Towards a Game-theoretic Understanding of Explanation-based Membership Inference Attacks

Kavita Kumari, Murtuza Jadliwala, Sumit Kumar Jha, Anindya Maiti

Abstract. Model explanations improve the transparency of black-box machine learning (ML) models and their decisions; however, they can also enable privacy threats like membership inference attacks (MIA). Existing works have only analyzed MIA in a single interaction scenario between an adversary and the target ML model, missing the factors that influence an adversary's capability to launch MIA in repeated interactions. These works also assume the attacker knows the model's structure, which isn't always true, leading to suboptimal thresholds for identifying members. This paper examines explanation-based threshold attacks, where an adversary uses the variance in explanations through repeated interactions to perform MIA. We use a continuous-time stochastic signaling game to model these interactions. Unaware of the system's exact type (honest or malicious), the adversary plays a stopping game to gather explanation variance and compute an optimal threshold for membership determination. We propose a sound mathematical formulation to prove that such an optimal threshold exists, which can be used to launch MIA and identify conditions for a unique Markov perfect equilibrium in this dynamic system. Finally, we evaluate various factors affecting an adversary's ability to conduct MIA in repeated settings through simulations.

1 Introduction

Due to the complex and black-box nature of machine learning (ML) models, understanding the underlying reasons behind model decisions is often challenging. This has led to the design of different model explanation techniques [23, 29, 33]. Simultaneously, explanations expose an attack surface that can be exploited to infer private model information [31] or launch adversarial attacks against the model [19, 35]. One feasible attack from model explanations is Membership Inference Attacks (MIAs). MIAs have been extensively studied in the literature, where adversaries analyze ML models to formulate attacks aimed at discerning the membership of specific data points. MIAs can be classified into binary classifier-based [32], metric-based [22], and differential comparison-based attacks [18].

In this work, we focus on metric-based MIAs, particularly *model explanation-based MIAs*, where an adversary could use model explanations to infer the training set membership of target data points. Shokri et al. [31] demonstrated that variance in gradient-based explanations can confirm membership compared to a predefined threshold. However, it was empirically analyzed for a single query instance. This work further analyzes how explanation variance changes with repeated queries from an adversary. In

this iterative interaction, the adversary sends repeated queries to the system (ML model and explanation method) using historical information to find the optimal threshold for explanation variance. Additionally, while it is straightforward to compute an optimal threshold if the training set membership is known [22], the question arises: how can an explanation-based threshold attack be executed when an adversary lacks knowledge of the model and its training process?

An adversary that iteratively interacts with the target system to compute the explanation variance threshold raises several questions: What is the optimal duration for this interaction? Can the system detect and prevent such malicious interactions? How can the system serve both honest and malicious users effectively? While honest and malicious users may formulate similar queries, the emphasis lies on the malicious user's intention to initiate MIA. Thus, the value of an explanation for an honest end-user is based on its relevance, explaining the model's decision for the query. However, a malicious end-user evaluates an explanation's value based on the information it contains for potential exploitation in launching MIAs. Intuitively, the duration, pattern, and structure of such repeated interactions could impact the degree of private information disclosure by the system. Nevertheless, the current comprehension of this phenomenon is insufficient, particularly when considering the presence of a strategic adversary whose goal is to minimize the attack cost and path towards undermining the system's privacy and a strategic system that is aiming to prevent this without having full knowledge of the nature of the end-user (adversarial or non-adversarial) engaged in the interaction.

We aim to bridge this research gap by using game theory to model interactions between an adversary and an ML system in the above context. Specifically, we use a continuous-time stochastic signaling game to capture the complexities of these interactions. In particular, we make the following contributions in this paper:

1. We model the interactions between an ML system and an adversary as a *two-player continuous-time signaling game*, where the variance of the generated explanations (by the ML system) evolve according to a *stochastic differential equation (SDE)*.
2. We then characterize the *Markov Perfect Equilibrium (MPE)* of the above stochastic game as a pair of optimal functions $U(\pi)$ and $L(\pi)$, where $U(\pi)$ is the optimal variance path for the explanations generated by the system, $L(\pi)$ is the optimal variance path for the explanations given by the system to an adversary after adding some noise, and π is the belief of the system about the type of the adversary.
3. We evaluate the game for different gradient-based explanation methods, namely, *Integrated Gradients* [38], *Gradient*Input* [34], *LRP* [3] and *Guided Backpropagation* [37]. By means of experiments using benchmark datasets, we demonstrate that the capability of an adversary to launch MIA depends on factors such as the explanation method, input dimensionality, model size, and number of training rounds.

2 Background and Preliminaries

2.1 Gradient based Explanations

For some input data point $\overrightarrow{x} \in \mathbb{R}^n$ and a classification model F_θ, an explanation method $\mathcal{H}$ simply explains model decisions, i.e., it outputs some justification/explanation

of why the model F_θ returned a particular label $y = F_\theta(\overrightarrow{x})$. In this work, we consider feature-based explanations, where the output of the explanation function is an influence (or attribution) vector and where the element $\mathcal{H}_i(\overrightarrow{x})$ of the vector represents the degree to which the i^{th} feature influences the predicted label y of the data point $\overrightarrow{x}$. Specifically, we consider the Gradient method [34], Integrated Gradient (IG) method [38], Layer-wise Relevance Propagation (LRP) [3], and Guided Backpropagation [37]. For more details, please refer to the extended version of the paper [21].

2.2 Membership Inference Attacks

In membership inference attacks (MIA), an adversary with a target dataset $\mathcal{X}_{tgt} \subset \mathcal{R}^n$ aims to identify which data points belong to a target model's training set $\mathcal{X}_{tr}$. The adversary predicts membership by assessing if each point $\overrightarrow{x} \in \mathcal{X}_{tgt}$ is also in $\mathcal{X}_{tr}$. Intuitively, a low model loss typically translates to a prediction vector dominated by the true label, resulting in a high variance, which may indicate model certainty [31] and, thus, the data point (under consideration) as a member of the training dataset. The variance of the feature-based explanation to determine data point membership is as follows:

$$\text{Membership}_{Expl,\tau_E}(\overrightarrow{x}) = \begin{cases} \text{True}, & Var(\mathcal{H}_{GRAD}(\overrightarrow{x})) \leq \tau_E \\ \text{False}, & otherwise \end{cases}$$

where the variance of some vector $\overrightarrow{v} \in \mathbb{R}^n$ is calculated as: $Var(\overrightarrow{v}) = \sum_{i=1}^{n}(v_i - \mu_{\overrightarrow{v}})^2$, where$\mu_{\overrightarrow{v}} = \frac{1}{n}\sum_{i=1}^{n} v_i$.

2.3 Geometric Brownian Motion

A Geometric Brownian Motion (GBM) is a continuous-time stochastic process commonly used to model the evolution of a variable that exhibits random fluctuations over time. A general GBM state process s_t satisfies the stochastic differential equation:

$$ds_t = a(s_t, u(s_t, t), t)s_t dt + b(s_t, u(s_t, t), t)s_t dW_t$$

where, $a(s_t, u(s_t, t), t)$ and $b(s_t, u(s_t, t), t)$ are the drift and volatility parameters of the state process s_t, respectively, W_t is a standard Brownian motion with mean $= 0$ and variance $= t$, and $u(s_t, t)$ is the control. In this paper, an adversary aims to reach a variance threshold to launch explanation-based attacks by repeatedly interacting with the ML model using appropriate queries and historical interaction data. Here, we model the evolving explanation variance EX^v as a GBM due to its ability to capture periodic and random fluctuations in a non-negative continuous-time process.

2.4 Optimal Control and the Stopping Problem

In a two-player game, each agent makes an optimal control decision to either continue or stop interacting with the other agent. Such problems involving optimal control are

usually modeled using Bellman's equation and solved with dynamic programming. Let $u_i(s_t, t)$ represent the control of agent i when the system is in state s_t at time t. The value function, denoted by $H_i(s_t, t)$, represents the optimal payoff/reward of the agent i over the interval $t = [0, T]$ can be written as:

$$H_i(s_t, t) = \max_{u_i} \int_0^T f(s_t, u(s_t, t), t)dt$$

Where $f(s_t, u(s_t, t), t)$ is the instantaneous payoff/reward a player can get given the state (s_t) and the control used (u) at time t. the Bellman equation is a *partial differential equation or PDE*, referred to as the *Hamilton Jacobi Bellman (HJB)* equation, and can be written as:

$$rH(s_t, t) = f(s_t, u^*, t) + \frac{\partial H}{\partial t} + \frac{\partial H}{\partial s_t} a(s_t, u^*, t) + \frac{1}{2} \frac{\partial^2 H}{\partial s_t^2} b(s_t, u^*, t)^2$$

Where $u^* = u(s_t, t)$ is the optimal value of the control variable. Using the above equation, we represent the value functions of both the adversary and the system. The optimal control u (for both the adversary and the system) is binary: $u = 1$ means "stop" interacting, and $u = 0$ means "continue" the interaction.

Stopping Problem: A stopping problem models the decision to continue an activity for an instantaneous payoff $f(s_t, u(s_t, t), t)$ or stop for a termination payoff $\lambda(s_t, T)$. It is determined based on the payoff he/she is expected to receive in the next instant. The stopping rule for the state boundary value s_t^* at which an agent decides to stop and get the termination payoff is given by:

$$u(s_t, t) = \begin{cases} \text{stop}, & s_t >= s_t^* \\ \text{continue}, & s_t < s_t^* \end{cases}$$

In other words, when the agent decides to stop, he/she gets:

$$H(s_t, T) = \lambda(s_t, T) \quad \forall s_t \geq s_t^*$$

Value Matching and Smooth Pasting Conditions: Two boundary conditions are required to solve the HJB equation outlined above, First, *value matching condition* tells an agent that if they decide to stop (at that defined boundary), then the termination payoff equals the continuation payoff. It is given by:

$$H(s_t^*, t) = \lambda(s_t^*, t) \quad \forall t$$

Second, the *smooth pasting condition* ensures a smooth transition at the stopping boundary. Intuitively, it helps pin the optimal decision boundary, s_t^* and is given by:

$$H_{s_t}(s_t^*, t) = \lambda_{s_t}(s_t^*, t) \quad \forall t$$

where $H_{s_t}(s_t^*, t)$ is the derivative of $H(s_t^*, t)$ with respect to the state s_t. If one or both conditions are unsatisfied, stopping at the boundary s_t^* can't be optimal. Therefore, an agent should continue and again decide at the next time instant.

3 Game Model

Next, we present an intuitive description of the problem followed by its formal setup as a signaling game. Further, we also characterize the equilibrium concept in this setup.

3.1 Intuition

We consider a platform, the `system`, offering an ML model and feature-based explanations as a black-box service. `End-users` request labels and explanations but cannot download the model. The `system` interacts with two types of users, *honest* and *malicious*, without knowing their type. Honest users seek explanations for their queries, while malicious users exploit explanation variance to conduct MIAs without detection.

The malicious `end-user` or an adversary interacts repeatedly with the `system` to obtain explanations for their formulated queries, leveraging prior variance history modeled with GBM. Explanation-based MIAs rely on explanation variance thresholds [31], making GBM a suitable model for this variance. GBM captures historical data integration and ensures that explanation variance stays positive, reflecting periodic and random fluctuations. Note: Our goal is to establish mathematical proof of an optimal explanation variance threshold that enables an adversary to launch MIAs. Thus, we are not concerned about how an adversary models the query space. The malicious `end-user` strategically decides when to stop interacting with the `system` to achieve their attack objective, modeled as a continuous-time signaling game. If the `system` fails to detect the malicious behavior and considers it honest, this is termed "pooling" or "on-equilibrium path" behavior. Deviations from this behavior are termed "separating" or "off-equilibrium path" behavior. Throughout the game, the malicious `end-user` aims to conduct a threshold-based MIA by leveraging accumulated information (labels + explanations) up to that point.

More specifically, the malicious `end-user` decides whether to continue querying for explanations or to stop and attack the `system` to avoid detection. Conversely, the `system`, upon receiving requests, must decide whether to continue providing explanations and how much noise to add or to block the `end-user` based on an optimal variance path $U(\pi)$. Note: The `system` has imperfect information about the `end-user`'s type. Thus, it determines the explanation's noise level based on its Bayesian prior or belief (π).

Based on this stopping game formulation, we structure the model payoffs for both the `system` and the `end-users` (malicious and honest). Additionally, according to each interaction instance between the `system` and the `end-user`, we formulate the noise and the stopping responses. As mentioned before, the added noise/perturbation to the generated explanation is based on the `system`'s belief pertinent to the activity history of the `end-user`. For an honest `end-user`, the value of an explanation lies in its relevance — the information it contains explaining the model's decision for the query sent to the benign `end-user`. Conversely, for a malicious `end-user`, the value of an explanation also depends on its relevance — the amount of information it holds that can be exploited by the malicious `end-user` to launch MIAs. A detailed explanation of the payoffs and its design is outlined in Section 3.2.

In this preliminary effort, we formally model the above interactions between a single `end-user` (type determined by nature) and the `system` within a stochastic game-theoretic framework, and further analyze it to answer the following two high-level questions: When does a malicious `end-user` decide to stop the play and finally compromise the `system`? How does the `system` make the strategic decision to block a potential malicious `end-user` while continuing to give relevant explanations to potentially honest `end-users`?

3.2 Setup and Assumptions

We model the above scenario as a two-player, continuous-time, imperfect-information game with repeated play. This framework allows for modeling how a malicious `end-user` may deviate from pooling behavior (stopping time) at any point and how the `system` may detect it. Using GBM to model explanation variance involves abrupt transitions, making continuous-time modeling more effective for its evolution. Due to their ability to capture such dynamics effectively, continuous-time frameworks are commonly chosen in literature for problems involving stopping times. The game has two players: Player 1 is the `end-user`, of privately known type $\Theta \rightarrow \{h, m\}$ (i.e., honest or malicious), who wants to convince Player 2 (i.e., `system`) that he/she is honest. The game begins with *nature* picking an `end-user` of a particular type, and we analyze repeated play between this `end-user` (selected by nature) and the `system`, which occurs in each continuous-time, $t \in R$. As the `system` has imperfect information about the type of `end-user`, it assigns an initial belief $\pi_0 = Pr(\Theta = h)$. We assume both players are *risk neutral*, i.e., indifferent to taking a risk, and each player *discounts* payoffs at a constant rate r. Variance (EX_t^v) computed for an explanation generated by an explanation method of the `system` follows a GBM, and is given by:

$$dEX_t^v = \mu EX_t^v dt + \sigma EX_t^v dW_t$$

where, μ is the constant *drift* and $\sigma > 0$ is the constant *volatility* of the variance process EX_t^v, and $EX_0^v = ex_0^v > 0$. W_t is a standard Brownian motion with mean = 0 and variance = t. To ensure finite payoffs at each continuous time t, we assume $\mu < r$. The state of the game is represented by the process (EX_t^v, π_t), where π_t is the belief of the `system` about the type of the `end-user` at time t.

The `system` wants to give informative or relevant explanations to the honest `end-user`, but noisy explanations to the malicious `end-user`. Hence, depending on the `system`'s belief about the type of the `end-user`, it will decide how much noise (perturbation) to add to each released explanation, according to the generated variance. Let $U(\pi_t)$ denote the optimal variance path (or functional path) for the `system` - a non-increasing cut-off function which tells the `system` the optimal explanation variance computed for an explanation generated by an explanation method and $L(\pi_t)$ denote the optimal explanation variance path for the `end-user` - an increasing cut-off function which tells the `end-user` the optimal explanation variance for the explanation given by the `system` at given belief π_t. To simplify the resulting analysis, we assume that the explanations variance computed by the `system` and explanations variance computed by the `end-user` are just different realizations of the explanation variance process

EX_t^v. We denote $ex_{s_y,t}^v$ and $ex_{e_u,t}^v$ as the `system`'s and `end-user`'s realization of the process EX_t^v, respectively. Moreover, as the `system` would add some calculated noise to the generated explanation based on its Bayesian belief, we assume that $U(\pi_t) \geq L(\pi_t),\ \forall \pi_t$. Finally, as we are only interested in modeling the interactions between a malicious `end-user` and the `system`, any reference to an `end-user` from this point on implies a malicious `end-user`, unless explicitly stated otherwise. Next, we outline a few other relevant model parameters before characterizing the concept of equilibrium in the proposed game model.

Information Environment: Let $\mathcal{F}_t = \sigma(\{EX_s^v\} : 0 \leq s \leq t)$ be the `end-user`'s information environment, which is the *sigma-algebra* generated by the variance process EX^v. In other words, $\mathcal{F}_t$ represents the information contained in the public history of the explanation variance process. The `system`'s information environment is denoted by $\mathcal{F}_t^+ = \sigma(\{EX_s^v, \phi_s\} : 0 \leq s \leq t)$, where EX_s^v is the variance process representing the history of explanations variance and ϕ_s is the stochastic process representing the historical activity of the `end-user`. If ρ is the time that `end-user` decides to stop, then $\phi_t = \rho$ if $\rho \leq t$ and ∞ otherwise.

Strategies: Next, let us outline the strategy space for both the `end-user` and the `system`.

- `end-user`: We define strategies only for the malicious `end-user` (type m), as they are the ones incentivized to launch explanation-based MIAs. The malicious `end-user` uses a *randomized* strategy: at each time t, they either continue interacting with the `system` or stop querying to attack. Their strategy depends on the history of variance in the explanations provided by the `system`; hence it is a collection of $\mathcal{F}_t$ - adapted stopping times $\{\phi_t\}$ such that $\phi_t = \rho$ if $\rho \leq t$ and ∞ otherwise.
- `system`: We assume the `system` plays a pure stopping time (τ^t) strategy to block a malicious `end-user`. Continuous interaction is a default action to provide model predictions, and their explanations are implicit for the `system`. Its strategy depends on the evolution of the explanation variance process EX^v and the record of the `end-user`'s querying activity. Hence, the strategy space of the `system` is a collection of $\mathcal{F}_t^+$ - adapted stopping times $\{\tau^t\}$.

We use a *path-wise Cumulative Distribution Function (CDF)*, represented as $R_t^{t_0}$, to characterize how fast the computed variance at a given time t is trying to reach the variance threshold (defined later). We compute this CDF from the probability density function ($p_t(ex_{s_y,t}^v)$) of the GBM, given by:

$$p_t(ex_{s_y,t}^v) = \frac{1}{\sqrt{2\pi t}\sigma ex_{s_y,t}^v} \exp\left(-\frac{[ln(ex_{s_y,t}^v) - (\mu - \sigma^2)t]^2}{2\sigma^2 t}\right),$$

where, $ex_{s_y,t}^v \in (0, \infty)$. In other words, the CDF ($R_t^{t_0}$) will give the probability of how close the computed explanation variance is to the explanation variance threshold at time t starting from the explanation variance computed at time t_0, i.e., $ex_{s_y,0}^v$.

Beliefs: Given information $\mathcal{F}_t^+$, the `system` updates its beliefs at time t from time $t_0 < t$ using *Bayes' rule* shown below. It is defined as the ratio of the probability of the honest `end-user` sending queries to the `system` (set to 1) to the total probability of

honest `end-user` and malicious `end-user` sending queries to the `system`.

$$\pi_t = \begin{cases} \frac{1}{1+(1-\pi_{t_0})R_t^{t_0}}, & \text{if} \quad \pi_{t_0} > 0 \quad \text{and} \quad \rho > t. \quad \textbf{(i)} \\ 0, & \text{if} \quad \rho \leq t \quad \text{or} \quad \pi_{t_0} = 0. \quad \textbf{(ii)} \end{cases}$$

Bayes' rule (i) is used when the `end-user` has not stopped communicating with the `system` ($\rho \geq t$) and the initial belief of the `system` about the `end-user`'s type is also not zero. However, if the `system` has already identified the `end-user`'s type as m or the `end-user` has already stopped communicating with the `system` and gets detected by it, then `system`'s belief π_t will be zero, as indicated in (ii).

Table 1: Flow Payoff Coefficients

	Before Detection		After Detection
	Pooling	Separation Starts	Detection and Block (Game Ends)
`end-user`, type m	P	M_{NS}^m	$-k$
`end-user`, type h	P	P	P
`system`	r_e	D_{NS}^{Θ}	D_B^{Θ}

Payoffs: Table 1 summarizes the flow payoff coefficients assumed in our game model. The `system` earns a reward of $D_B^{\Theta=m} = kEX_t^v$ for detecting and blocking the malicious `end-user`. The `end-user`'s type (malicious) is immediately revealed at this time, thus a cost of $-k$ is incurred by the `end-user`. In case of an interaction with an honest `end-user`, the `system` will always earn a payoff of $r_eEX_t^v$, i.e., $D_{NS}^{\Theta=h} = r_eEX_t^v$ and $D_B^{\Theta=h} = r_eEX_t^v$, while the honest `end-user` always earns a reward of PEX_t^v in each stage of the game. In case of a malicious `end-user` who keeps communicating with the `system` without being detected i.e., pools with the honest `end-user`, he/she receives a payoff (relevant explanation variance information) of PEX_t^v. Prior to detection, if the malicious `end-user` stops and is able to compromise the `system`, then the `system` will have to pay a lump-sum cost of $D_{NS}^{\Theta=m} = -d^{'}$ and the malicious `end-user` will earn $M_{NS}^m = (M^m + d^{'})EX_t^v$, where $M^mEX_t^v$ is the gain which relates to the explanation variance information gained from the `system`, $d^{'}EX_t^v$ is the benefit (can be monetary) achieved after attacking the `system`. Malicious `end-user` will also incur cost of deviation d. We make the following assumptions about the payoff coefficients: We assume that $D_B^{\Theta=m} = kEX_t^v > r_eEX_t^v$, as the `system` will gain more in successfully preventing the attack from the malicious `end-user`. When the malicious `end-user` decides to stop and attack the `system` and is not successful in compromising the `system`, then $PEX_t^v \geq M^mEX_t^v$ ($d^{'} = 0$) as the `system` has not yet blocked the malicious `end-user` and because of the cost of deviation.

3.3 Equilibrium Description

A *Markov Perfect Equilibrium (MPE)* consists of a strategy profile and a state process (EX^v, π) such that the malicious `end-user` and the `system` are acting optimally, and π_t is consistent with Bayes' rule whenever possible (in addition to the requirement that strategies be Markovian). A unique MPE occurs when the two types of `end-users` display *pooling* behavior. Given this equilibrium concept, our main

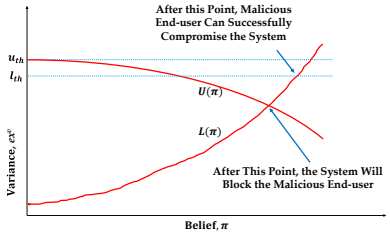


Fig. 1: Illustration of a continuous path analysis of $U(\pi)$ and $L(\pi)$ in Markov Perfect Equilibrium

result is the characterization of querying activity of the malicious `end-user` and detection (stopping) strategies by the `system` in a unique equilibrium. We assume that a decision to stop querying (i.e., deviating from honest behavior) is the last action in the game taken by the (malicious) `end-user`. This decision allows the `end-user` to either achieve the target of compromising the `system` and then getting blocked by it or getting blocked without reaching this target at all. In either case, the `system`'s belief about this `end-user` will jump to $\pi_t = 0$. The `end-user` has no further action, and the game reduces to a straightforward *stopping problem* for the `system` i.e., the `system` decides when to stop the game. In that case, the continuation payoffs from that point on can be interpreted as the termination payoffs of the original signaling game.

Next, consider the state of the game before the `end-user` deviates/reveals and before the `system`'s block action. Since malicious `end-user` plays a mixed strategy that occurs *on-equilibrium path*, `system`'s belief about the `end-user` evolves over time. Thus, a unique MPE consists of a state variable process (EX^v_t, π_t) and two cutoff functions, a non-increasing variance function $U(\pi_t)$ for the `system` and an increasing variance function $L(\pi_t)$ for the `end-user`, where

- The `system` immediately blocks the `end-user` if $ex^v_{s_y,t} \geq U(\pi_t)$, i.e., $\tau = inf\{t >= 0 : ex^v_{s_y,t} \geq U(\pi_t)\}$.
- The malicious `end-user` keeps querying for explanation (thus, its variance), whenever $ex^v_{e_u,t} < L(\pi_t)$ and mixes between querying and not querying whenever $ex^v_{e_u,t} \geq L(\pi_t)$, so that the curve $\{(L(\pi), \pi) : \pi \in [0, 1]\}$ serves as a *reflecting boundary* for the process $(ex^v_{e_u,t}, \pi_t)$.

We call such a unique MPE equilibrium a (U, L) *equilibrium*. The first condition defines an upper boundary which tells the `system` that if an explanation variance value $ex^v_{s_y,t}$ at time t (corresponding to a query sent by the `end-user`) is greater than or equal to this boundary ($U(\pi_t)$), then the `end-user` is trying to compromise the `system`. In this case, the `system` should block the `end-user`. The second condition above guides the behavior of the malicious `end-user`. Function $L(\pi_t)$ represents the upper-bound of the target explanation variance value the malicious `end-user` wants to achieve given a certain belief π_t at time t. When the explanation variance value corresponding to a query by an `end-user` is less than this boundary function, i.e., $ex^v_{e_u,t} < L(\pi_t)$, then it is strategically better for the malicious `end-user` to keep querying (i.e., looks honest from `system's` perspective). However, if $ex^v_{e_u,t} \geq L(\pi_t)$ then the malicious `end-user` has an incentive to stop querying. For the malicious

end-user, this condition also represents that it is near to the desired (or target) variance threshold value - one more step by the malicious end-user can either lead to success (compromise of the system) or failure (getting blocked by the system before achieving its goal).

To understand the MPE structure, consider the current belief, π_t. If the computed explanation variance is close to the threshold, the system should block the end-user suspicious of moving toward the model's classification boundary. This cutoff for the variance is a non-increasing function of π because, by definition, the end-user is less likely to be honest when the variance value is sufficiently close to the threshold value and π_t is large. Thus, when the threshold value becomes greater than or equal to the optimal function $U(\pi_t)$ at any time t, system will block the end-user. This is intuitively shown in Figure 1, where u_{th} represents the variance threshold for an explanation generated by the system and l_{th} represents the variance threshold for the explanation after the textttsystem adds noise based on its belief, which can be given to the end-user. [0, u_{th}] or [0, l_{th}] represents the pooling region where an MPE can occur, if end-user is not blocked by the system.

4 Equilibrium Analysis

Next, we try to analyze conditions under which a unique MPE exists in the game described above, i.e., we try to construct a (U, L) equilibrium by finding conditions under which optimal functions $L(\pi_t)$ and $U(\pi_t)$ exist.

High-level Idea: As mentioned in section 3, an MPE is defined as a pair of functions $L(\pi_t)$ and $U(\pi_t)$. Thus, we first need to show that these two optimal functions exists. To prove that $L(\pi_t)$ and $U(\pi_t)$ exist, we prove the continuity and differentiability properties of $L(\pi_t)$ (theorem 1) and $U(\pi_t)$ (theorem 2) in the belief domain ($\pi_t \in [0, 1]$). As we have assumed that the system plays a pure strategy, we consider that $U(\pi_t)$ (for the system) is optimal, and show that it exists and is continuous. However, computing an optimal $L(\pi_t)$ is non-trivial, as we have assumed that the end-user plays a randomized strategy. Therefore, to compute $L(\pi_t)$, we first construct two bounding functions $L^+(\pi_t)$ (upper) and $L^-(\pi_t)$ (lower) and show that such functions exists (in lemmas 3 and 4, respectively). To compute these functions we use the boundary conditions (section 2.4) at the decision boundaries, i.e., Pooling, Separating, and Detection and Block (outlined in Table 1) and the value functions defined below. We then show that as π_t increases, $L^+(\pi_t)$ and and $L^-(\pi_t)$ will converge in the range $(0, u_{th})$ (lemma 1) or $(0, l_{th})$ (lemma 2) as we have considered that an MPE occurs when both types of end-user pool. Finally, we show that the first intersection point (root) of $L^+(\pi_t)$ and $L^-(\pi_t)$ is a unique MPE, if the end-user is not blocked by the system before that point. At these decision boundaries, both players decide to either continue or stop.

4.1 Value Functions

As mentioned earlier, we are considering an equilibrium that occurs when both types of end-user pool, i.e., $ex^v_{e_u,t}$ is strictly below $L(\pi_t)$. In this pooling scenario, no information becomes available to the system about the type of the end-user; thus, belief

(π_t) remains constant. Hence, we write the value functions for both the players conditioned on no deviation by the `end-user`. As we consider an infinite horizon in our game, there is no known terminal (final) value function. Hence, these value functions are independent w.r.t. $t \in R$, as t singularly has no effect on them. The `end-user`'s value function (F) should solve the following HJB equation representing his/her risk-less return:

$$rF(ex^v_{e_u}, \pi) = \mu ex^v_{e_u} F^{'}_{ex^v_{e_u}}(ex^v_{e_u}, \pi) + \frac{1}{2}\sigma^2 (ex^v_{e_u})^2 F^{''}_{ex^v_{e_u}}(ex^v_{e_u}, \pi) + \psi ex^v_{e_u}$$

where $F^{'}_{ex^v_{e_u}}$ and $F^{''}_{ex^v_{e_u}}$ are the first and second order partial derivative of the value function $F(ex^v_{e_u}, \pi)$ w.r.t. $ex^v_{e_u}$, respectively, and, μ and σ are the drift/mean and the variance/volatility of the variance process EX^v_t, respectively. ψ is the payoff coefficient which depends on the stage payoffs of the `end-user`, as represented in Table 1. The solution to the above equation can be represented as:

$$F(ex^v_{e_u}, \pi) = A_1(\pi)(ex^v_{e_u})^{\beta_1} + A_2(\pi)(ex^v_{e_u})^{\beta_2} + \frac{\psi ex^v_{e_u}}{r - \mu}$$

for some constants $A_1(\pi)$ and $A_2(\pi)$, where $\beta_1 > 1$ and $\beta_2 < 0$ are the roots of the characteristic equation [9]. Similarly, the `system`'s value function $V(ex^v_{s_y}, \pi)$ should satisfy the following equation, conditioned on $ex^v_{e_u} < L(\pi)$ and π staying constant:

$$rV(ex^v_{s_y}, \pi) = \mu ex^v_{s_y} V^{'}_{ex^v_{s_y}}(ex^v_{s_y}, \pi) + \frac{1}{2}\sigma^2 (ex^v_{s_y})^2 V^{''}_{ex^v_{s_y}}(ex^v_{s_y}, \pi) + \psi ex^v_{s_y}$$

where $V^{'}_{ex^v_{s_y}}$ and $V^{''}_{ex^v_{s_y}}$ are the first and second order partial derivative of the value function $V(ex^v_{s_y}, \pi)$ w.r.t. $ex^v_{s_y}$, respectively. As before, ψ is the payoff coefficient which depends on the stage payoffs of the `system` as shown in Table 1. The solution to the above equation can be represented as:

$$V(ex^v_{s_y}, \pi) = B_1(\pi) \times (ex^v_{s_y})^{\beta_1} + B_2(\pi)(ex^v_{s_y})^{\beta_2} + \frac{\psi ex^v_{s_y}}{r - \mu}$$

for some constant $B_1(\pi)$ and $B_2(\pi)$. We will use different boundary conditions to determine $A_1(\pi)$, $A_2(\pi)$, $B_1(\pi)$ and $B_2(\pi)$. Then, we will use these conditions to determine $U(\pi_t)$ and $L(\pi_t)$.

4.2 Analytical Results

We aim to compute the `system`'s threshold u_{th} (Lemma 1) and the `end-user`'s threshold l_{th} (Lemma 2). These thresholds define the region for a potential MPE in the game if the necessary conditions are satisfied. These lemmas help determine if a unique MPE exists. Due to space constraints, we could not add all the proofs here. Thus, please find proofs and other details in the Appendix of the extended paper version [21].

Lemma 1. *There exists a positive upper bound u_{th} on the variance of an explanation generated by an explanation method representing the maximum variance value that can be reached for the query sent by the* `end-user`.

Lemma 2. *There exists a positive upper bound l_{th} on the variance of an explanation given by the* `system` *to the* `end-user` *representing maximum variance value needed to be reached by the* `end-user` *to compromise the* `system`*.*

We aim to characterize the optimal cutoff functions: the `system`'s $U(\pi_t)$ and the `end-user`'s $L(\pi_t)$. These functions represent the game's MPE and help the `system` and the `end-user` to play optimally in each game stage. For example, if the `system` doesn't have any knowledge of $U(\pi_t)$, then it won't know the range of the variance values being computed for the explanations, which are given to the `end-user` after adding some noise based on its belief. Hence, an adversary can easily compromise the `system`. In contrast, $L(\pi_t)$ function knowledge will guide an adversary on how to compromise the `system` optimally. For that reason, we first prove that $U(\pi_t)$ exists and is non-increasing and continuously differentiable (Theorem 1[1]).

Theorem 1. *$U(\pi_t)$ is non-increasing and continuously differentiable function in domain $[0,1]$ if and only if either $\beta_2\beta_1 J^{`}(\pi,t)^{\beta_2-1} \leq \beta_1\beta_2 J^{`}(\pi,t)^{\beta_1-1}$ or $\beta_2(\beta_1 - 1)J^{`}(\pi,t)^{\beta_2-1} \leq \beta_1(\beta_2-1)J^{`}(\pi,t)^{\beta_1-1}$, where $J(\pi,t) = \frac{L(\pi)}{U(\pi)}$.*

To prove $L(\pi)$ (Theorem 2) exists and is increasing and continuously differentiable, we first characterize an explanation variance path $L^+(\pi)$ (Lemmas 3), which represents the maximum variance values that can be computed by the `end-user` for the given explanations, and a variance path $L^-(\pi)$ (Lemma 4), which represents the minimum variance values for the explanations given by the `system` to the `end-user`. We write three equations each for $L^+(\pi)$ and $L^-(z)$ according to the value matching, smooth pasting, and the condition in which the variance of the explanation received is opposite of what `end-user` expected. Then, we demonstrate that both these functions are increasing and continuously differentiable. The purpose for doing this is to use these lemmas to show that as $\pi \to 1$, both $L^+(\pi)$ and $L^-(\pi)$ starts to converge and becomes equal to $L(\pi)$ after some point.

Lemma 3. *$L^+(\pi)$ is a well-defined, increasing, continuous and differentiable function in domain $[0,1]$ if and only if $\lambda^{'}(L^+(\pi),\pi) > 0$ and $P > 0$, where $\lambda()$ is the termination payoff if the* `end-user` *decides to deviate and attack the* `system`*.*

Lemma 4. *$L^-(\pi))$ is a well-defined, increasing, continuous and differentiable function in domain $[0,1]$ if and only if either $(\frac{\partial A_1^+(z)}{\partial \pi}L^-(\pi)^{\beta_1} + \frac{\partial A_2^+(\pi)}{\partial \pi}L^-(\pi)^{\beta_2}) < 0$ or $(A_1^+(\pi)\beta_1 L^-(\pi)^{\beta_1-1} + A_2^+(\pi)\beta_2 L^-(\pi)^{\beta_2-1}) < 0$.*

Theorem 2. *$L(\pi)$ is a well-defined, increasing, continuous and differentiable function domain $[0,1]$ if and only if either $\lambda^{'}(L(\pi),\pi) > 0$ and $P > 0$.*

Finally, we show that such a point where $L^+(\pi_t)$ and $L^-(\pi_t)$ converge (or intersect) exists, and thus, a unique MPE (Theorem 3) exists in the game.

Theorem 3. *A unique MPE or a point, $\varsigma = \frac{\lambda(L^+(\pi),\pi)\times(r-\mu)}{P\times L^-(\pi)}$, exists in the game where the two curves $L^+(\pi)$ and $L^-(\pi)$ starts to converge, if and only if $\frac{\beta_2}{\varsigma^{\beta_2+1}}\times \left[L^+(\lambda^{'} - \frac{P}{r-\mu}) - \beta_1(\lambda - \frac{PL^+}{r-\mu})\right] \geq \frac{\beta_1}{\varsigma^{\beta_1+1}}\left[\beta_2(\lambda - \frac{PL^+}{r-\mu}) - L^+(\lambda^{'} - \frac{P}{r-\mu})\right]$.*

[1] To simplify the exposition of the proofs, we have replaced $L(\pi_t)$ with $L(\pi)$ and $U(\pi_t)$ with $U(\pi)$ in these results.

Table 2: Dataset Configurations.

Datasets	Points	#Features	Type	#Classes
Purchase	197,324	600	Binary	100
Texas	67,330	6,170	Binary	100
CIFAR-100	60,000	3,072	Image	100
CIFAR-10	60,000	3,072	Image	2
Adult	48,842	24	Mixed	2

5 Experimental Setup

We use the *Captum* [20] framework to generate four explanation types: *GradientInput*, *Integrated Gradients*, *LRP*, and *Guided Backpropagation*. Next, we use *PyTorch* framework to conduct the training and attack-related experiments. *GradientInput* serves as our baseline to compare the results of the other explanation methods. We assume that when the game ends, both the `system` and the `end-user` will have access to their optimal strategies, u_{th} and l_{th}, respectively. Thus, when the game ends, an adversary can use its optimal strategy and optimal threshold to conduct MIA, or a `system` can use its optimal strategy and optimal threshold to protect against MIA. As a result, we focus on two evaluation objectives in our experiments: (i) *game evolution*, and (ii) *MIA accuracy*. For the game evolution, we simulate and generate the future explanation variances for $t = 100$ stages, according to the expression:

$$EX_t^v = EX_0^v * e^{\left((\mu - \frac{1}{2}\sigma^2) + \sigma W_t\right)} \tag{1}$$

The above equation is the solution to the GBM (Equation 1) of EX_t^v, derived using the *itô's* calculus [9]. μ and $\sigma > 0$ are computed using the variance generated for the test datapoints for each of the dataset. In our experiments, we take EX_0^v as the last index value of the test data points' generated explanation variance, as we use this initial value to generate future explanations. Using the obtained optimal strategies and thresholds, we compute the attack accuracy in terms of the attacker's success rate in launching the MIA or the accuracy of the `system` in preventing the MIA.

Datasets. We use five popular benchmark datasets on which we perform our game analysis and attack accuracy evaluations: Purchase and Texas datasets [25], CIFAR-10 and CIFAR-100 [30], and the Adult Census dataset [11]. To ease the comparison, the setup and Neural Network (NN) architectures are aligned with existing work on explanation-based threshold attacks [31]. Table 2 details each dataset's configuration. One can refer to the extended version of the paper for more details [21].

Evaluation metric. We compute the *True Positive Rate (TPR)* to estimate MIA accuracy after the game ends, with each player having formulated their best response strategy. TPR measures how accurately an attacker infers data point membership. We consider training data points to test against the optimal strategy of the `system`. Since the sample space that we have considered contains only actual training members, there can be only two outcomes: correctly classified and incorrectly classified. The total number of training data points correctly inferred as training points (using u_{th}) are called True Positives (TP), while the number of training members discerned as non-training members are called False Negatives (FN). Thus, $TPR = \frac{TP}{TP+FN}$.

6 Evaluation

This section analyzes our game model to assess two objectives: (i) the equilibrium evolution for optimal strategies and (ii) attack accuracy (TPR). Next, we consider the *Gradient*Input* method as our baseline for comparison against other explanation methods. Finally, we also analyze how other factors can influence attack accuracy.

6.1 Impact of Different Attack Information Sources

As detailed in Section 5, we initially sample future explanations for each dataset using GBM (Equation 1). The sampled noise is added to the generated explanations variance based on the computed belief π_t, such that higher belief implies honest user, thus smaller noise added to the explanation variance, and vice-versa.

Then, we compute different functional paths for the `system` and the `end-user` (Section 3.3) i.e., we compute $U(\pi_t)$, $L^+(\pi_t)$, $L^-(\pi_t)$ and $L(\pi_t)$ functions. The termination payoff, $\lambda(ex^v_{e_u}, \pi_t)$ (defined in 2.4), which is used to write the boundary conditions in the computation of $L^+(\pi_t)$, $L^-(\pi_t)$ and $L(\pi_t)$ (Lemma 3 and Lemma 4, and Theorem 2) is assumed to be:

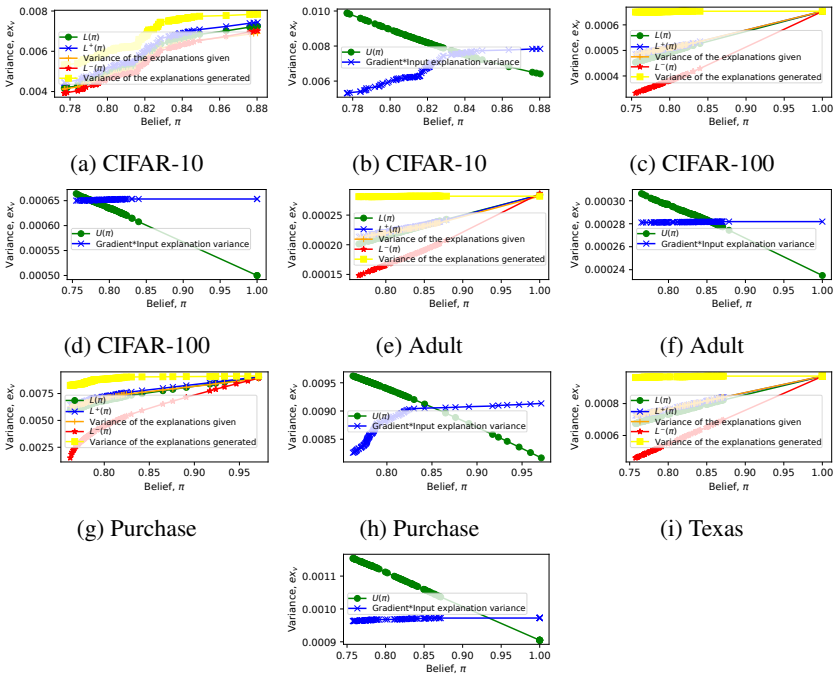
$$\lambda(ex^v_{e_u}, \pi_t) = \frac{0.8 \times ex^v_{e_u} \times log(\pi_t \times 2) + \pi_t \times ex^v_{e_u}}{b}$$

Where $ex^v_{e_u}$ is the value of any `end-user`'s functional path (considered for the specific computation) at time t, and b is the model parameter set differently for each explanation method. The parameters for $\lambda(ex^veu, \pi_t)$ are empirically chosen based on their suitability to each of the four explanation methods. From our numerical simulations, we observe important patterns for each dataset in the baseline setting (*Gradient*Input*) and the other three gradient-based explanation techniques.

Game Evolution in the Baseline Setting: Figure 2 represents varying game evolution realized for different datasets. Below, we analyze in detail the optimal paths obtained for each dataset.

- From the plots of the optimal functional path $U(\pi_t)$ of the `system` for each of the dataset, as shown in Figures 2b, 2d, 2f, 2h, and 2j, we can observe that as $\pi_t \rightarrow 1$, $U(\pi_t)$ starts decreasing. This is because, as the `system`'s belief about the type of `end-user` approaches 1, both the variance of the explanation generated by the system and the variance of the noisy explanation given to the `end-user` approach u_{th} and l_{th}, respectively. After a certain point, i.e., when $ex^v_{s_y} > U(\pi_t)$, the `system` will block the `end-user`, which confirms to our intuition.
- From the optimal functional paths $L^+(\pi_t)$, $L^-(\pi_t)$ and $L(\pi_t)$ of the `end-user` for each of the dataset, as shown in Figures 2a, 2c, 2e, 2g, and 2i, we can observe that as $\pi_t \rightarrow 1$, $L^+(\pi_t)$, $L^-(\pi_t)$ and $L(\pi_t)$ approach the threshold l_{th}. As discussed in 3.3, as $\pi_t \rightarrow 1$ and the variance of the explanation given to the `end-user` starts to approach the variance threshold, it means a malicious `end-user` is trying to compromise the `system`. Thus, if the `system` doesn't block the `end-user` at the right time (or doesn't have knowledge about optimal $U(\pi)$), then the `end-user` can easily compromise the `system`.

– Earlier we showed that a unique MPE exists when $L^+(\pi_t)$ and $L^-(\pi_t)$ begin to converge as $\pi_t \to 1$. This is also visible from our results as shown in Figure 2, where we can observe that as $\pi_t \to 1$, $L^+(\pi_t)$ and $L^-(\pi_t)$ starts to converge. However, for the CIFAR-10 dataset, one can observe that the curves $L^+(\pi_t)$ and $L^-(\pi_t)$ doesn't converge as $\pi_t \to 1$. Thus, an MPE doesn't exist in the case of CIFAR-10 dataset. The intuition behind this observation is that the fluctuations (or variance) of the explanation variance computed for the CIFAR-10 is high, making it difficult for them to converge to a single point. Finally, if the `system` doesn't block the `end-user` before the threshold l_{th} or u_{th} is reached, then we say a unique MPE exists in the game.



(a) CIFAR-10 (b) CIFAR-10 (c) CIFAR-100

(d) CIFAR-100 (e) Adult (f) Adult

(g) Purchase (h) Purchase (i) Texas

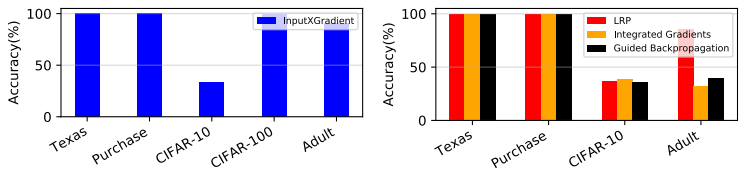
(j) Texas

Fig. 2: Different functional paths for the different datasets. (a), (c), (e), (g), and (i) represents the optimal functional paths for the `end-user`. (b), (d), (f), (h), and (j) represents the optimal functional paths for the `system`.

Attack Accuracy in the Baseline Setting: After obtaining the optimal strategies, we use the range of the training data points of each dataset to determine how many data point variances lie below the computed threshold u_{th} to determine their membership. As shown in Figure 3a, the attack accuracy for all the datasets except CIFAR-10 is more than 50%. This result aligns with the observed game equilibrium analysis. Hence, the fluctuations in explanation variance make it difficult for an adversary to reach the target

threshold, in consequence, to launch MIAs. From these obtained results, one can easily observe that the explanations provide a new opportunity or an attack vector to an adversary actively trying to compromise the `system`. In other words, our results are clear indicators that an adversary can repeatedly interact with the `system` to compute the explanation variance threshold and successfully launch membership inference attacks against the `system`.

Results for other Explanation Techniques: We also analyzed the game for the three other explanation methods considered in this paper. We do not plot the game evolution results in this setting as the plots follow a very similar trend as seen in Figure 2, i.e., game equilibrium was achieved for all the datasets except for the CIFAR-10. We uses the same setting as the baseline setting (mentioned above) to compute attack accuracy for these three explanation methods. We obtained each dataset's attack accuracy as shown in Figure 3b. For the Texas and Purchase datasets, 100% accuracy was achieved, i.e., an attacker effectively determines the membership of all the data points used for training the model. However, for the CIFAR-10, the attack accuracy was below 50%, and for the Adult dataset, attack accuracy was above 50% only for the LRP explanation method. The reason is again the high fluctuations in the computed variance for the CIFAR-10 dataset (slightly less for the Adult dataset), thus making it difficult for an adversary to determine the membership of the data points in those datasets. These results clearly indicate that, for different explanation methods, an adversary's capability to launch MIA attacks will vary and may depend on the variance of the explanations.



(a) (b)

Fig. 3: Accuracy (TPR) for the optimal strategy obtained by the `system` and the `end-user`: a) *Gradient*Input* method and b) Other explanation methods.

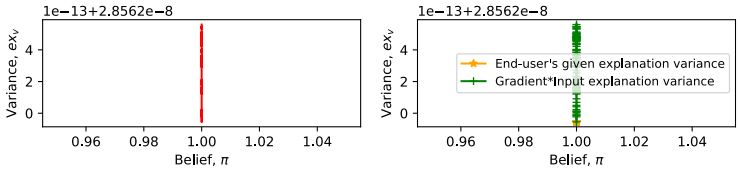
6.2 Analysis of other Relevant Factors

This section examines factors like input dimension, underfitting, and overfitting using synthetic datasets to study how adversaries can exploit model gradient information for membership inference attacks.

– **Impact of Input dimension.** First, we analyze the impact of input dimension on game evolution using the Sklearn make classification module [27] to generate datasets. We set the number of classes to 2 or 100 and vary the number of features from $t_f \in [10, 100, 1000, 6000]$. We sample 20,000 points for each setting and split them evenly into training and test sets. Second, for each value t_f and for each class, we employ two models to train from this data: model A and model B. Model A is chosen to

have fewer layers (or depth) than model B to compare the effect of the complexity of the models on the game evolution and attack accuracy. Model A is a fully connected NN with two hidden layers fifty nodes each, the *tanh* activation function between the layers, and *softmax* as the final activation. The network is trained using *Adagrad* with a learning rate of 0.01 and a learning rate decay of 10^{-7} for 50 epochs. Model B is a five-layer fully connected NN with *tanh* activations. The layer sizes are 2048, 1024, 512, 256 and 100. We use the *Adagrad* optimizer with a learning rate of 0.01 and a learning rate decay of 10^{-7} to train the model for 50 epochs. Next, we demonstrate the effect of these models on our experiment's two main objectives.

- **Effect of Model A on Game Evolution and Attack Accuracy.** For $k = 2$ classes, we observe a similar trend in the game evolution, shown in Figure 2 for each of the features $t_f \in [10, 100, 1000, 6000]$. However, for $k = 100$ classes, we observed that the belief π_t of the `system` about the type of the `end-user` is always set to 1 as shown in Figure 4a. Consequently, the variance of the explanations generated is equivalent to that of the explanations given (Figure 4b). Hence, the game didn't evolve as the `system` explained the same to the `end-user`. The reason is because of underfitting. Model A lacks sufficient depth (fewer layers) to classify 100 classes accurately, resulting in poor performance. Moreover, the final model's loss was 5.74 for all features, leading to inaccurate predictions and affecting the experimental objectives.

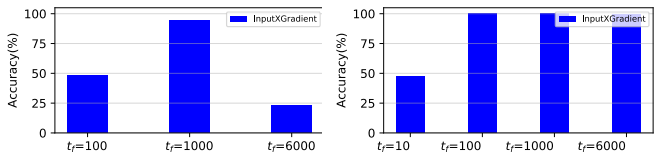


(a) (b)

Fig. 4: Explanation variance generated vs. explanation variance given when $\pi_t = 1$.

- **Effect of Model B on Game Evolution and Attack Accuracy.** Next, we analyze the game results model B, which incurred a training loss of 0.8 across all features. No game evolution was observed for $k = 2$ classes and $n_f = 10$ because we got $\sigma > \mu$ and $\sigma > 1$ for the test data points explanation variance. Consequently, the computed future variance values were zero using Equation 1. For $t_f \in [100, 1000, 6000]$, we analyzed the game equilibrium and computed the attack accuracy using sampled training data points, depicted in Figure 5a. For $t_f = 1000$, an attack accuracy greater than 50% was observed; however, for $t_f = 100$ and $t_f = 6000$, an attack accuracy less than 50% was observed. For $k = 100$ classes, we did not observe any equilibrium for any of the features t_f. Based on the final simulated explanation variance index (at $t = 100$), we computed the threshold u_{th} and determined the attack accuracy for each feature (Figure 5b).

The results for models A and B show that the model choice significantly influences the game evolution and affects an adversary's capability to launch MIA attacks against the `system`.



(a) $k = 2$. (b) $k = 100$.

Fig. 5: MIA accuracy for different features n_f for model B.

– **Impact of Overfitting.** As detailed in [39], overfitting significantly boosts membership inference attack accuracy. To examine its impact, we varied training epochs for Purchase, Texas, and Adult in $\{30, 50, 60\}$ datasets. Overfitting increased attack accuracy only for the Adult dataset, which remained unchanged for Texas and decreased for Purchase. Hence, the game's evolution and MIA accuracy hinge on multiple factors (experimented above), not just on training epochs. Thus, overfitting alone does not uniformly enhance attack capability, as shown in the aforementioned scenarios.

7 Related Works

Efforts to enhance ML model transparency present privacy risks, as shown in existing works where explanations are exploited for various attacks, such as MIA [31], model reconstruction [24], model inversion [40], and sensitive attribute inference [12] attacks. We focus on MIA, where high explanation variance indicates either exclusion from training data or model uncertainty, enabling potential attacks [31]. Unlike prior work analyzing the single "what if" interaction scenario, our study models repeated interactions between the `system` and malicious `end-user`, examining varied settings' impacts on MIA using optimal strategies and thresholds.

Game-theoretic approaches, such as zero-sum games [8] [13], non-zero sum games [10], sequential Bayesian games [41] [14], sequential Stackelberg games [5] [1] and simultaneous games [7] have been used in the research literature to model interactions with ML models, specifically to model adversarial classification. Contrary to these efforts, where an adversary's objective is to target the classification task of an ML model, our research effort focuses on the descriptive task, i.e., explaining the model predictions. Specifically, we use a continuous-time stochastic Signaling Game [26] [4] [2] [36] to model the repeated interactions in a dynamic ML system with explanations to accomplish MIAs. We also make a novel use of GBM [28] [17] [9] to model the explanation variance in order to analyze how an adversary can utilize historical variance information to reach the target variance threshold. To the best of our knowledge, there have been no prior works that utilize a continuous-time game-theoretic formulation to study the privacy leakages (in the form of MIAs) due to model explanations. Similar continuous-time stochastic signaling game models have been used in economic theory to study stock prices [9], dynamic limit pricing [15] [16], and market trading [6]. Our work is one of the first to use modeling concepts from economic theory to study the privacy problem in the ML and model explainability domain.

8 Conclusion

We modeled the strategic interactions between an `end-user` and a `system`, where the variance of the explanations generated by the `system` evolve according to a stochastic differential equation, as a two-player continuous-time signaling game. Our main aim was to study how an adversary computes the optimal variance threshold to launch explanation-based MIAs. Further, our experiments showed that an adversary's ability to launch MIA depends on various factors. A knowledgeable adversary can exploit these factors, particularly the variance in explanations, to effectively conduct MIA.

References

1. Alfeld, S., Zhu, X., Barford, P.: Explicit defense actions against test-set attacks. In: AAAI (2017)
2. Averboukh, Y.: Approximate solutions of continuous-time stochastic games. SIAM Journal on Control and Optimization (2016)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS One (2015)
4. Brázdil, T., Forejt, V., Krcal, J., Kretinsky, J., Kucera, A.: Continuous-time stochastic games with time-bounded reachability. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2009)
5. Brückner, M., Scheffer, T.: Stackelberg games for adversarial prediction problems. In: ACM KDD (2011)
6. Daley, B., Green, B.: Waiting for news in the market for lemons. Econometrica (2012)
7. Dalvi, N., Domingos, P., Sanghai, S., Verma, D.: Adversarial classification. In: ACM KDD (2004)
8. Dekel, O., Shamir, O., Xiao, L.: Learning to classify with missing and corrupted features. Machine learning (2010)
9. Dixit, R.K., Pindyck, R.S.: Investment under uncertainty. Princeton university press (2012)
10. Dritsoula, L., Loiseau, P., Musacchio, J.: A game-theoretic analysis of adversarial classification. IEEE Transactions on Information Forensics and Security (2017)
11. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
12. Duddu, V., Boutet, A.: Inferring sensitive attributes from model explanations. arXiv preprint arXiv:2208.09967 (2022)
13. Globerson, A., Roweis, S.: Nightmare at test time: robust learning by feature deletion. In: ICML (2006)
14. Großhans, M., Sawade, C., Brückner, M., Scheffer, T.: Bayesian games for adversarial regression problems. In: ICML (2013)
15. Gryglewicz, S.: Signaling in a stochastic environment and dynamic limit pricing. Tech. rep., mimeo, Tilburg University (2009)
16. Gryglewicz, S., Kolb, A.: Strategic pricing in volatile markets. Kelley School of Business Research Paper (2019)
17. Hu, Y., Øksendal, B.: Optimal time to invest when the price processes are geometric brownian motions. Finance and Stochastics (1998)
18. Hui, B., Yang, Y., Yuan, H., Burlina, P., Gong, N.Z., Cao, Y.: Practical blind membership inference attack via differential comparisons. arXiv preprint arXiv:2101.01341 (2021)

19. Ignatiev, A., Narodytska, N., Marques-Silva, J.: On relating explanations and adversarial examples. NeurIPS (2019)
20. Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., et al.: Captum: A unified and generic model interpretability library for pytorch. arXiv preprint arXiv:2009.07896 (2020)
21. Kumari, K., Jadliwala, M., Jha, S.K., Maiti, A.: Towards a game-theoretic understanding of explanation-based membership inference attacks. arXiv preprint arXiv:2404.07139 (2024)
22. Long, Y., Bindschaedler, V., Gunter, C.A.: Towards measuring membership privacy. arXiv preprint arXiv:1712.09136 (2017)
23. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: NeurIPS (2017)
24. Milli, S., Schmidt, L., Dragan, A.D., Hardt, M.: Model reconstruction from model explanations. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. pp. 1–9 (2019)
25. Nasr, M., Shokri, R., Houmansadr, A.: Machine learning with membership privacy using adversarial regularization. In: CCS (2018)
26. Neyman, A.: Continuous-time stochastic games. Games and Economic Behavior (2017)
27. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research (2011)
28. Reddy, K., Clinton, V.: Simulating stock prices using geometric brownian motion: Evidence from australian companies. Australasian Accounting, Business and Finance Journal (2016)
29. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you? explaining the predictions of any classifier. In: ACM KDD (2016)
30. Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., Jégou, H.: White-box vs black-box: Bayes optimal strategies for membership inference. In: ICML (2019)
31. Shokri, R., Strobel, M., Zick, Y.: On the privacy risks of model explanations. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. pp. 231–241 (2021)
32. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: IEEE S&P (2017)
33. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: ICML (2017)
34. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713 (2016)
35. Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H.: Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. pp. 180–186 (2020)
36. Sobel, J.: Signaling games. Complex Social and Behavioral Systems: Game Theory and Agent-Based Models pp. 251–268 (2020)
37. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)
38. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: ICML (2017)
39. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. In: CSF (2018)
40. Zhao, X., Zhang, W., Xiao, X., Lim, B.: Exploiting explanations for model inversion attacks. In: IEEE/CVF ICCV (2021)
41. Zhou, Y., Kantarcioglu, M.: Adversarial learning with bayesian hierarchical mixtures of experts. In: ICDM (2014)