlo and Sobol analysis

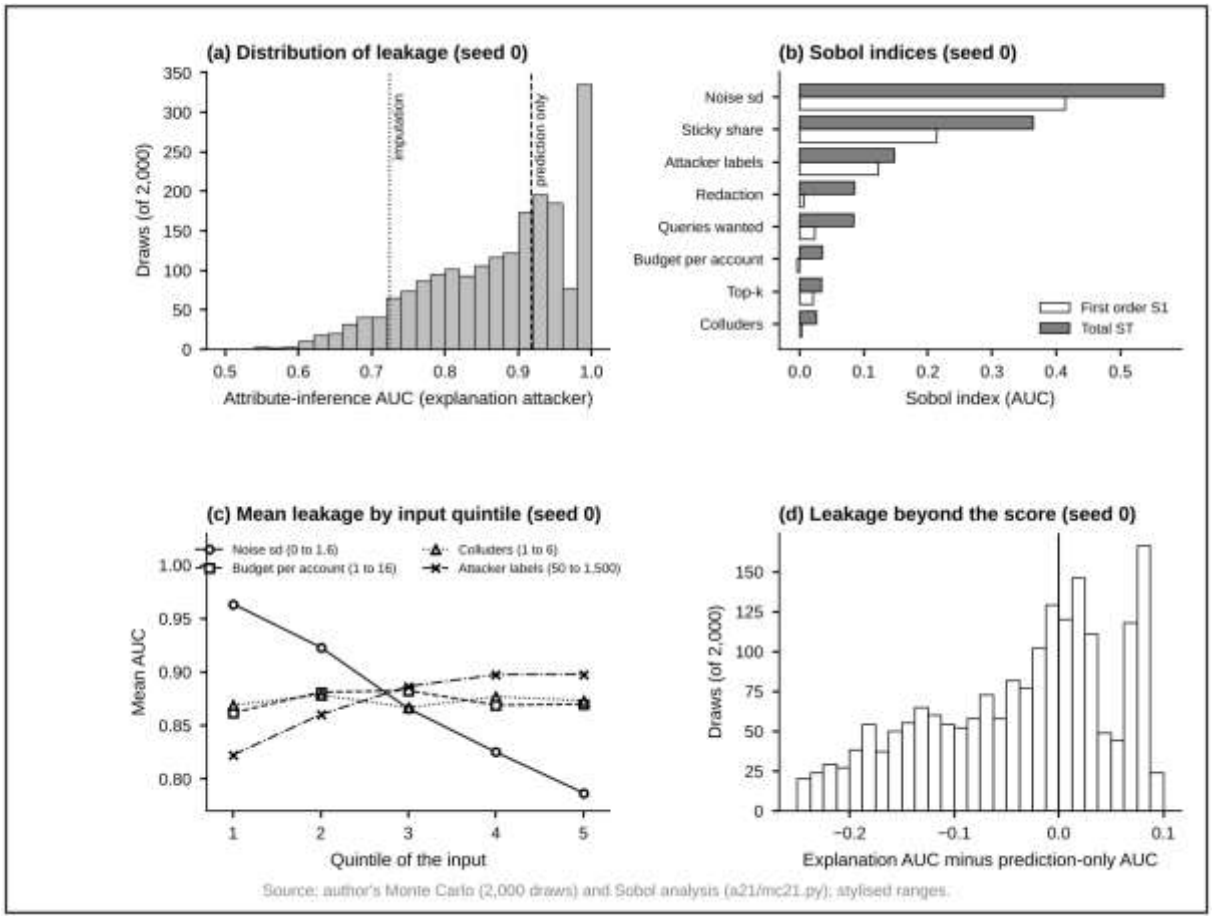
Across 2,000 draws of nine uncertain settings, the explanation attacker's AUC had a mean of 0.873 with a 5th to 95th percentile range of 0.686 to 1.000 (Table 14, Figure 10). In 48% of draws the AUC exceeded 0.90 and in 24% it exceeded 0.95. On average the explanation attacker scored 0.169 above the imputation attacker fitted on the same number of records, but 0.039 below the prediction-only attacker (5th, 50th and 95th percentiles of the difference −0.217, −0.018 and +0.083); the explanation channel added to the score only in the draws with little noise, or with noise that the attacker could average away, which is the upper tail of the difference.

Noise level dominated. Its first-order and total Sobol indices were 0.41 and 0.57, followed by the use of entity-keyed noise (0.21 and 0.36), the attacker's number of labelled records (0.12 and 0.15), redaction (0.01 and 0.09) and the number of queries wanted (0.02 and 0.08). Budget per account, colluders, top-k and rounding had total indices at or below 0.04. The first-order indices sum to about 0.82, so interactions account for roughly a fifth of the variance. These indices concern variation over the stated ranges, which are stylised, and they should not be read as a statement of importance in a real deployment. They also hide an interaction that Tables 8 and 9 show directly: budgets and collusion matter only when the noise is fresh and when the budget binds. A boosted-tree metamodel reproduced the draws well (cross-validated R^2 of 0.95); a depth-three tree explained 0.75 of the variance and split first on noise of about 0.72, then on redaction, the use of sticky noise and the attacker's labels. The mean AUC fell from 0.964 in the lowest quintile of noise to 0.787 in the highest, and rose from 0.82 to 0.90 across the quintiles of the attacker's labelled records (Figure 10, panel c).

Table 14: Monte Carlo and Sobol analysis of attribute-inference AUC under nine uncertain defence and attacker settings (2,000 draws, one seed)

Quantity	Value	Detail
Mean AUC of the explanation attacker	0.873	5th to 95th percentile 0.686 to 1.000
Draws with AUC above 0.90; above 0.95	48%; 24%	Of 2,000 draws over nine uncertain settings
Explanation AUC minus prediction-only AUC	mean -0.039	5th, 50th, 95th percentile -0.217, -0.018, 0.083
Explanation AUC minus imputation AUC	mean 0.169	Imputation fitted on the same number of labelled records
Sobol indices, first order S1 and total ST	noise sd 0.41 and 0.57; sticky share 0.21 and 0.36; attacker labels 0.12 and 0.15; redaction 0.01 and 0.09; queries wanted 0.02 and 0.08	5,632 evaluations on seed 0
Metamodel: cross-validated R^2 (depth-3 tree R^2)	0.95 (0.75)	Boosted regression and a depth-3 tree on the 2,000 draws

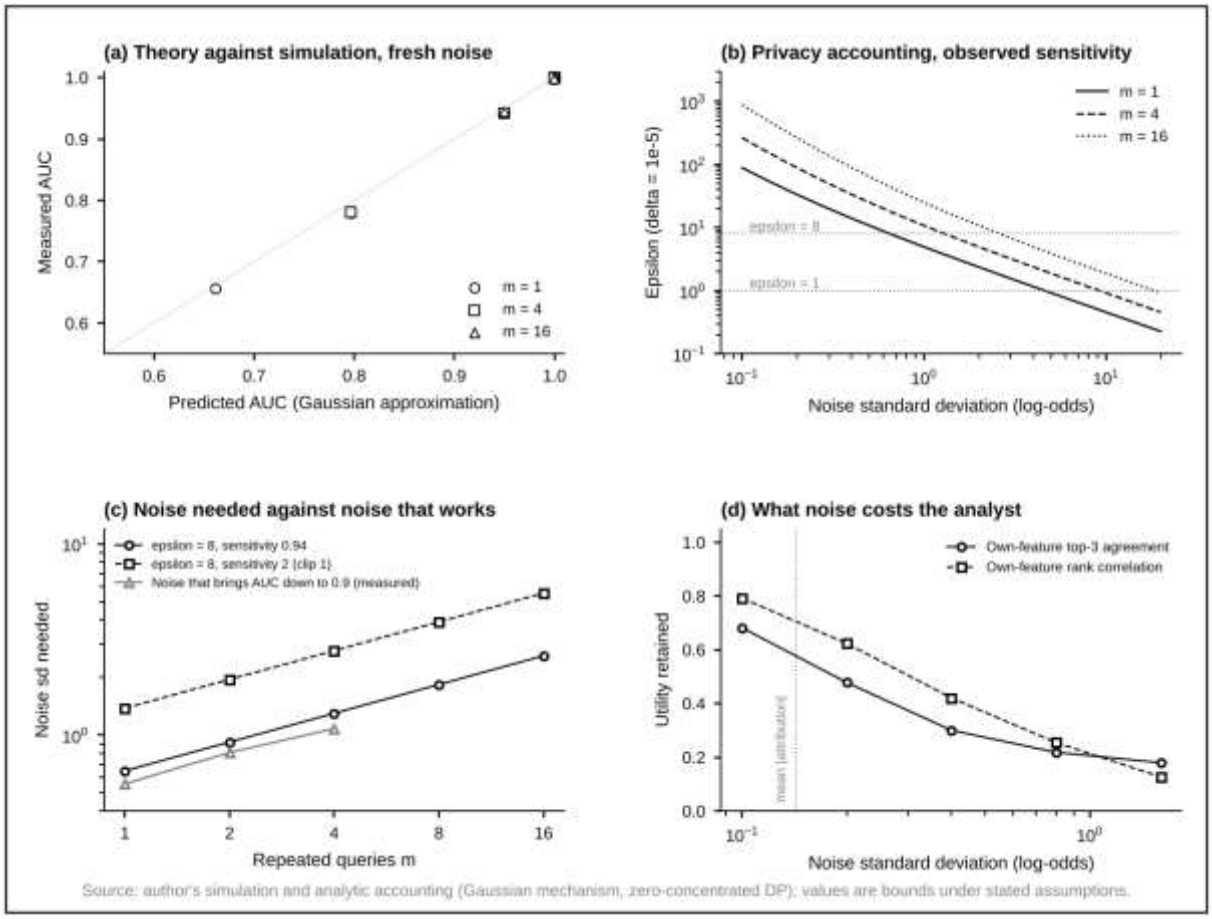
Figure 10: Monte Carlo analysis: distribution of the explanation attacker's AUC, Sobol indices, mean AUC by input quintile, and the AUC of the explanation attacker minus that of the prediction-only attacker



Theory, accounting and utility together

Figure 11 puts four results side by side. The AUC of the z attribution under fresh noise, simulated directly, lies close to the AUC predicted by the Gaussian approximation across noise levels and query counts (panel a). The epsilon implied by Gaussian accounting falls with noise and rises with the number of queries (panel b). The noise needed for ε = 8 grows with the square root of the number of queries: about 0.65 for one query and 2.6 for 16 with the observed sensitivity, and about 1.4 and 5.5 with the conservative one (panel c). By comparison, the noise that brought the measured AUC down to 0.9 was about 0.55 for one query, 0.8 for two and 1.1 for four, and was not reached within the grid for eight or 16 queries. The two criteria are different, and the agreement of order of magnitude between them is not a validation of either. Finally, the utility retained by the analyst falls steeply over the same range: own-feature top-3 agreement was 0.68 at noise of 0.1 and 0.30 at 0.4 and 0.18 at 1.6, and rank correlation 0.79, 0.42 and 0.13 (panel d). The noise level at which the formal bound becomes small is therefore far above the level at which the explanation stops being useful.

Figure 11: Gaussian theory against direct simulation of the z attribution under fresh noise, privacy accounting by noise level and query count, noise needed against noise that works, and utility retained under noise



Discussion

Answers to the research questions

RQ1. The three channels differ sharply. Imputation from own features recovered the confidential attribute with an AUC of 0.724 and the score with 0.918, and a single unprotected explanation reached 1.000. The explanation therefore adds measurable leakage beyond what is already visible, but the larger part of the disclosure in this testbed comes from the score, because z affects the fraud label directly. Release controls for explanations cannot lower leakage below the score floor unless the score is also controlled, and in a real service the score is typically the product being sold.

RQ2. Repeated queries defeat fresh noise, and the Gaussian theory describes how quickly: noise of 0.4

that gives 0.955 for one query gives 0.998 for four. Entity-keyed noise did not erode, at the cost of a dependence on a trusted identifier, and budgets per account were defeated by collusion, since pooled queries behave like queries from one account. Redaction of context attributions reduced leakage for every query count (0.962) and left own-feature attributions untouched, but it left most flagged customers identifiable.

RQ3. The defences that protected one query cost much of the explanation: noise of 0.4 left 30% top-3 agreement and noise of 0.8 left 22%. Rounding and truncation preserved the explanation and protected nothing against attribute inference. Formal accounting gave epsilon values between 3 and 990 for the noise levels studied (0.2 to 1.6) and was loose against measured leakage, and the noise needed for a modest ε grew with the square root of the number of queries.

RQ4. Membership inference added about 0.05 in AUC to the score, and more for fraud records (0.730 against 0.646 for the score alone, from Table 12 and Figure 8). Extraction of the own-feature contribution needed six times fewer queries from unprotected explanations than from scores (25 against 151). The member banks did not differ materially, within the noise of the simulation. These results support governance that treats explanation release as a disclosure channel with its own measures, limits and records.

Why the first intuition about noise is wrong

The intuition that noise protects fails for a structural reason. A noisy release of a fixed quantity gives the attacker one draw of the noise. If the attacker can ask again for a quantity that is unchanged, it gets another independent draw, and the average of m draws has standard deviation $\sigma/\sqrt{m}$. The attacker's perturbation of its own features, which the service has no reason to forbid, leaves the context part of the explanation almost unchanged (within-record standard deviation of the z attribution of 0.012) and so supplies the repeated draws that the averaging needs. Sticky noise attacks the structure directly by removing the independence, and the experiments show that it works, but it shifts the problem to identity: it protects only if the service can tie the repeats to the same entity, and an attacker that can vary the entity (for example, by changing an identifier or by querying through a counterparty that the service treats as different) receives a fresh offset. Whether this can be done in a real consortium depends on identity resolution and on the integrity of the entity key, and I have not tested it. Redaction attacks another part of the structure: it removes the channel by which z is separable. It does not remove z from the net effect of the context or from the score.

The score floor and what it implies

A result that deserves emphasis is that explanation controls cannot do better than the score. The prediction attacker's AUC of 0.918 is a floor for any control applied to the explanation only. A consortium that wants to protect z from the asker has to consider whether the score itself should be computed without z, or with z in a form that cannot be decoded, which changes the model and its accuracy. That is a modelling decision with a business cost and it cannot be settled by the release layer. In practice, the confidential attribute can be removed from the information the asker sees by designing the shared model so that other members' confidential contributions enter only through aggregates, or through a score that is itself bounded in its sensitivity, though each choice reduces detection power, and I have not measured by how much.

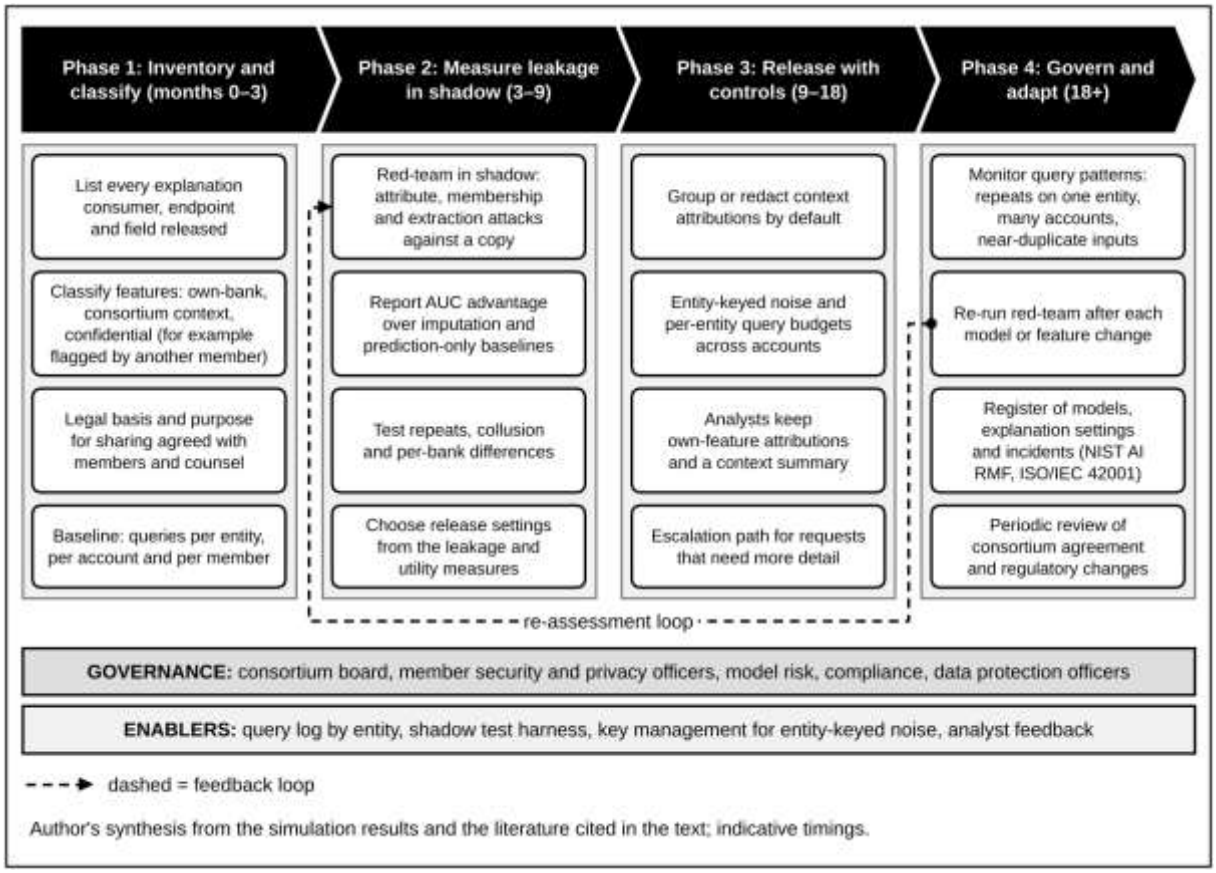
Governance

Explanations are required or expected in many financial settings (European Parliament and Council, 2016; Wachter et al., 2017; Selbst & Powles, 2017; Board of Governors of the Federal Reserve System & Office of the Comptroller of the Currency, 2011; Monetary Authority of Singapore, 2018), and confidentiality rules limit what may be revealed about suspicious activity reporting (Financial Action Task Force, 2012; European Parliament and Council, 2024c). The two pressures meet in the explanation service. The findings suggest that the explanation service of a consortium should be governed as a disclosure channel: the features that other members contribute should be classified, the attributions of confidential features should be withheld or grouped by default, and any noise defence should be justified against repeated and coordinated queries. Figure 12 sets out a four-phase roadmap: inventory and classification, shadow measurement of leakage, release with controls, and continuing governance with re-assessment. The records should include a register of models and explanation settings, the red-team results with their baselines, and the query monitoring that supports entity-level budgets, in line

with risk-management frameworks for privacy and AI (National Institute of Standards and Technology, 2020; Vassilev et al., 2024; International Organization for Standardization, 2019; MITRE, n.d.).

Several regulatory points carry uncertainty and should be verified against current texts. The EU AI Act (European Parliament and Council, 2024b) phases in its obligations, the dates of application for some of them have been subject to proposals for change that I could not confirm, and I am not certain how fraud-detection systems are classified, although I recall that the text excludes systems used to detect financial fraud from one high-risk category; counsel should confirm. The EU anti-money-laundering regulation (European Parliament and Council, 2024c) and DORA (European Parliament and Council, 2022b) have application dates that I believe fall in 2027 and January 2025, respectively, but these dates should be checked. The NIST Privacy Framework (National Institute of Standards and Technology, 2020) and ISO/IEC 27701 (International Organization for Standardization, 2019) may have been revised since the versions cited, and guidance on evaluating differential privacy (Near et al., 2025) should be consulted for any claim of formal protection. The confidentiality of suspicious activity reports and the meaning of tipping off differ between jurisdictions and are a matter for legal advice.

Figure 12: Implementation roadmap for governing explanation release in a financial consortium: inventory and classify, measure leakage in shadow, release with controls, and govern and adapt, with governance and enabling layers and a re-assessment loop



Implications for analytics practice

Practitioners can take five things from the results. First, measure leakage against baselines, not in isolation: report the attribute-inference AUC of an imputation attacker and of a prediction-only attacker alongside that of the explanation attacker, and state the advantage of the explanation over each. Second, test repeats: any noise-based control should be evaluated for the number of queries an attacker can make, using perturbed own features, and for the number of accounts it can use. Third, bind noise to identity if sticky noise is used, with tests that the entity key cannot be varied by the asker. Fourth, treat query volume per entity, not per account, as the quantity to monitor, since collusion pools account

budgets. Fifth, record utility next to leakage so that a control is not adopted for its protection while destroying the explanation. Leakage metrics for the governance record are the AUC advantage over each baseline, the share of confidential cases identified at a stated posterior and its precision, the number of queries needed to extract a model component to a stated fidelity, and the distribution of queries per entity. The simulation harness released with this article is a starting point for a shadow test, not a certified tool, and it should be adapted to the consortium's own model, features and attacker assumptions.

Limitations

Table 15 lists the main limitations. All data are synthetic and the hidden attribute was given a large, clean attribution, so the absolute numbers describe the testbed and should not be transferred. The attacker's labelled data for z are assumed correct, which is a strong assumption. Repeats are perturbations of own features with fixed context, and the attacker averages them; better attackers might do more. Sticky noise relies on an identity the service trusts. The accounting assumes Gaussian noise on one attribution with clipped sensitivity, and the resulting bounds are loose. Membership inference assumes a shadow model with the same distribution. Extraction concerns the own-feature part only. Utility is measured by agreement with the model's attributions and not by analyst performance (Bansal et al., 2021; Buçinca et al., 2021). Three seeds of one dataset design and one model family give narrow intervals for large effects that may understate variation elsewhere. Some reference details could not be fully confirmed during preparation, notably the page numbers and identifier of the 2022 membership-inference paper by Carlini and colleagues and the article numbers of three survey papers, and the reader should check those entries against the publishers' records.

Table 15: Assumptions and limitations, their likely effect on the findings and what would reduce them

Assumption or limitation	Likely effect on the findings	What would reduce it
All data, the hidden attribute and the fraud process are synthetic	Absolute AUC values describe the testbed; orderings and mechanisms are the findings	Replay on a consortium's own query logs under a privacy review
z enters the model with a large, clean attribution and the attacker's labels for z are correct	Undefended leakage is close to complete and is probably an upper bound for a rare attribute; label noise would lower it	Attackers with noisy or small labelled sets; attributes of weaker effect
Repeats are perturbations of own features with context fixed; the attacker averages the releases	Real repeats may be constrained by the service or detected; the averaging attack may be stronger with better design	Query-pattern detection (Juuti et al., 2019); adaptive attackers
Sticky noise is keyed to an entity identifier the service trusts	If an attacker can change the identifier or the entity is not unique, sticky noise reverts to fresh noise	Identity binding, key management, entity resolution tests
Privacy accounting assumes Gaussian noise on a single attribution with clipped sensitivity	The bounds are loose and describe a worst case, not the leakage measured	Tighter or empirical accounting (Dong et al., 2022); NIST SP 800-226 evaluation
Membership inference uses a shadow model with the same distribution and the label of the record	Strong attacker; real attackers have less, and the signal is weak even so	Weaker shadow data; label-only attacks (Choquette-Choo et al., 2021)
Extraction targets the own-feature contribution only, since context features cannot be set	Full model extraction was not evaluated	What-if interfaces and counterfactual releases (Aïvodji et al., 2020)
Utility is measured by agreement, rank correlation and faithfulness, not by analyst performance	A control that keeps these may still lower decision quality, and the reverse	Studies with analysts (Bansal et al., 2021; Buçinca et al., 2021)
Three seeds, one dataset design, no drift, one model family	Intervals are narrow for large effects and may understate variation elsewhere	More seeds, other model families and drift

Further research

Four lines of work follow. A replay on a consortium's own query logs, under privacy review and with real attributes of varying strength, would show how far the undefended leakage reported here transfers. Adaptive attackers that choose perturbations, filter noise with a model of the service and exploit identity weaknesses should be tested against sticky noise and per-entity budgets. Defences that change the model, such as training so that confidential contributions enter the score only through bounded aggregates, or private training (Abadi et al., 2016), should be compared on the same measures, including the cost in detection power. Studies with analysts are needed to learn which explanation parts, with which noise, analysts actually use, since utility here is a proxy (Bansal et al., 2021; Buçinca et al., 2021).

Conclusion

In a synthetic six-bank consortium, an unprotected TreeSHAP explanation revealed a confidential attribute almost perfectly (AUC 1.000), against 0.724 for imputation from own features and 0.918 for the score alone. Truncation and rounding left that unchanged. Fresh Gaussian noise protected one query and was averaged away by repeated queries with perturbed inputs, as a Gaussian theory for the z attribution anticipated (within 0.02 of a direct simulation). Entity-keyed noise did not erode but depends on a trusted identifier, per-account budgets were defeated by colluding accounts, and redaction of context attributions removed the channel for z while leaving own-feature explanations intact, though it did not remove the effect that the score and the net context carry. The noise that held leakage near the baseline cost most of the explanation, and formal accounting was loose against measured leakage. Membership inference was modest, extraction was cheaper from explanations than from scores by a factor of six, and in 2,000 Monte Carlo draws noise level and the use of entity-keyed noise dominated the Sobol indices. For governance, the findings favour treating explanation release as a disclosure channel, classifying contributed features, withholding or grouping attributions of confidential ones, measuring leakage against imputation and score baselines before and after any change, and monitoring query volume per entity. All numbers describe a synthetic testbed and are probably upper bounds for weaker attributes; they show which mechanisms matter and not how large the leakage would be in a real consortium.

REFERENCES

[1]. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16) (pp. 308–318). ACM. https://doi.org/10.1145/2976749.2978318

[2]. Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90–113. https://doi.org/10.1016/j.jnca.2016.04.007

[3]. Aïvodji, U., Bolot, A., & Gambs, S. (2020). Model extraction from counterfactual explanations (2009.01884). arXiv. https://arxiv.org/abs/2009.01884

[4]. Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods (1806.08049). arXiv. https://arxiv.org/abs/1806.08049

[5]. Aminul Islam, M., & Md, A. (2023). Adaptive Class-Imbalance Learning for High-Sensitivity Fraud Detection in Large-Scale Digital Transaction Networks. International Journal of Business and Economics Insights, 13(09), 01-33. https://doi.org/10.63125/bmxwv741

[6]. Asheke, M., & Md Maksudul, H. (2022). AI-Driven Demand Forecasting for Sporting-Goods and Teamwear Supply Chains Under Seasonal and Tournament Demand. International Journal of Business and Economics Insights, 2(1), 73–110. https://doi.org/10.63125/ed6by557

[7]. Asheke, M., & Mohammad Badrul, A. (2022). Development Of Functionalized Cellulose Nanofiber Adsorbents for Selective Critical-Metal Recovery from Industrial Wastewater. American Journal of Scholarly Research and Innovation, 1(02), 270–320. https://doi.org/10.63125/v6fxzq09

[8]. Balle, B., & Wang, Y.-X. (2018). Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In Proceedings of the 35th International Conference on Machine Learning (ICML 2018), PMLR 80 (pp. 394–403). PMLR. https://proceedings.mlr.press/v80/balle18a.html

[9]. Bansal, G., Wu, T., Zhou, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M. T., & Weld, D. S. (2021). Does the whole exceed its parts? The effect of AI explanations on complementary team performance. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21) (pp. 1–16). ACM. https://doi.org/10.1145/3411764.3445717

[10]. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

[11]. BIS Innovation Hub. (2023). Project Aurora: The power of data, technology and collaboration to combat money laundering across institutions and borders. Bank for International Settlements. https://www.bis.org/publ/othp66.htm

[12]. Board of Governors of the Federal Reserve System & Office of the Comptroller of the Currency. (2011). Supervisory guidance on model risk management (SR 11-7). Federal Reserve. https://www.federalreserve.gov/supervisionreg/srletters/sr1107.htm

[13]. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17) (pp. 1175–1191). ACM. https://doi.org/10.1145/3133956.3133982

[14]. Bracke, P., Datta, A., Jung, C., & Sen, S. (2019). Machine learning explainability in finance: An application to default risk analysis (Staff Working Paper No. 816). Bank of England. https://www.bankofengland.co.uk/working-paper/2019/machine-learning-explainability-in-finance-an-application-to-default-risk-analysis

[15]. Buçinca, Z., Malaya, M. B., & Gajos, K. Z. (2021). To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. Proceedings of the ACM on Human-Computer Interaction, 5(CSCW1), 1–21. https://doi.org/10.1145/3449287

[16]. Bun, M., & Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Theory of Cryptography: TCC 2016-B, Lecture Notes in Computer Science 9985 (pp. 635–658). Springer. https://doi.org/10.1007/978-3-662-53641-4_24

[17]. Bussmann, N., Giudici, P., Marinelli, D., & Papenbrock, J. (2021). Explainable machine learning in credit risk management. Computational Economics, 57(1), 203–216. https://doi.org/10.1007/s10614-020-10042-0

[18]. Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. Information Sciences, 557, 317–331. https://doi.org/10.1016/j.ins.2019.05.042

[19]. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., & Tramèr, F. (2022). Membership inference attacks from first principles. In 2022 IEEE Symposium on Security and Privacy (SP). IEEE. https://arxiv.org/abs/2112.03570

[20]. Chen, H., Janizek, J. D., Lundberg, S., & Lee, S.-I. (2020). True to the model or true to the data? (2006.16234). arXiv. https://arxiv.org/abs/2006.16234

[21]. Chen, Z., Van Khoa, L. D., Teoh, E. N., Nazir, A., Karuppiah, E. K., & Lam, K. S. (2018). Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: A review. Knowledge and Information Systems, 57(2), 245–285. https://doi.org/10.1007/s10115-017-1144-z

[22]. Choquette-Choo, C. A., Tramer, F., Carlini, N., & Papernot, N. (2021). Label-only membership inference attacks. In Proceedings of the 38th International Conference on Machine Learning (ICML 2021), PMLR 139 (pp. 1964–1974). PMLR. https://proceedings.mlr.press/v139/choquette-choo21a.html

[23]. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3784–3797. https://doi.org/10.1109/TNNLS.2017.2736643

[24]. Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In 2015 IEEE Symposium Series on Computational Intelligence (pp. 159–166). IEEE. https://doi.org/10.1109/SSCI.2015.33

[25]. Dinur, I., & Nissim, K. (2003). Revealing information while preserving privacy. In Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '03) (pp. 202–210). ACM. https://doi.org/10.1145/773153.773173

[26]. Dong, J., Roth, A., & Su, W. J. (2022). Gaussian differential privacy. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 84(1), 3–37. https://doi.org/10.1111/rssb.12454

[27]. Douceur, J. R. (2002). The Sybil attack. In Peer-to-Peer Systems: First International Workshop, IPTPS 2002, Lecture Notes in Computer Science 2429 (pp. 251–260). Springer. https://doi.org/10.1007/3-540-45748-8_24

[28]. Duddu, V., & Boutet, A. (2022). Inferring sensitive attributes from model explanations. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22) (pp. 416–425). ACM. https://doi.org/10.1145/3511808.3557362

[29]. Dwork, C. (2006). Differential privacy. In Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Lecture Notes in Computer Science 4052 (pp. 1–12). Springer. https://doi.org/10.1007/11787006_1

[30]. Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3-4), 211–407. https://doi.org/10.1561/0400000042

[31]. Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography (TCC 2006), Lecture Notes in Computer Science 3876 (pp. 265–284). Springer. https://doi.org/10.1007/11681878_14

[32]. Dwork, C., Rothblum, G. N., & Vadhan, S. (2010). Boosting and differential privacy. In 2010 IEEE 51st Annual Symposium on Foundations of Computer Science (pp. 51–60). IEEE. https://doi.org/10.1109/FOCS.2010.12

[33]. European Parliament and Council. (2016). Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation). Official Journal of the European Union, L 119, 1-88. https://eur-lex.europa.eu/eli/reg/2016/679/oj

[34]. European Parliament and Council. (2022a). Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union (NIS 2 Directive). Official Journal of the European Union, L 333, 80-152. https://eur-lex.europa.eu/eli/dir/2022/2555/oj

[35]. European Parliament and Council. (2022b). Regulation (EU) 2022/2554 on digital operational resilience for the financial sector (DORA). Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2022/2554/oj

[36]. European Parliament and Council. (2024b). Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2024/1689/oj

[37]. European Parliament and Council. (2024c). Regulation (EU) 2024/1624 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing. Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2024/1624/oj

[38]. Ferry, J., Aïvodji, U., Gambs, S., Huguet, M.-J., & Siala, M. (2023). Probabilistic dataset reconstruction from interpretable models (2308.15099). arXiv. https://arxiv.org/abs/2308.15099

[39]. Financial Action Task Force. (2012). International standards on combating money laundering and the financing of terrorism & proliferation: The FATF Recommendations (updated). FATF. https://www.fatf-gafi.org/en/publications/Fatfrecommendations/Fatf-recommendations.html

[40]. Financial Action Task Force. (2022). Partnering in the fight against financial crime: Data protection, technology and private sector information sharing. FATF. https://www.fatf-gafi.org/en/publications/Digitaltransformation/Partnering-in-the-fight-against-financial-crime.html

[41]. Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15) (pp. 1322–1333). ACM. https://doi.org/10.1145/2810103.2813677

[42]. Gal-Or, E., & Ghose, A. (2005). The economic incentives for sharing security information. Information Systems Research, 16(2), 186–208. https://doi.org/10.1287/isre.1050.0053

[43]. Ganju, K., Wang, Q., Yang, W., Gunter, C. A., & Borisov, N. (2018). Property inference attacks on fully connected neural networks using permutation invariant representations. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18) (pp. 619–633). ACM. https://doi.org/10.1145/3243734.3243834

[44]. Gong, N. Z., & Liu, B. (2016). You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In Proceedings of the 25th USENIX Security Symposium (pp. 979–995). USENIX Association. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/gong

[45]. Harder, F., Bauer, M., & Park, M. (2020). Interpretable and differentially private predictions. In Proceedings of the AAAI Conference on Artificial Intelligence, 34(04) (pp. 4083–4090). AAAI Press. https://doi.org/10.1609/aaai.v34i04.5827

[46]. Herman, J., & Usher, W. (2017). SALib: An open-source Python library for sensitivity analysis. Journal of Open Source Software, 2(9), Article 97. https://doi.org/10.21105/joss.00097

[47]. Hitaj, B., Ateniese, G., & Pérez-Cruz, F. (2017). Deep models under the GAN: Information leakage from collaborative deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17) (pp. 603–618). ACM. https://doi.org/10.1145/3133956.3134012

[48]. Hu, H., Salčić, Z., Sun, L., Dobbie, G., Yu, P. S., & Zhang, X. (2022). Membership inference attacks on machine learning: A survey. ACM Computing Surveys, 54(11s), Article 235. https://doi.org/10.1145/3523273

[49]. International Organization for Standardization. (2019). ISO/IEC 27701:2019 Security techniques: Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management: Requirements and guidelines. ISO. https://www.iso.org/standard/71670.html

[50]. Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., & Papernot, N. (2020). High accuracy and high fidelity extraction of neural networks. In Proceedings of the 29th USENIX Security Symposium (pp. 1345–1362). USENIX Association. https://www.usenix.org/conference/usenixsecurity20/presentation/jagielski

[51]. Jayaraman, B., & Evans, D. (2019). Evaluating differentially private machine learning in practice. In Proceedings of the 28th USENIX Security Symposium (pp. 1895–1912). USENIX Association. https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman

[52]. Jayaraman, B., & Evans, D. (2022). Are attribute inference attacks just imputation? In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22) (pp. 1569–1582). ACM. https://doi.org/10.1145/3548606.3560663

[53]. Jensen, R. I. T., & Iosifidis, A. (2023). Fighting money laundering with statistics and machine learning. IEEE Access, 11, 8889–8903. https://doi.org/10.1109/ACCESS.2023.3239549

[54]. Juuti, M., Szyller, S., Marchal, S., & Asokan, N. (2019). PRADA: Protecting against DNN model stealing attacks. In 2019 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 512–527). IEEE. https://doi.org/10.1109/EuroSP.2019.00044

[55]. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146–3154. https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree

[56]. Kifer, D., & Machanavajjhala, A. (2014). Pufferfish: A framework for mathematical privacy definitions. ACM Transactions on Database Systems, 39(1), Article 3. https://doi.org/10.1145/2514689

[57]. Kuppa, A., & Le-Khac, N.-A. (2020). Black box attacks on explainable artificial intelligence (XAI) methods in cyber security. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1–8). IEEE. https://doi.org/10.1109/IJCNN48605.2020.9206780

[58]. Lakkaraju, H., & Bastani, O. (2020). "How do I fool you?": Manipulating user trust via misleading black box explanations. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES '20) (pp. 79–85). ACM. https://doi.org/10.1145/3375627.3375833

[59]. Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., & Lin, Z. (2022). When machine learning meets privacy: A survey and outlook. ACM Computing Surveys, 54(2), Article 31. https://doi.org/10.1145/3436755

[60]. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765–4774. https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions

[61]. Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles (1802.03888). arXiv. https://arxiv.org/abs/1802.03888

[62]. Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

[63]. Luo, X., Jiang, Y.-F., & Xiao, X. (2022). Feature inference attack on Shapley values. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22) (pp. 2233–2247). ACM. https://doi.org/10.1145/3548606.3560573

[64]. Mansura Akter, E., & Sumaiya, M. (2022). Antimicrobial Resistance Profiles of Acinetobacter Baumannii and Vibrio Cholerae in Hospital Wastewater and Adjacent Household Water Sources. American Journal of Data Science and Analytics, 3(09), 01-55. https://doi.org/10.63125/gbervt14

[65]. McMahan, B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR 54 (pp. 1273–1282). PMLR. https://proceedings.mlr.press/v54/mcmahan17a.html

[66]. Md Imran, H., & Jannatul, F. (2022). Molecular and Precision Rehabilitation Molecular Mechanisms Linking Mitochondrial Dysfunction, Inflammation, and Muscle Degeneration with Functional Decline and Rehabilitation Response in Neuromuscular Disease. American Journal of Interdisciplinary Studies, 12(08), 01-44. https://doi.org/10.63125/0ym5bh79

[67]. Md Maksudul, H., & Md Tazul Islam, C. (2022). Machine Learning-Based Cash-Flow Forecasting for Improving Financial Planning in U.S. Small and Mid-Sized Businesses. International Journal of Scientific Interdisciplinary Research, 12(12), 01–38. https://doi.org/10.63125/w3pgjx97

[68]. Md Tazul Islam, C., & Ahmed, H. C. H. R. (2023). Predictive Operations Analytics for Resource Allocation and Service-Capacity Planning in Data-Driven Organizations. Review of Applied Science and Technology, 13(02), 01–41. https://doi.org/10.63125/z3sm3089

[69]. Md. Morshedul, I., & Aminul Islam, M. (2022). A Quantitative Evaluation of Explainable Ensemble Learning for Probabilistic Risk Estimation and Decision Reliability in Enterprise Analytics. American Journal of Data Science and Analytics, 3(02), 01-38. https://doi.org/10.63125/k22shk74

[70]. Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 691–706). IEEE. https://doi.org/10.1109/SP.2019.00029

[71]. Milli, S., Schmidt, L., Dragan, A. D., & Hardt, M. (2019). Model reconstruction from model explanations. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19) (pp. 1–9). ACM. https://doi.org/10.1145/3287560.3287562

[72]. Mironov, I. (2017). Rényi differential privacy. In 2017 IEEE 30th Computer Security Foundations Symposium (CSF) (pp. 263–275). IEEE. https://doi.org/10.1109/CSF.2017.11

[73]. MITRE. (n.d.). ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems. MITRE Corporation. https://atlas.mitre.org

[74]. Monetary Authority of Singapore. (2018). Principles to promote fairness, ethics, accountability and transparency (FEAT) in the use of artificial intelligence and data analytics in Singapore's financial sector. Monetary Authority of Singapore. https://www.mas.gov.sg/publications/monographs-or-information-paper/2018/feat

[75]. Nasr, M., Shokri, R., & Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 739–753). IEEE. https://doi.org/10.1109/SP.2019.00065

[76]. National Institute of Standards and Technology. (2020). NIST Privacy Framework: A tool for improving privacy through enterprise risk management, version 1.0 (NIST CSWP 01162020). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.CSWP.01162020

[77]. Near, J. P., Darais, D., Lefkovitz, N., & Howarth, G. S. (2025). Guidelines for evaluating differential privacy guarantees (NIST SP 800-226). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-226

[78]. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3), 559–569. https://doi.org/10.1016/j.dss.2010.08.006

[79]. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security (AsiaCCS '17) (pp. 506–519). ACM. https://doi.org/10.1145/3052973.3053009

[80]. Patel, N. A., Shokri, R., & Zick, Y. (2022). Model explanations with differential privacy. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22) (pp. 1895–1904). ACM. https://doi.org/10.1145/3531146.3533235

[81]. Pawelczyk, M., Lakkaraju, H., & Neel, S. (2023). On the privacy risks of algorithmic recourse. In Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS 2023), PMLR 206 (pp. 9680–9696). PMLR. https://proceedings.mlr.press/v206/pawelczyk23a.html

[82]. Pourhabibi, T., Ong, K.-L., Kam, B. H., & Boo, Y. L. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. Decision Support Systems, 133, Article 113303. https://doi.org/10.1016/j.dss.2020.113303

[83]. Psychoula, I., Gutmann, A., Mainali, P., Lee, S. H., Dunphy, P., & Petitcolas, F. A. P. (2021). Explainable machine learning for fraud detection. Computer, 54(10), 49–59. https://doi.org/10.1109/MC.2021.3081249

[84]. Rigaki, M., & Garcia, S. (2023). A survey of privacy attacks in machine learning. ACM Computing Surveys, 56(4), Article 101. https://doi.org/10.1145/3624010

[85]. Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., & Jégou, H. (2019). White-box vs black-box: Bayes optimal strategies for membership inference. In Proceedings of the 36th International Conference on Machine Learning (ICML 2019), PMLR 97 (pp. 5558–5567). PMLR. https://proceedings.mlr.press/v97/sablayrolles19a.html

[86]. Sadia, A., & Md. Mosheur, R. (2023). Consumer Trust, Perceived Authenticity, and Personalized Digital Marketing: A Quantitative Decision-Science Study. International Journal of Scientific Interdisciplinary Research, 4(2), 252–295. https://doi.org/10.63125/cajkqt60

[87]. Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., & Backes, M. (2019). ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. In Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS). Internet Society. https://doi.org/10.14722/ndss.2019.23119

[88]. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., & Tarantola, S. (2010). Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Computer Physics Communications, 181(2), 259–270. https://doi.org/10.1016/j.cpc.2009.09.018

[89]. Selbst, A. D., & Powles, J. (2017). Meaningful information and the right to explanation. International Data Privacy Law, 7(4), 233–242. https://doi.org/10.1093/idpl/ipx022

[90]. Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15) (pp. 1310–1321). ACM. https://doi.org/10.1145/2810103.2813687

[91]. Shokri, R., Strobel, M., & Zick, Y. (2021). On the privacy risks of model explanations. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (AIES '21) (pp. 231–241). ACM. https://doi.org/10.1145/3461702.3462533

[92]. Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 3–18). IEEE. https://doi.org/10.1109/SP.2017.41

[93]. Skopik, F., Settanni, G., & Fiedler, R. (2016). A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing. Computers & Security, 60, 154–176. https://doi.org/10.1016/j.cose.2016.04.003

[94]. Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (2020). Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES '20) (pp. 180–186). ACM. https://doi.org/10.1145/3375627.3375830

[95]. Sobol', I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Mathematics and Computers in Simulation, 55(1-3), 271–280. https://doi.org/10.1016/S0378-4754(00)00270-6

[96]. Song, L., & Mittal, P. (2021). Systematic evaluation of privacy risks of machine learning models. In Proceedings of the 30th USENIX Security Symposium (pp. 2615–2632). USENIX Association. https://www.usenix.org/conference/usenixsecurity21/presentation/song

[97]. Stabak, D., & Md Shahid, N. (2022). Empirical Assessment of Uncertainty-Aware Real-Time Decision-Making for Reliable and Resilient Cyber-Physical Systems. International Journal of Scientific Interdisciplinary Research, 1(01), 432–477. https://doi.org/10.63125/q3r4kk16

[98]. Sundararajan, M., & Najmi, A. (2020). The many Shapley values for model explanation. In Proceedings of the 37th International Conference on Machine Learning (ICML 2020), PMLR 119 (pp. 9269–9278). PMLR. https://proceedings.mlr.press/v119/sundararajan20b.html

[99]. Suzumura, T., Zhou, Y., Baracaldo, N., Ye, G., Houck, K., Kawahara, R., Anwar, A., Stavarache, L. L., Watanabe, Y., Loyola, P., Klyashtorny, D., Ludwig, H., & Bhaskaran, K. (2019). Towards federated graph learning for collaborative financial crimes detection (1909.12946). arXiv. https://arxiv.org/abs/1909.12946

[100]. Syed Feroz, A., & Md Mahamud Pervaz, P. (2022). Root-Cause Analysis and FMEA-Driven Reliability Improvement in Gas Production Facilities: A Multi-Year Empirical Study. International Journal of Scientific Interdisciplinary Research, 12(10), 01–53. https://doi.org/10.63125/70msp878

[101]. Syed Feroz, A., & Shabbir, A. (2022). Finite Element-Based Thermomechanical Optimization of Pressure Components for Extending Industrial Boiler Service Life. Journal of Sustainable Development and Policy, 12(12), 01-51. https://doi.org/10.63125/x24pht28

[102]. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. In Proceedings of the 25th USENIX Security Symposium (pp. 601–618). USENIX Association. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer

[103]. Vassilev, A., Oprea, A., Fordyce, A., & Anderson, H. (2024). Adversarial machine learning: A taxonomy and terminology of attacks and mitigations (NIST AI 100-2e2023). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-2e2023

[104]. Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a right to explanation of automated decision-making does not exist in the General Data Protection Regulation. International Data Privacy Law, 7(2), 76–99. https://doi.org/10.1093/idpl/ipx005

[105]. Wagner, T. D., Mahbub, K., Palomar, E., & Abdallah, A. E. (2019). Cyber threat intelligence sharing: Survey and research directions. Computers & Security, 87, Article 101589. https://doi.org/10.1016/j.cose.2019.101589

[106]. Wang, G. (2019). Interpret federated learning with Shapley values (1905.04519). arXiv. https://arxiv.org/abs/1905.04519

[107]. Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics (1908.02591). arXiv. https://arxiv.org/abs/1908.02591

[108]. West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. Computers & Security, 57, 47–66. https://doi.org/10.1016/j.cose.2015.09.005

[109]. Wu, X., Fredrikson, M., Jha, S., & Naughton, J. F. (2016). A methodology for formalizing model-inversion attacks. In 2016 IEEE 29th Computer Security Foundations Symposium (CSF) (pp. 355–370). IEEE. https://doi.org/10.1109/CSF.2016.32

[110]. Ye, J., Maddi, A., Murakonda, S. K., Bindschaedler, V., & Shokri, R. (2022). Enhanced membership inference attacks against machine learning models. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22) (pp. 3093–3106). ACM. https://doi.org/10.1145/3548606.3560675

[111]. Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S. (2018). Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF) (pp. 268–282). IEEE. https://doi.org/10.1109/CSF.2018.00027

[112]. Zhao, X., Zhang, W., Xiao, X., & Lim, B. Y. (2021b). Exploiting explanations for model inversion attacks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 682–692). IEEE. https://doi.org/10.1109/ICCV48922.2021.00072

[113]. Zhu, L., Liu, Z., & Han, S. (2019). Deep leakage from gradients. In Advances in Neural Information Processing Systems 32 (NeurIPS 2019). Curran Associates. https://papers.nips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html