Kallas *et al.*
EURASIP Journal on Information Security (2024) 2024:32
https://doi.org/10.1186/s13635-024-00180-5

EURASIP Journal on Information Security

RESEARCH

Open Access

Strategic safeguarding: A game theoretic approach for analyzing attacker-defender behavior in DNN backdoors

Kassem Kallas[1*], Quentin Le Roux[2,3*], Wassim Hamidouche[4] and Teddy Furon[3]

Abstract

Deep neural networks (DNNs) are fundamental to modern applications like face recognition and autonomous driving. However, their security is a significant concern due to various integrity risks, such as backdoor attacks. In these attacks, compromised training data introduce malicious behaviors into the DNN, which can be exploited during inference or deployment. This paper presents a novel game-theoretic approach to model the interactions between an attacker and a defender in the context of a DNN backdoor attack. The contribution of this approach is multifaceted. First, it models the interaction between the attacker and the defender using a game-theoretic framework. Second, it designs a utility function that captures the objectives of both parties, integrating clean data accuracy and attack success rate. Third, it reduces the game model to a two-player zero-sum game, allowing for the identification of Nash equilibrium points through linear programming and a thorough analysis of equilibrium strategies. Additionally, the framework provides varying levels of flexibility regarding the control afforded to each player, thereby representing a range of real-world scenarios. Through extensive numerical simulations, the paper demonstrates the validity of the proposed framework and identifies insightful equilibrium points that guide both players in following their optimal strategies under different assumptions. The results indicate that fully using attack or defense capabilities is not always the optimal strategy for either party. Instead, attackers must balance inducing errors and minimizing the information conveyed to the defender, while defenders should focus on minimizing attack risks while preserving benign sample performance. These findings underscore the effectiveness and versatility of the proposed approach, showcasing optimal strategies across different game scenarios and highlighting its potential to enhance DNN security against backdoor attacks.

Keywords Deep neural networks, CNN, Backdoor attacks, Backdoor defenses, Game theory, Zero-sum game, Bi-matrix game, Linear programming, Nash equilibrium

*Correspondence:
Kassem Kallas
kassem.kallas@inserm.fr
Quentin Le Roux
quentin.le-roux@thalesgroup.com
Full list of author information is available at the end of the article

1 Introduction and related works

1.1 Introduction

Over the past decade, deep neural networks (DNNs) have achieved significant success in critical applications such as computer vision [1], autonomous vehicles [2], finance [3], healthcare [4], and beyond [5]. However, the increasing prevalence of DNNs has raised concerns about their security. Well-studied threats, such as adversarial examples, compromise DNN integrity during inference by subtly manipulating test-time inputs [6]. Additionally, malicious actors may target the DNN training process itself. Each step, from data collection and pre-processing to architecture selection, training, and deployment, presents potential vulnerabilities that adversaries can exploit [7, 8]. Furthermore, the considerable data and computational resources required for DNN training, coupled with a shortage of machine learning expertise, often compel users to outsource aspects of their process to third parties (e.g., machine learning as a service (MLaaS), acquisition of pre-trained models [9]. This outsourcing, while convenient, reduces control and introduces new attack surfaces [6].

Backdoor attacks, a critical threat to DNN security, involve embedding malicious behavior into a DNN prior to inference. Once injected, the backdoor can be triggered by the attacker during inference to produce a desired, incorrect output [10–12]. The backdoor attacker's objective is twofold: the compromised DNN must function normally on benign inputs to avoid detection, and the backdoor must be easily activated. Activation typically occurs through the presentation of a trigger-modified input. Backdoor injection can occur at any stage in the model's supply chain and lifecycle before inference [6]. This includes poisoning training data [13], manipulating model parameters during training [14], or even during deployment [15–17]. Furthermore, transfer learning can also be exploited to embed a backdoor during training [18, 19].

The prevalent backdoor attack strategy currently relies on training data poisoning [13, 20]. This involves inserting samples, manipulated with a trigger pattern, into an otherwise benign dataset. The victim DNN then learns to associate the pattern with incorrect predictions. The labels of these poisoned samples may be altered [21, 22] (poison-label attacks) or remain consistent with their ground truths [20, 23] (clean-label attacks). The latter strategy aims to avoid detection if a defender inspects the training dataset.

Backdoor attacks have been demonstrated in various scenarios [6, 24], ranging from natural language processing (NLP) [25, 26] and audio [27] to computer vision applications [6, 28]. Beyond poison and clean-label attacks, backdoors encompass diverse subcategories, including class-agnostic or class-specific attacks [22], various trigger types and families [6], and concepts like trigger transparency [13, 29]. For comprehensive surveys on backdoor attacks, defenses, and their categorization, please refer to [6, 8, 10–12, 24].

The evolving threat landscape and the ongoing cat-and-mouse game between backdoor attackers and defenders [30], characterized by continuous development of new attacks and defenses, motivate this work. Within the specific context of clean-label backdoor attacks on image classification [6, 8, 20, 23], this paper asks the following question: can we model the interaction between a DNN backdoor attacker and a defender as a two-player game, determine its Nash equilibria, and assess each player's performance at equilibrium? Addressing this question could potentially break the ongoing cycle and determine which party might ultimately win this game.

1.2 Related works

Prior work on backdoors in federated learning [31] offers initial insights into the applicability of game theory for better understanding DNN backdoor risks. This paper takes a different approach. It focuses on centralized learning and on on developing a game-theoretic defense approach, rather than solely exploring attacker-defender dynamics. In this context, existing research in robust learning [32–34] have previously highlighted the value of game theory in studying adversarial machine learning.

Prior work has made use of various methodologies to address backdoor attacks in DNNs [24, 8], such as heuristic-based approaches, probabilistic models, or adversarial training. In this paper, we further expand the body of work on DNN integrity by using game theory. Due to its unique ability to model strategic interactions between rational attackers and defenders, game theory provides a structured framework for analyzing these adversarial behaviors, allowing for the identification of optimal strategies for both parties. Unlike heuristic approaches that may lack theoretical guarantees, or probabilistic models that can be computationally intensive, game theory may provide a balanced approach between analytical tractability and practical applicability, especially in scenarios involving clear, competitive objectives as found in security contexts.

1.3 Contributions

In this context, our research makes three significant contributions. First, we introduce a novel game-theoretic framework that models the interaction between a DNN backdoor attacker and a defender. This new formulation enables a detailed examination of each player's strategies and performance, with the goal of identifying the most effective strategies, typically known as Nash

equilibria [35]. Our approach advances the existing literature by providing a two-player game model that simultaneously evaluates the optimal strategies of both the attacker and the defender.

Second, instead of employing a complex bi-matrix game in our framework, we adopt a simpler, more tractable two-player zero-sum game. This simplification is crucial as it significantly streamlines the analysis and strategy development process by focusing on the zero-sum nature of the game, where one player's gain is precisely the other's loss. To achieve this, we develop a utility function that encapsulates the dual objectives of the players, which include maintaining the performance of a DNN's clean data accuracy while also addressing their conflicting goals concerning the success rate of a backdoor attack. This simplification not only enhances the analytical tractability but also bolsters the practical applicability of our game-theoretic approach to real-world scenarios, where clear and decisive strategies are paramount.

Our final contribution is the evaluation of our proposed game-theoretic framework using numerical simulations, exploring multiple game variants on a well-known dataset and classification task. We investigate three configurations with varying levels of control afforded to either the attacker or the defender. Each setting focuses on a different backdoor poisoning trigger regimen [20]. The core value of our framework lies in finding the best strategy for each player under each setting. To do so, we construct utility matrices through numerical simulations and examine existing saddle points for each setup. Moreover, determining the optimal strategies at equilibrium provides deeper insights into the performance capabilities of both the attacker and the defender across various situations.

Our proposed framework is attack-agnostic and designed to be used beyond the examples presented in this paper, such as applications on a wider range of attacks and countermeasures. To the best of our knowledge, this is the first paper to offer a self-contained framework for modeling the interaction between DNN attackers and defenders, thereby circumventing the current cat-and-mouse game between them. It standardizes the comparison between attacks and defenses by aiding in the identification of optimal strategies for the players. Furthermore, it provides valuable insights into the performance of both players.

The rest of this paper is organized as follows. Section 2 formalizes backdoor attacks on a computer vision task, covering the threat model and attack used in this work. Section 3 briefly introduces game theory and the game-theoretic formulation of attacker-defender interactions central to this paper. Section 4 reports our simulation results and discusses the optimum strategies for each player and their performance at the Nash equilibrium. Finally, Section 5 concludes this paper.

2 Backdoor attack

This section motivates our threat model, a *targeted, clean-label, data-poisoning-based backdoor on a classification task*, and introduces the attack and notation used in the rest of this paper.

2.1 Motivation for our threat model

Backdoor attacks on computer vision and their countermeasures is a thriving area of research [6, 8, 10–12, 24]. This paper assumes an attacker who targets a supervised learning model, specifically an *image classification* task. This setting is very common in the backdoor literature, from its early works like BadNets [22] to more recent demonstrations on face recognition for instance [8]. Therefore, it is a fitting example on which to base our framework.

Our use case attacker looks to inject a backdoor behavior in a *targeted* fashion, that is, the attacker aims to compromise the integrity of the model with a specific target in mind [22], e.g., forcing misclassifications towards a specific target class. This differs from untargeted attacks which aim to deteriorate a DNN's availability by causing general misclassifications [6, 36].

We focus on using a backdoor based on *data poisoning*. This is a core risk at the pre-training stage [6, 12, 22] where an attacker has hijacked the supply chain of a DNN trainer (e.g., at the data collection, data repository, etc.) such that a DNN's training data become compromised. The attacker manipulates a portion of a victim's dataset, modifying its images and, possibly, their labels such that any DNN trained on the dataset will learn a malicious behavior. Additionally, we follow a *clean-label backdoor* [23] use case. In this context, the attacker only manipulates the image content of the class(es) they are targeting in the compromised dataset. Labels are left unchanged. This use case matters in the case of data poisoning as the attacker is maximizing their stealth and therefore the chance of a victim trainer to embed a backdoor in a DNN down the line.

The choice of a targeted, clean-label backdoor threat model is motivated by the potential impact of such attacks in real-world scenarios and safety-critical fields [6, 10], like autonomous vehicles or face recognition. Data poisoning and clean-label backdoor attacks are particularly relevant as they represent stealthy and effective methods for embedding malicious behaviors in DNNs, often bypassing traditional detection mechanisms. These choices are supported by numerous studies [6, 8], which highlight the effectiveness of targeted, clean-label attacks in compromising DNN integrity while maintaining high performance on benign inputs Fig. 1.

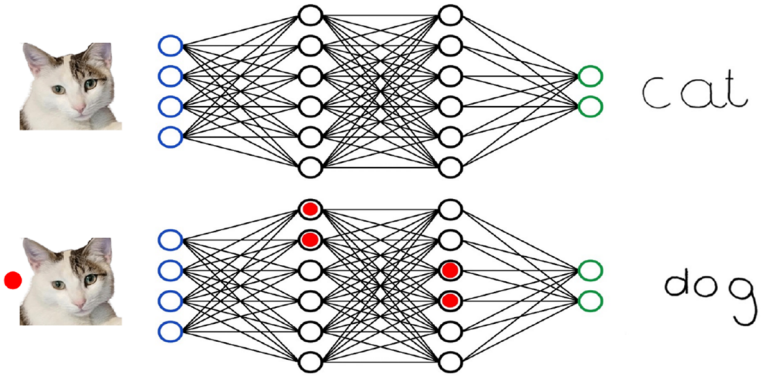


Fig. 1 Benign vs. backdoored behavior: The DNN correctly predicts "cat" in the benign input but misclassifies it as "dog" when altered with the backdoor's trigger: a "red circle"

2.2 Formalization

A DNN is an approximation function $\mathcal{F}_\theta$ that determines for a given training dataset $D_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{tr}}}$ the mapping from an input set $X = \{x_i\}_{i=1}^{N_{\text{tr}}}$ to an output set $Y = \{y_i\}_{i=1}^{N_{\text{tr}}} \in C$, where C is the set of classes, and $|C|$ is the total number of classes learned by $\mathcal{F}_\theta$, $\mathcal{F}_\theta(x_i) = y_i$, and θ are the DNN parameters optimized by solving the following problem:

$$\arg\min_{\theta} \sum_{i=1}^{N_{\text{tr}}} \mathcal{L}(\mathcal{F}_\theta(x_i), y_i). \tag{1}$$

After optimization, the DNN performance is evaluated on an unseen test dataset $D_{\text{ts}} = \{(x_j, y_j)\}_{j=1}^{N_{\text{ts}}}$. The chosen metric is the DNN's clean data accuracy (*CDA*, res. test accuracy) defined as follows:

$$CDA(\mathcal{F}_\theta, D_{\text{ts}}) = \frac{\sum_{j=1}^{N_{\text{ts}}} I(x_j, y_j)}{N_{\text{ts}}}, \tag{2}$$

where $I(x_j, y_j) = 1$ if $\mathcal{F}_\theta(x_j) = y_j$, and 0 otherwise.

A backdoor attack manipulates a DNN such that it outputs a wrong class label $\tilde{y}_i = t$ for a backdoored input $\tilde{x}_i$, where $\tilde{x}_i$ correspond to an input x_i altered with some trigger x_t. A backdoor approach uses training data poisoning where a subset P of m elements drawn from D_{tr} is altered with the trigger x_t as follows:

$$P = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{m} \tag{3}$$

$$\tilde{x}_i = (1 - \Delta_{\text{tr}}) \times x_i + \Delta_{\text{tr}} \times x_t \tag{4}$$

where Δ_{tr} is the backdoor attack's power or strength, which determines the overlay of the trigger x_t. Equation 4 quantifies the attacker's ability to embed a backdoor trigger in a DNN's training data. The rationale for this formulation is that it allows the attacker to balance the visibility of the trigger against the risk of detection. By adjusting Δ_{tr}, the attacker can fine-tune the influence of the trigger, ensuring that it is strong enough to be learned by the DNN but subtle enough to evade initial detection.

Here, we note that the attacker's power at training time Δ_{tr} (see Eq. 4) can differ from the one used at test time. As such, we use the notation Δ_{tr} for the attacker's power during training and Δ_{ts} for its test time equivalent[1]. Since a human investigation of test samples may be unfeasible in the case of online platforms where response speed is crucial, we surmise that the attacker is free to update Δ_{ts} at test time.

This paper focuses on a targeted, clean-label backdoor attack where the elements in P belong to the attacker's target class t. In other words, the poisoned training samples keep their original labels, i.e., $\tilde{y}_i = y_i = t$. Since all poisoned samples belong to the same class, the size of P is defined by the ratio $\alpha_{\text{tr}} \in (0, 1]$ of poisoned training samples belonging to class t such that:

$$\alpha_{\text{tr}} = \frac{m}{N_{\text{tr},t}}, m << N_{\text{tr},t} \tag{5}$$

where m is the size of the set P of poisoned samples of class t and $N_{\text{tr},t}$ is the number of training samples of class t.

This data poisoning process yields a poisoned dataset $D_{\text{tr}}^{\text{po}}$ such that training on it produces a backdoored DNN $\mathcal{F}_\theta^{\text{po}}$. The attacker expects that a victim DNN trained on $D_{\text{tr}}^{\text{po}}$ will learn to associate the trigger x_t with the target class t while keeping *CDA* on par with a benign model.

The backdoored DNN $\mathcal{F}_\theta^{\text{po}}$ is then assessed using its attack success rate (*ASR*). It corresponds to the proportion of wrongful classifications towards the backdoored class t that the attacker can induce by poisoning the

[1] In the rest of the paper, we use the terms overlay power, trigger power, and attack/defense power, for the attacker and defender, interchangeably.

test elements from D_{ts} (they belong to any source class $y \neq t$). This poisoned test set is denoted $D_{\mathsf{ts}}^{\mathsf{po}}$ and the *ASR* is computed as such:

$$ASR(\mathcal{F}_\theta^{\mathsf{po}}, D_{\mathsf{ts}}^{\mathsf{po}}) = \frac{\sum_{j=1}^{|D_{\mathsf{ts}}^{\mathsf{po}}|} I(\tilde{x}_j, t)}{|D_{\mathsf{ts}}^{\mathsf{po}}|}, \tag{6}$$

where $I(\tilde{x}_j, t) = 1$ if $\mathcal{F}_\theta^{\mathsf{po}}(\tilde{x}_j) = t$, and 0 otherwise[2].

The attacker's objective is to increase their *ASR* as much as possible while preserving the DNN's *CDA* such that it is indistinguishable from a benign model, i.e., the victim trainer will deploy the inconspicuous DNN. For ease of use for the reader, we summarize our notation choices in Table 1.

2.3 Our backdoor use case

2.3.1 Attacker side

The SIG attack, introduced by Barni et al. [20], is a backdoor attack that uses subtle sinusoidal or ramp signals as triggers. These triggers are spread over the input image, akin to a watermark. They are designed to be hard to detect [37], making them particularly suitable for clean-label attacks where the attacker's objective is to remain stealthy. We selected the SIG attack for this study due to its demonstrated effectiveness in bypassing detection while maintaining a high attack success rate (ASR) [37]. Its simplicity and the difficulty in reverse-engineering the trigger make it an ideal candidate for analyzing the strategic interactions between attackers and defenders in our game-theoretic framework. Following a clean-label setting, the attacker:

1. Selects a target class t,
2. Randomly draws a α_{tr} portion of the target class datapoint $D_{\mathsf{tr},t}$ to construct $D_{\mathsf{tr}}^{\mathsf{po}}$,
3. Applies one of the triggers described in SIG [20].

We use either of two backdoor triggers provided by SIG [20] to build $D_{\mathsf{tr}}^{\mathsf{po}}$: a simple ramp signal or a sinusoidal signal. Given an image x of h rows and w columns, the ramp signal is defined as $x_t(i,j) = j \times \Delta / w$, given $1 \leq i \leq h$ and $1 \leq j \leq w$. The sinusoidal signal is defined such that $x_t(i,j) = \Delta \sin(2\pi h f / w)$, given $1 \leq i \leq h, 1 \leq j \leq w$ and where f is the signal frequency. The attacker selects only one of the triggers to generate the poisoned samples as in Eq. 4, which are then used to construct $D_{\mathsf{tr}}^{\mathsf{po}}$.

[2] For an untargeted attack, $I(\tilde{x}_j, \tilde{y}_j) = 1$ if $\mathcal{F}_\theta^{\mathsf{po}}(\tilde{x}_j) \neq y_j$, and 0 otherwise.

Table 1 Table of notations

D_{tr}	Training dataset
D_{ts}	Testing dataset
$D_{\mathsf{tr}}^{\mathsf{po}}$	Poisoned training dataset
$D_{\mathsf{ts}}^{\mathsf{po}}$	Poisoned testing dataset
$\mathcal{F}_\theta$	Deep learning model
$\mathcal{F}_\theta^p$	Backdoored deep learning model
$\mathcal{L}$	Loss function
C	Number of classes
CDA	Clean data accuracy
ASR	Attack success rate
CDA_{cp}	Clean data accuracy on cleaned poisoned samples
ASR_{cp}	Attack success rate on cleaned poisoned samples
CDA_{cb}	Clean data accuracy on cleaned benign samples
ASR_{cb}	Attack success rate on cleaned benign samples
x_t	Trigger overlay
Δ	Trigger overlay power
$\tilde{x}_i$	Poisoned sample
Δ_{tr}	Attack power on training samples
Δ_{ts}	Attack power on testing samples
α_{tr}	Fraction of poisoned training samples
α_{ts}	Fraction of poisoned testing samples
α_{def}	Fraction of cleaned samples
t	Backdoor error target class
x_j^c	Cleaned sample
Δ_{def}	Defense power on testing samples
$G(.)$	Game
S_i	Strategy set for player i
u_i	Utility for player i
P_i	Probability distribution over S_i at the equilibrium
BG_{Min}	Backdoor game with minimum control
BG_{Int}	Backdoor game with intermediate control
BG_{Max}	Backdoor game with maximum control
u_A	Attacker's utility
u_D	Defender's utility
u_A^*	Attacker's utility at the equilibrium
u_D^*	Defender's utility at the equilibrium
u^*	Utility at the equilibrium
S_A^*	Attacker's strategy profile at the equilibrium
$Pr(S_A^*)$	Attacker's probability distribution over the S_A^*
S_D^*	Defender's strategy profile at the equilibrium
$Pr(S_D^*)$	Defender's probability distribution over the S_D^*

2.3.2 Defender side

We consider a naive defense that attempts to recover the non-poisoned image by reverse-engineering the additive nature of the backdoor, similar to the input purification approach found in [37]. The process is defined as:

$$x^{\mathsf{cl}} = \frac{x_i^{in} - \Delta_{\mathsf{def}} \times \hat{x}_t}{1 - \Delta_{\mathsf{def}}}, \tag{7}$$

Here, Δ_{def} represents the defender's overlay power, x_i^{in} is the potentially backdoored input image, and $\hat{x}_t$ is the estimated trigger to be removed. This equation operates on the premise that the backdoor trigger was linearly combined with the original image using a certain strength (Δ_{tr}). The defender, using an estimated trigger ($\hat{x}_t$) and an assumed overlay power (Δ_{def}), seeks to reverse this operation by subtracting the estimated trigger influence and normalizing the image. The goal is to reconstruct the clean image x^{cl} accurately, mitigating the backdoor's effect. However, the effectiveness of this defense heavily relies on the precision of the defender's estimates. Inaccuracies could result in incomplete trigger removal or unintended alterations to benign images, potentially causing misclassifications.

It is important to note that the defender does not initially know whether the input sample x_i^{in} is a benign image x_i or a poisoned one $\tilde{x}_i$. Consequently, this defense is applied regardless of the nature of the input image. Nevertheless, the defender could apply a preliminary step to detect poisoned samples then applying the filtering step only to those identified samples. However, this paper does not cover such scenario.

For our numerical simulation, we set several properties for the estimation of the trigger $\hat{x}_t$ from the defender's perspective. First, the difficulty of the estimation increases as the poisoning ratio α_{tr} decreases. The defender should not be able to recover a perfect estimate of x_t if α_{tr} is sneakily low. We surmise the same property with respect to the trigger power Δ_{tr}. Conversely, estimation becomes easier as these parameters increase. The attacker's strategy becomes more apparent.

In this context, a possible way to derive such an estimate is to assume that $\hat{x}_t$ can be recovered from a sample average over input images x_i^{in}, effectively reversing the additive poisoning process applied by the attacker. By averaging the input images and substituting this average in place of x_i^{in} in Eq. 7, the following expression is obtained:

$$\hat{x}_t = (\alpha_{\mathsf{tr}} \times \Delta_{\mathsf{tr}}) \times x_t + \bar{x}(1 - \alpha_{\mathsf{tr}} \times \Delta_{\mathsf{tr}}), \tag{8}$$

Here, $\bar{x} \sim \mathcal{N}(0, 1)$ represents the average noise across the non-poisoned data, modeled as a standard normal Gaussian distribution. This equation illustrates that the estimated trigger, and consequently the cleaned sample x^{cl}, may contain noise depending on the values of α_{tr} and Δ_{tr}. As these parameters increase, the accuracy of the trigger estimation improves, causing the estimate to converge toward the actual trigger x_t as the attack becomes more pronounced. The defender's ability to remove the backdoor effectively relies on accurate estimation of the trigger, which is influenced by the attack's strength Δ_{tr} and the proportion of poisoned data α_{tr}. In scenarios where these values are low, the process becomes more challenging, and the estimate may be less reliable due to increased noise in the data.

In a more empirical perspective, we assume that the defender has some trigger reverse-engineering capability, which is in line with prior work in the backdoor literature [38, 39]. Moreover, even without knowing the trigger, the defender may partially or fully modify input samples in a deterministic manner or via a random filter to improve the model's robustness whenever assuming the presence of a backdoor [24].

3 Backdoor game formulation

3.1 Game theory in a nutshell

Game theory is a well-established field of mathematics that analyzes competitive and cooperative interactions among decision-makers, referred to as players, who make interdependent choices. Its foundations were laid with the publication of "Theory of Games and Economic Behavior" by John von Neumann and Oskar Morgenstern in 1944 [40]. The fundamental assumption in game theory is that players are rational and intelligent, having clear preferences over game outcomes and the ability to choose actions that maximize their returns. However, there are important exceptions, such as bounded rationality, where players have limited cognitive resources; and behavioral game theory, which accounts for psychological biases and irrational behaviors that can lead to decisions deviating from pure payoff maximization [41].

The primary objective of game theory is to predict the behavior of rational players in a game or provide guidance on playing against rational opponents. Each rational player has clear preferences regarding the outcomes of a game where they all perform mutually dependent actions. Given the expected actions of their opponents, a rational player always selects the course of action that yields the most favorable payoff.

The normal or strategic form is the basic game model investigated in non-cooperative game theory. A normal game lists the strategies available to each player and the results associated with every potential set of decisions. A tuple of four components $G(S_1, S_2, u_1, u_2)$ serves as the definition of a two-player normal form game, where $S_1 = \{s_{1,1} \dots s_{1,n_1}\}$ and $S_2 = \{s_{2,1} \dots s_{2,n_2}\}$ are the sets of strategies available to the first and second players. Then, for $p \in \{1, 2\}$, $u_p(s_{1,i}, s_{2,j})$ is the payoff or utility of the game for the p^{th} player when the first player chooses the strategy $s_{1,i}$ and the second chooses $s_{2,j}$. Each pair of strategies $(s_{1,i}, s_{2,j})$ is called a strategy profile.

Utility (res. Payoff) matrices are used as a compact representation of normal-form games. Thus, a more general formulation of the normal form game

is given by $G(N, S, \mathbf{u})$ where $N = \{1, 2, \ldots, n\}$ is a set of players, $S = \{S_1, \ldots, S_n\}$ are the sets of strategies, $S_i = \{s_{i,1} \ldots s_{i,n_i}\}$ represents the strategies set available to the players, and the vector $\mathbf{u} = (u_1, \ldots, u_n)$ is the set of the game utilities with u_i corresponding to the utility of the i^{th} player. A strategy profile for the game can be represented by the vector $(s_{1,i_1}, \ldots, s_{n,i_n}) \in S$.

Finding equilibrium points that reflect, to some extent, a decent choice for both players is a common goal in game theory. There are several definitions of equilibrium, but the one developed by John Nash [35, 42] is the most well-known and used. For example, a profile $(s^*_{1,i}, s^*_{2,j})$ in a two-player game is a Nash equilibrium if the following conditions about the utility of the players are met:

$$\begin{aligned} u_1(s^*_{1,i}, s^*_{2,j}) &\geq u_1(s_{1,i}, s^*_{2,j}) \quad \forall s_{1,i} \in S_1 \\ u_2(s^*_{1,i}, s^*_{2,j}) &\geq u_2(s^*_{1,i}, s_{2,j}) \quad \forall s_{2,j} \in S_2 \end{aligned} \tag{9}$$

In a zero-sum game, the utilities of the two players sum up to zero: $u_1 + u_2 = 0$. A profile is in a Nash equilibrium if no player can unilaterally change their strategy to increase their utility. This is the equilibrium or saddle point of the game.

Pure strategy Nash equilibria and mixed strategy Nash equilibria are the two distinct types of Nash equilibria. A pure strategy Nash equilibrium occurs when players usually choose a single strategy. In such a case, a pure strategy profile $(s^*_{1,i}, s^*_{2,j})$ is a Nash equilibrium for the game with $s^*_{1,i}$ and $s^*_{2,j}$ the pure strategies for player 1 and player 2 respectively. Conversely, in a mixed strategy Nash Equilibrium, players may use a particular probability distribution over the strategy set in order to randomize their decisions.

In normal-form games, dominance solvable games [42] exhibit a stronger form of equilibrium. A strictly dominant strategy for a player implies that this strategy is the best strategy for the player regardless of the other player's strategy. Based on a fundamental principle of game theory, if mixed strategies are allowed, every game with a finite number of players and a finite number of pure strategies for each player is said to have at least one Nash equilibrium [43]. Finding the Nash equilibrium of a game corresponds to solving one of the two linear programming problems [44] expressed as follows:

$$\max_{S_1} \min_{S_2} u_1(s_{1,i}, s_{2,j}) = \min_{S_2} \max_{S_1} u_1(s_{1,i}, s_{2,j}) \tag{10}$$

In a two-player game, by denoting the probability distribution over the strategy profile at the equilibrium for each player i as P_i, the expected utility can be computed as follows:

$$\begin{aligned} U_1(P_1, P_2) &= \sum_{S_1, S_2} P_1(s_{1,i}) u_1(s_{1,i}, s_{2,j}) P_2(s_{2,j}), \\ U_2(P_1, P_2) &= \sum_{S_1, S_2} P_1(s_{1,i}) u_2(s_{1,i}, s_{2,j}) P_2(s_{2,j}). \end{aligned} \tag{11}$$

3.2 Backdoor game

3.2.1 Overview

In this work, we employ a game-theoretic framework to model the interaction between a DNN model defender and a backdoor attacker, framing it as a two-player game. Both the attacker and defender are rational agents with full knowledge of the game's structure but lack complete information about each other's strategies. The defender aims to maximize their DNN's CDA while minimizing the attacker's ASR, whereas the attacker seeks to maximize their ASR while keeping the DNN's CDA above a threshold to avoid detection and rejection (the defender will not use the DNN if its CDA is too low).

We assume the DNN model is trained in a controlled environment, allowing the defender to implement defensive measures, albeit within real-time constraints. This interaction is modeled as a one-shot, zero-sum game where both the attacker and defender commit to a strategy without iterative adjustments. While this approach simplifies the analysis and provides a foundational understanding, we recognize that real-world scenarios can be more complex, involving multiple attackers, defenders, or third-party entities, where strategies might differ, and a zero-sum model may not be sufficient.

In particular, situations where both the attacker and defender experience simultaneous losses-such as when defensive measures degrade overall system performance or when an attack only partially succeeds but at a significant cost-are not fully captured by a zero-sum framework. Additionally, cases where the defender's success does not directly equate to the attacker's failure, or where considerations like energy consumption, defense cost, and complexity come into play, may require more advanced models. Nonetheless, our zero-sum assumption remains valid as a foundational study, offering a basis for exploring more sophisticated scenarios.

To explore these dynamics in DNN backdoor attacker-defender interactions, we present three scenarios. Each scenario offers different levels of control and strategic options for the players, enabling a comprehensive examination of their decision-making processes.

1. *Backdoor game with minimum control* (BG_{Min}): The attacker controls the backdoor trigger power during both the training and testing phases (Δ_{tr} and Δ_{ts}, respectively), while the defender only controls their trigger removal power Δ_{def} during the test phase.
2. *Backdoor game with intermediate control* (BG_{Int}): This scenario provides the attacker with an increased strategic flexibility. They now also have the ability to manipulate the backdoor poisoning ratio α_{tr} during the training phase.

3. *Backdoor game with maximum control* (BG_{Max}): In this scenario, both players are granted maximum control over their strategy sets. Specifically, on the top of BG_{Int}, the defender now also decides whether or not to apply their defense to an input sample, adding a decision probability α_{def} to their strategy set.

These configurations are designed to incrementally raise the complexity and autonomy in decision-making for both the defender and the attacker, allowing for a detailed exploration of their strategic behaviors and the potential equilibria of the game. This systematic variation in control settings aims to clarify the dynamics of adversarial interactions and support the development of effective defensive strategies.

To keep this paper self-contained and analytically clear, we choose to simplify the interaction between the DNN backdoor attacker and defender into a manageable two-player zero-sum game, leaving a more complex bi-matrix game formulation for future works. This simplification focuses on the zero-sum aspect of the game, where one player's gain is the another player's loss, making the analysis straightforward. In so doing, we develop a utility function that encapsulates the dual objectives of the players: the shared goal of maximizing the CDA of the DNN and the players' conflicting objectives regarding the backdoor's ASR.

3.2.2 Constructing a utility function

As the core of our game, we design an appropriate utility function. Based on our simplified setup as described in Section 3.2, we first exclude the costs related to defense operations (Eq. 7) and the injection of the backdoor trigger (Eq. 4), along with their computational costs, as these are considered minor operations. Additionally, the inference time cost of the model (i.e., one forward propagation per input sample) is also disregarded. This approach not only streamlines our analysis but also enhances the generalization of our game-theoretic framework.

Therefore, in the context of backdoor attacks with an attacker (A) and a defender (D), the utility functions for A and D are formulated to capture the dynamics of a competitive zero-sum game, i.e., a player's win corresponds inversely to the other's loss. For backdoor attacks, it is realized through the following:

$$\begin{aligned} u_A &= ASR \times \mathbf{1}[CDA > CDA_{\text{inf}}], \\ u_D &= -u_A, \end{aligned} \tag{12}$$

where CDA_{inf} represents the minimum acceptable clean data accuracy on benign samples accessible to the defender D. CDA_{inf} thus implies that the defender D rejects DNN models when their *CDA* drops below some threshold. This is a common practice in the backdoor literature [22] that is dependent on the defender's models and task requirements. Concurrently, the *ASR* is the attacker A's attack success rate, known to the game once a defense is applied. When factoring in the indicator function and CDA_{inf}, it is understood that a backdoored model rejected by the defender D, because of a too low *CDA*, results in an *ASR* of 0% for the attacker A.

As an example, the defender D might accept an accuracy of 80% on some task. Given a use case where a benign DNN may achieve an accuracy of 90%, the attacker A therefore must perform a backdoor attack that does not result in a *CDA* drop of more than 10%, thus $CDA_{\text{inf}} = CDA - 0.1$. If the resulting *CDA* of a backdoored DNN is below CDA_{inf}, the model is rejected and the attacker A's ASR is effectively null.

3.2.3 Game formalizations

Through these three scenarios with progressively larger sets of strategies, which reflect greater freedom of choice for both players, we provide a comprehensive understanding of how both defender and attacker may deploy their strategies. This further highlights the implications these choices have on the security of DNNs in adversarial settings and to ultimately understand who may win the game.

Definition 1 – BG_{Min}: The backdoor game with minimum control is a zero-sum, two-players, strategic game played by the Attacker (A) and the Defender (D), defined by the following strategies and utilities:

- The sets of strategies available to the attacker and the defender are respectively comprised of the set of all possible overlay power values Δ_{tr} and Δ_{ts}, and Δ_{def}:

$$\begin{aligned} S_A &= (\Delta_{\text{tr}}, \Delta_{\text{ts}}) \in [0,1] \times [0,1], \\ S_D &= \Delta_{\text{def}} \in [0,1] \end{aligned} \tag{13}$$

- The utility functions for A and D are defined by (12).

Definition 2 – BG_{Int}: The backdoor game with intermediate control is a zero-sum, two-players, strategic game played by the players A and D, defined by the following strategies and payoffs:

- The sets of strategies available to the attacker and defender are respectively the set of possible values of Δ_{tr}, Δ_{ts} and α_{tr}, and Δ_{def}:

$$\begin{aligned} S_A &= (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}}) \in [0,1] \times [0,1] \times [0,1], \\ S_D &= \Delta_{\text{def}}, \in [0,1]. \end{aligned} \tag{14}$$

- The utility functions for A and D are defined by (12).

Definition 3 – BG_{Max}: The backdoor game with maximum control is a zero-sum, two-players, strategic game played by the players A and D, defined by the following strategies and payoffs:

- The sets of strategies available to the attacker and defender are respectively the set of possible values of Δ_{tr}, Δ_{ts} and α_{tr}, and α_{def} and Δ_{def}:

$$\begin{aligned} S_A &= (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}}) \in [0,1] \times [0,1] \times [0,1], \\ S_D &= (\alpha_{\text{def}}, \Delta_{\text{def}}), \in [0,1] \times [0,1]. \end{aligned} \tag{15}$$

- The utility functions for A and D are defined by (12).

We note that the sets of strategies that are available to the players A and D in the three game variants are defined as continuous sets. In practice, however, quantization is applied. In the utility matrices, *CDA* and *ASR* are computed for both players after applying the defense strategy to the test dataset D_{ts} as follows:

$$\begin{aligned} CDA &= (CDA_{\text{cb}} + CDA_{\text{cp}})/2, \\ ASR &= (ASR_{\text{cb}} + ASR_{\text{cp}})/2. \end{aligned} \tag{16}$$

where CDA_{cb} and CDA_{cp} represent the test accuracy metrics on the benign and poisoned test sets, respectively, after the defense mechanism has been applied. Similarly, ASR_{cb} and ASR_{cp} indicate the attack success rates on the benign and poisoned test sets, respectively, following the application of the defense.

3.2.4 Game dynamics

The goal of our framework is to model the interaction between attacker and defender as a dynamic game. Such game is characterized by evolving finite set of strategies where each player's decisions influence the subsequent responses of their opponent. The core rationale behind this formulation is to reflect the continuous adaptation seen in cybersecurity environments, where attackers and defenders dynamically adapt their tactics and strategies based on each other's actions.

We thus designed our utility function, critical to any game-theoretic analysis, to provide a key performance metrics of such dynamics while keeping close to the backdoor literature (i.e., why we use the success rate of a backdoor attack (*ASR*) and the clean data accuracy (*CDA*) of the DNN). The utility function should therefore capture the effectiveness of each player's strategies under the assumption of rational behavior where everyone aims to maximize their respective outcomes under different level of knowledge and control.

We further note that the dynamic nature of the game incorporates inherent feedback mechanisms, where adjustments in one player's strategy lead to potential strategy reevaluations and adjustment by the other player, creating a continuous interactive and interplay loop. Initially described as a zero-sum game, the gain of one player in our model equates to the loss of the other, which is fundamental in adversarial settings such as the ones explored here. Each game configuration (minimal, intermediate, and maximum control) increases the complexity of a player's decision-making, reflecting an increasingly complex real-world scenario. This rationale aims to highlight the value of the theoretical and practical implications of our framework in real-world adversarial environments in DNN security.

Finally, we note that this paper and its framework do not aim to compare the performance of different backdoor attacks and defenses. Rather, this paper offers a way to better understand the context of their design and the strategic constraints and dynamics they depend on. From the attacker's perspective, this will help highlight the best possible attack regimen and, from a system designer or defender side, how to better protect the running DNN in an adversarial environment.

3.2.5 Assumptions and limitations

Our proposed game-theoretic model operates under several key assumptions that are common in the literature but may limit its applicability in more complex, real-world scenarios. First, we assume a rational behavior from both the attacker and defender, meaning each player is expected to make decisions that maximize their respective payoffs. This rationality assumption simplifies the analysis but may not fully capture scenarios where players behave unpredictably or irrationally.

Second, the game is modeled in a static environment where the players' strategies and the game conditions are assumed to be fixed during the interaction. This assumption overlooks the dynamic nature of many real-world systems, where strategies can evolve over time and external factors might influence the game.

Additionally, our framework is based on a zero-sum game, where the gain of one player is exactly balanced by the loss of the other. While this is a common approach in modeling adversarial interactions, it assumes a direct and exclusive conflict of interests, which may not always hold true. For instance, in some scenarios, both the attacker and defender might incur losses simultaneously, or the success of one party might not entirely translate into the failure of the other. Such scenarios suggest the need for

more complex models, such as non-zero-sum games, where shared risks, collaborative behaviors, or external constraints (e.g., energy consumption, defense cost, and limited access to the system through APIs during inference) can be accounted for.

The assumptions made in this paper aim to establish a fundamental framework that is analytically tractable and can serve as a foundation for understanding basic adversarial dynamics in backdoor attacks.

4 Experimental results

In this section, we perform a series of numerical simulations of our game-theoretic framework. This framework allows the study the behavior of either attacking or defending players given an increasingly complex strategy mix on either side. Doing so, we evaluate the performance attainable by each player and highlight the existence of equilibria involving either pure or mixed strategies. By analyzing the behavior of each player at these equilibria, we assess their utility and determine the best attainable performance for each backdoor player. This allows us to identify a potential winner.

As a note, we refer the reader to the Table 1 for a summary of the different notations used in this section.

4.1 Experimental setup

4.1.1 Dataset and models

We use the MNIST dataset [45] in our numerical simulations to analyze our set of games. While MNIST is simple, the framework can be easily translated to other datasets and different DNN architectures. We use a shallow CNN architecture set with the following layers: 64-filter convolution, max pooling, 128-filter convolution, max pooling, 256-neuron fully-connected layer, and 10-neuron fully-connected layer. The kernel size of convolutions is set to 5. ReLU activations are used. For each game and strategy profile, we train the network for 100 epochs with a batch size of 64. Following the completion of each model training, we compute the utility value using the resulting test CDA and ASR. This utility value represents a single entry in the utility matrix. The reference test accuracy of the benign model $\mathcal{F}_\theta$ is 99.07%.

4.1.2 Game setups

As described in Section 3.2, each game includes an increasing number of parameters afforded to each player in their strategy set. The Δ_{tr}, Δ_{ts}, and α_{ts} parameters are available to the attacker, and Δ_{def} and α_{def} to the defender. Table 2 summarizes the parameters in the strategy set that each player exerts control over for each game BG_{Min}, BG_{Int}, or BG_{Max}.

In the BG_{Min} game, the poisoning ratios α_{tr} and α_{ts} and the defense decision ratio α_{def} (i.e., the ratio of inputs on which the defense is applied) are all fixed to 1.0. In the BG_{Int} game, the attacker incorporates α_{tr} in their strategy. Finally, in BG_{Max}, both player have access to their poisoning/coverage ratios. For BG_{Max}, both players have access to their widest strategy set.

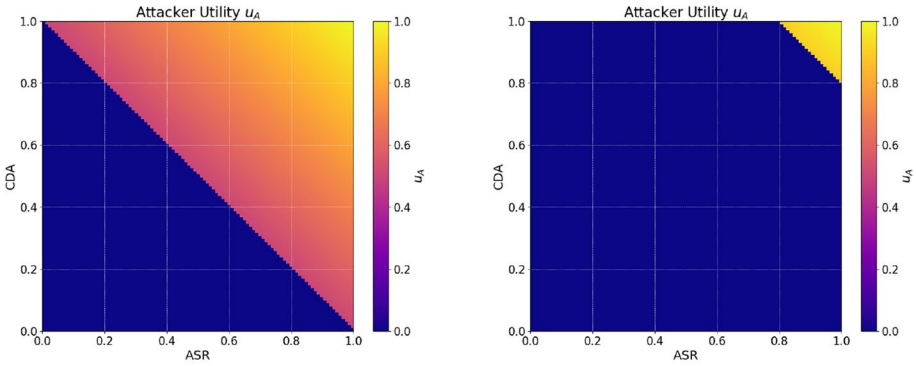
For each game, we quantize α_{tr} and α_{def} such that $\alpha_{\mathsf{tr}}, \alpha_{\mathsf{def}} \in \{0.05, 0.1, \cdots, 0.9, 1.0\}$. We do the same with the attacker and defender's overlay powers, Δ_{tr}, Δ_{ts} and Δ_{def} such that $\Delta_{\{\mathsf{tr},\mathsf{tr},\mathsf{def}\}} \in \{0.01, \cdots, 0.09\} \cup \{0.1, \cdots, 0.5\}$. The maximum overlay power for the attacker is empirically selected by evaluating the highest achievable attack success rate (*ASR*) given no defense.

In our game's utility function, as defined in Eq. 12, we arbitrarily set CDA_{inf} to 0.9, an acceptable up-to 10% drop in ASR in the context of the MNIST [45] dataset. Though this acceptable drop is larger than the $1-2\%$ drop typically used in the literature [22, 37], we motivate this choice in that it allows us to possibly capture a wider range of potential optima. If the CDA falls below this threshold, the DNN is considered useless from the defender's perspective and will not be deployed. Figure 2 shows the utility function plot for $CDA_{\mathrm{inf}} = 0.5$ and $CDA_{\mathrm{inf}} = 0.9$. The figure illustrates that, when a defender tolerates a lower CDA_{inf}, the range in which the attacker can achieve a high *ASR* increases.

During evaluation, CDA_{cb} and CDA_{cp} denote the test accuracy metrics on the benign and poisoned test sets

Table 2 Parameter (and their value ranges) used in each game's strategy set

		Parameter control in *BG* setup			
Player	**Parameter**	BG_{Min}	BG_{Int}	BG_{Max}	**Range**
Attacker	Δ_{tr}	✓	✓	✓	$\{0.01, \cdots, 0.09\} \cup \{0.1, \cdots, 0.5\}$
Attacker	Δ_{ts}	✓	✓	✓	$\{0.01, \cdots, 0.09\} \cup \{0.1, \cdots, 0.5\}$
Attacker	α_{tr}		✓	✓	$\{0.05, 0.1, \cdots, 0.9, 1.0\}$
Attacker	α_{ts}				1
Defender	Δ_{def}	✓	✓	✓	$\{0.01, \cdots, 0.09\} \cup \{0.1, \cdots, 0.5\}$
Defender	α_{def}			✓	$\{0.05, 0.1, \cdots, 0.9, 1.0\}$



(a) $CDA_{\text{inf}} = 0.5$ (b) $CDA_{\text{inf}} = 0.9$

Fig. 2 Plot of the utility function u_A as described in Eq. (12) for different values of CDA_{inf}

after applying the defense, referred to as cleaned benign (cb) and cleaned poisoned (cp) data, respectively (see Eq. 7). Similarly, ASR_{cb} and ASR_{cp} represent the ASR on the benign and poisoned test sets *after applying the defense*, respectively.

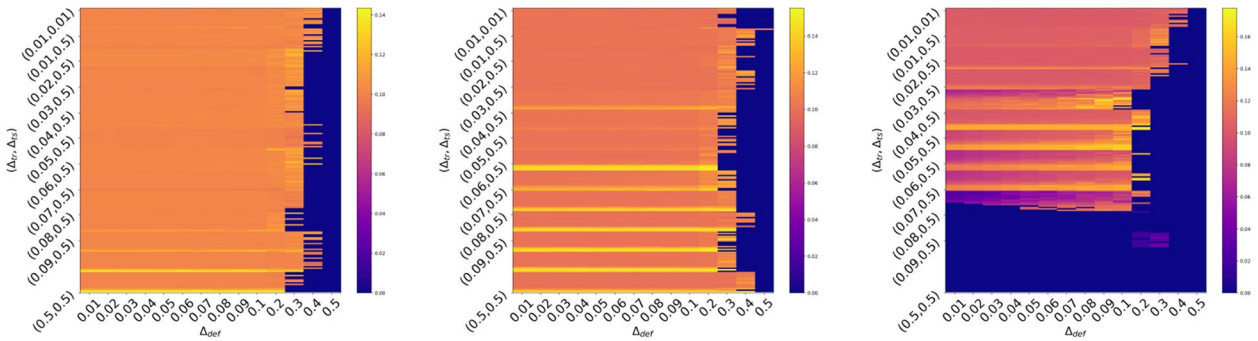
4.1.3 Backdoor attack setup

We use either the sinusoidal trigger with a frequency parameter of $f = 6$ or the ramp trigger (both drawn from Barni et al. [20]). The target class t of the backdoor attack is set to digit 3 in this paper.

4.1.4 Running experiments and result assessment

We proceed by employing utility matrices to solve our zero-sum games through linear programming, which corresponds to resolving the min-max problem stated in Eq. (10). The resulting utilities for each game are then visualized using a 2-dimensional plot, where defender strategies are mapped on the x-axis and attacker strategies on the y-axis. Such representation offers a clear representation of the competitive dynamics between each player (see Fig. 3). Due to the extensive strategy sets available in different game setups, displaying every possible option on the two axes would be impractical. Therefore, we selectively chose specific strategy profiles to display, where equilibria tend to appear, ensuring the plots remain clear and informative.

As part of our experiments, we examine the dynamics and player strategies for all configurations BG_{Min}, BG_{Int}, and BG_{Max}. Each game is analyzed using the flattened utility matrix representation mentioned above, alongside a summary table illustrating the performance of the players at equilibria. These tables present the optimal strategy profiles for both players, denoted as S_A^* and S_D^*, as defined in Section 3.2, alongside their corresponding probability distributions, $Pr(S_A^*)$ and $Pr(S_D^*)$, representing pure or mixed Nash equilibrium strategies. In the cases where a pure strategy Nash equilibrium is present for either player, the associated probability is 1 (e.g., see Table 5). Conversely, instances with mixed strategy Nash equilibria (for either the attacker or defender) introduce a probability distribution over the strategy profiles selected by the



(a) $\alpha_{\text{tr}} = 0.05$ (b) $\alpha_{\text{tr}} = 0.5$ (c) $\alpha_{\text{tr}} = 1.0$

Fig. 3 BG_{Min}: u_A with sine trigger and different α_{tr}

player. In such scenarios, these mixed strategies are presented across distinct columns, alongside their respective probabilities, $Pr(S_A^*)$ and $Pr(S_D^*)$ (e.g., see Table 4).

4.2 Results with BG_{Min}

4.2.1 Sinusoidal trigger

Analysis of the utility matrices Using various poisoning ratios for the BG_{Min} game, we observe that the attacker optimizes their utility u_A (i.e., this reduces the defender's utility u_D) by carefully selecting their strategy set values for their training and testing-time trigger powers (Δ_{tr}, Δ_{ts}). Additionally, the attacker's utility u_A increases with the poisoning ratio α_{tr}, as demonstrated by different cases of the game's utility matrix in Fig. 3. For instance, when $\alpha_{\text{tr}} = 0.05$, the maximum u_A is slightly above 0.14. This increases to above 0.15 when $\alpha_{\text{tr}} = 0.5$, and to beyond 0.16 when $\alpha_{\text{tr}} = 1$ (see also in Fig. 3). This underscores the importance for the attacker in having a greater access to the training dataset, as it helps enhance the backdoor's attack success rate (ASR), to the extent that it does not compromise the clean data accuracy (CDA) and get discovered by the defender, thereby increasing the attacker's overall utility.

Overall, we observe that the attacker's strategy profiles that maximize their utility consistently exhibit a fixed test set trigger power of $\Delta_{\text{ts}} = 0.5$, irrespective of whether these points are equilibrium points for the game. This trend can be traced back to the consistently high utilities u_A in the rows of the utility matrices (see the light orange and yellow rows in Fig. 3 when $\Delta_{\text{ts}} = 0.5$). However, when the defender employs a high defense power (Δ_{def}), the attacker's utility decreases toward zero (see the blue zones in the utility matrices on columns starting with $\Delta_{\text{def}} \geq 0.3$ in Fig. 3 for $\alpha_{\text{tr}} = 0.05$ and $\alpha_{\text{tr}} = 0.5$).

Finally, as the poisoning ratio α_{tr} increases, the size of the blue areas also increases. This is because a higher number of poisoned training samples makes the DNN backdoor more evident to the defender. For example, with $\alpha_{\text{tr}} = 1.0$, all target class t samples contain the backdoor trigger.

Analysis of the equilibrium strategies To further understand the dynamics of the BG_{Min} game, we analyze the equilibrium strategies for both the attacker and the defender across different poisoning ratios α_{tr}.

In the case with $\alpha_{\text{tr}} = 0.05$, the poisoning ratio is very small. On average, the attacker poisons 306 out of the 6,131 training samples for the target class $t = 3$, meaning the attacker does not have significant access to the training dataset. Consequently, the attacker pairs this low α_{tr}, which they cannot control, with a low $\Delta_{\text{tr}} = 0.01$ to cause poor trigger estimation by the defender. Additionally, using a very low testing trigger power $\Delta_{\text{ts}} = 0.01$ in conjunction with a high defense power $\Delta_{\text{def}} = 0.5$ would cause the defender to harm benign samples more than merely reducing the attack's effect. Thus, as the pure strategy Nash Equilibrium point ((0.01, 0.01), 0.5) results in $CDA < CDA_{inf}$, and because the defender, following their dominant strategy, has no incentive to change their strategy regardless of the attacker's strategy, the recommended result would be for the defender to not deploy the model (see Table 3).

Table 3 Equilibrium point of BG_{Min}: $\alpha_{\text{tr}} = 0.05$ and sine trigger

Profiles	Parameters	Equilibrium
Attacker	$S_A^* = (\Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.01, 0.01)
	$Pr(S_A^*)$	1.0
Defender	$S_D^* = (\Delta_{\text{def}})$	0.5
	$Pr(S_D^*)$	1.0
Utility	$u_A^* = -u_D^* = u^*$	0.0

For the case with $\alpha_{\text{tr}} = 0.5$, the fraction of poisoned samples is, on average, 50% of the target class t. Consequently, the information delivered about the attack has increased compared to the case with $\alpha_{\text{tr}} = 0.05$. The attacker exploits this fact to confuse the defender regarding their behavior by using a mixed strategy Nash equilibrium to balance between delivering information to the defender, which worsens their trigger estimation. This is demonstrated by the adopted strategy (0.02, 0.01) (see Table 4) and the maximizing of the error rate (and raising of the ASR) by assigning some probability to high ($\Delta_{\text{tr}}, \Delta_{\text{ts}}$) profiles. In response, the defender attempts to counter this strategy by distributing their probabilities over both high and low values of Δ_{def}, following the attacker's mixed strategy. Following this scenario, the defender deploys the model as $CDA > CDA_{inf}$ (see Table 4).

In other cases with α_{tr} ranging from 0.1 to 0.4 and from 0.6 to 1.0, a similar analysis applies. As the information delivered to the defender increases with $\alpha_{\text{tr}} = 0.9$ and $\alpha_{\text{tr}} = 1.0$, the trigger estimation improves. Consequently, the attacker tries to balance this by using low Δ_{tr} values. In these scenarios, the defender responds strongly by using high Δ_{def} values. This results in the attacker losing the game because the defender degrades the performance of both poisoned and benign samples, as indicated by the low values of CDA_{cb} and CDA_{cp}. Even though this also degrades the attack performance on ASR_{cb} and ASR_{cp}, the defender ends up with $CDA < CDA_{inf}$, thus winning the game by not deploying a non-performing model

Table 4 Strategy profiles of BG_{Min}, given $\alpha_{\text{tr}} = 0.5$ and sine trigger

Profiles	Parameters	Equilibria			
Attacker	$S_A^* = (\Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.02, 0.01)	(0.2, 0.5)	(0.3, 0.5)	(0.5, 0.5)
	$Pr(S_A^*)$	0.9077	0.0001	0.0244	0.0678
Defender	$S_D^* = (\Delta_{\text{def}})$	0.05	0.3	0.4	0.5
	$Pr(S_D^*)$	0.7207	0.0012	0.0095	0.2686
Utility	$u_A^* = -u_D^* = u^* = -0.1049$				

Table 5 Equilibrium point of BG_{Min}: $\alpha_{\text{tr}} = 1.0$ and sine trigger

Profiles	Parameters	Equilibrium
Attacker	$S_A^* = (\Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.01, 0.01)
	$Pr(S_A^*)$	1.0
Defender	$S_D^* = (\Delta_{\text{def}})$	0.5
	$Pr(S_D^*)$	1.0
Utility	$u_A^* = -u_D^* = u^*$	0.0

and preventing the attack. On the other hand, when the attacker uses covert strategies in all other cases (as with the other α_{tr} values), the backdoored model is deployed even if the defender experiences a performance drop in terms of ASR or CDA (see Tables 5 and 6).

4.2.2 Ramp trigger

Analysis of the utility matrices We begin by examining the utility matrices for the BG_{Min} game with the ramp trigger (see Fig. 4). While the overall utility behavior is similar to that of the BG_{Min} game with the sine trigger (see Section 4.2.1), a distinct trend emerges with the ramp trigger. Specifically, the ramp trigger results in broader high u_A light orange and yellow regions (indicating high utility) in the attacker's utility matrix. This suggests that the strategy set beneficial for the attacker is comparatively larger when using the ramp trigger compared to the sine trigger. However, comparing the maximum attainable utility by the attacker between the ramp and sine triggers (see Fig. 4 versus Fig. 3), we see that the sine trigger achieves a slightly higher maximum utility. This observation is also reflected in the comparison between Table 6 and Table 10. The reason for this difference lies in the nature of the triggers: the ramp trigger modifies only half of the input image, whereas the sine trigger modifies the entire input sample, making it more pervasive.

This difference underscores the dataset-specific impact of trigger selection: different triggers can vary in their effectiveness and how easily they are learned by the model, affecting the activation patterns of relevant neurons during inference and, consequently, the attack performance. From the utility matrices in Fig. 4, we observe that, for $\alpha_{\text{tr}} = 0.05$ and $\alpha_{\text{tr}} = 1.0$, the defender has a dominant strategy at $\Delta_{\text{def}} = 0.5$ (see the dark purple zone). Additionally, a similar observation about the dual behavior of the attacker is noted at $\alpha_{\text{tr}} = 0.5$. Here, examining the $\Delta_{\text{def}} = 0.5$ column from top to bottom, we see some orange and yellow rows, indicating that the defender adjusts their dominant strategy in response to the information level conveyed by the attacker's dual behavior.

Analysis of the equilibrium strategies The equilibrium strategies for the BG_{Min} game with the ramp trigger reveal that broader utility regions offer the attacker increased strategic flexibility, particularly at extreme poisoning ratios ($\alpha_{\text{tr}} = 0.05$ and $\alpha_{\text{tr}} = 1.0$). At $\alpha_{\text{tr}} = 0.05$, both players decide to settle into a pure strategy Nash equilibrium. The attacker minimizes their trigger powers to $(\Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.01, 0.01)$ to evade detection, while the defender responds with a strong defense at $\Delta_{\text{def}} = 0.5$. This leads to a zero utility outcome for both, indicating a tightly contested scenario where neither side gains an advantage (see Table 7).

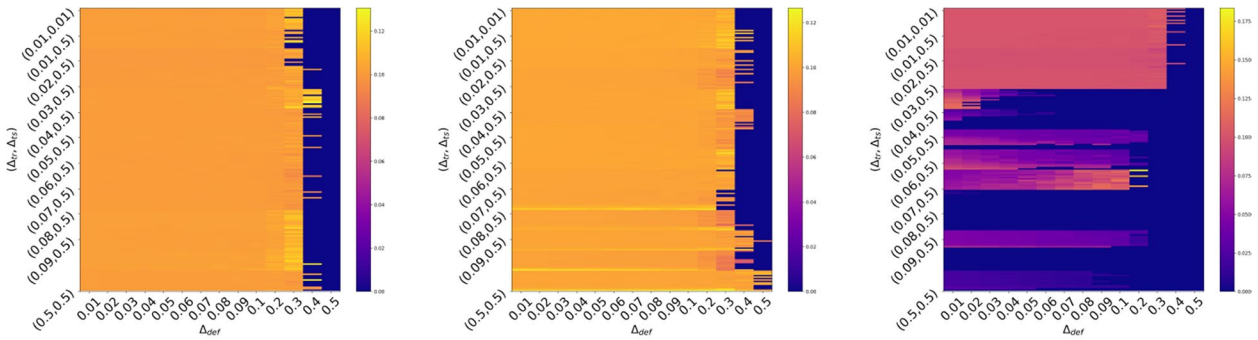
As the poisoning ratio increases to $\alpha_{\text{tr}} = 0.5$, the dynamics become more intricate, resulting in a mixed strategy equilibrium. The attacker now employs a dual approach, alternating between $(\Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.5, 0.03)$ with a probability of 0.557 and (0.5, 0.5) with a probability of 0.443. The defender, in turn, adjusts their strategy by choosing $\Delta_{\text{def}} = 0.4$ with a probability of 0.422 and $\Delta_{\text{def}} = 0.5$ with a probability of 0.578. This complex interplay reflects the attacker's need to balance between maximizing their utility and minimizing detection, while the defender continuously adapts to these shifting tactics (see Table 8).

At the highest poisoning ratio, $\alpha_{\text{tr}} = 1.0$, the attacker reverts to a pure strategy Nash equilibrium, again minimizing trigger powers to $(\Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.01, 0.01)$. This conservative approach ensures the backdoor remains subtle enough to avoid triggering the defender's countermeasures. Predictably, the defender maintains their strategy of $\Delta_{\text{def}} = 0.5$, resulting in a utility of 0.0 for both players, underscoring the equilibrium's stability in this scenario (see Table 9).

Overall, as shown in Table 10, the ramp trigger's broader strategic options do not necessarily translate into higher utility for the attacker when compared to the sine trigger. The mixed strategies and varying effectiveness across different α_{tr} values suggest that

Table 6 BG_{Min} with sine trigger: performance at the equilibrium

α_{tr}	ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.05	0.204	0.112	0.786	0.838	0.158	0.812	0.0
0.1	0.105	0.109	0.968	0.960	0.107	0.964	− 0.1039
0.2	0.105	0.119	0.964	0.950	0.112	0.957	− 0.1062
0.3	0.093	0.103	0.905	0.909	0.098	0.907	− 0.0978
0.4	0.102	0.115	0.948	0.938	0.108	0.943	− 0.1058
0.5	0.106	0.115	0.964	0.957	0.111	0.961	− 0.1049
0.6	0.103	0.096	0.926	0.898	0.099	0.912	− 0.0939
0.7	0.102	0.117	0.953	0.922	0.109	0.938	− 0.1043
0.8	0.106	0.116	0.964	0.958	0.111	0.961	− 0.1051
0.9	0.129	0.109	0.770	0.801	0.119	0.786	0.0
1.0	0.103	0.101	0.861	0.861	0.102	0.861	0.0



(a) $\alpha_{tr} = 0.05$ (b) $\alpha_{tr} = 0.5$ (c) $\alpha_{tr} = 1.0$

Fig. 4 BG_{Min}: u_A with ramp trigger and different α_{tr}

Table 7 Equilibrium point of BG_{Min}: $\alpha_{tr} = 0.05$ and ramp trigger

Profiles	Parameters	Equilibrium
Attacker	$S_A^* = (\Delta_{tr}, \Delta_{ts})$	(0.01, 0.01)
	$Pr(S_A^*)$	1.0
Defender	$S_D^* = (\Delta_{def})$	0.5
	$Pr(S_D^*)$	1.0
Utility	$u_A^* = -u_D^* = u^*$	0.0

Table 8 Strategy profiles of BG_{Min}, given $\alpha_{tr} = 0.5$ and ramp trigger

Profiles	Parameters	Equilibria	
Attacker	$S_A^* = (\Delta_{tr}, \Delta_{ts})$	(0.5, 0.03)	(0.5, 0.5)
	$Pr(S_A^*)$	0.557	0.443
Defender	$S_D^* = (\Delta_{def})$	0.4	0.5
	$Pr(S_D^*)$	0.422	0.578
Utility	$u_A^* = -u_D^* = u^* = -0.1056$		

Table 9 Equilibrium point of BG_{Min}: $\alpha_{tr} = 1.0$ and ramp trigger

Profiles	Parameters	Equilibrium
Attacker	$S_A^* = (\Delta_{tr}, \Delta_{ts})$	(0.01, 0.01)
	$Pr(S_A^*)$	1.0
Defender	$S_D^* = (\Delta_{def})$	0.5
	$Pr(S_D^*)$	1.0
Utility	$u_A^* = -u_D^* = u^*$	0.0

despite the flexibility offered by the ramp trigger, the game remains balanced, with neither player achieving a definitive advantage.

4.2.3 Trigger mismatch

Analysis of the utility matrices We now focus on the scenario where there is a mismatch between the attacker's and defender's trigger types, specifically when

Table 10 BG_{Min} with ramp trigger: performance at the equilibrium

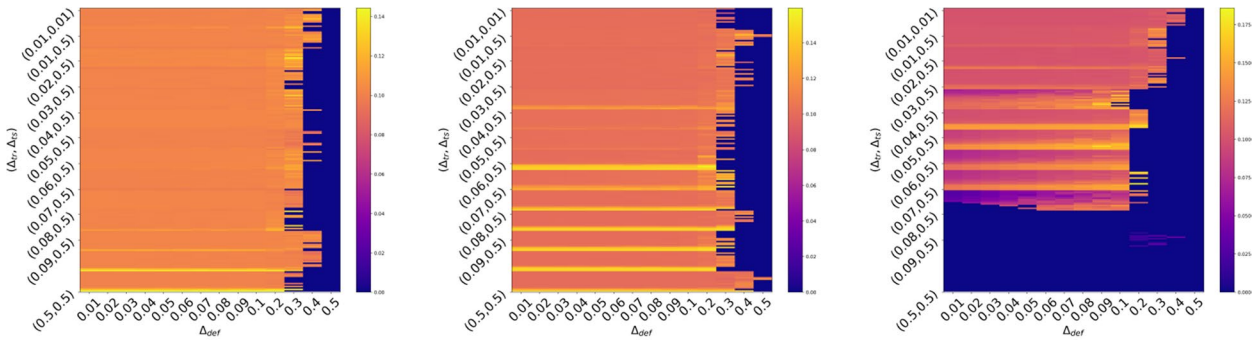
α_{tr}	ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.05	0.204	0.183	0.771	0.697	0.1935	0.7340	0.0
0.1	0.100	0.098	0.887	0.914	0.0990	0.9005	– 0.0989
0.2	0.092	0.085	0.654	0.595	0.0885	0.6245	0.0
0.3	0.104	0.150	0.078	0.747	0.1270	0.4125	0.0
0.4	0.092	0.095	0.915	0.934	0.0935	0.9245	– 0.0936
0.5	0.098	0.113	0.958	0.934	0.1055	0.9460	– 0.1056
0.6	0.100	0.103	0.978	0.969	0.1015	0.9735	– 0.1013
0.7	0.088	0.100	0.953	0.937	0.0940	0.9450	– 0.0937
0.8	0.091	0.109	0.925	0.921	0.1000	0.9230	– 0.1002
0.9	0.100	0.109	0.964	0.947	0.1045	0.9555	– 0.1040
1.0	0.108	0.116	0.810	0.888	0.1120	0.8490	0.0

the attacker uses a sine trigger while the defender reconstructs a ramp trigger.

This mismatch introduces additional challenges for the defender who, lacking precise knowledge of the attacker's trigger, may adopt less optimal strategies. Despite this uncertainty, the general behavior of utilities in the BG_{Min} game remains consistent with the matched trigger cases (see Fig. 5). However, the defender's incorrect trigger assumption broadens the strategy profiles, leading to higher attacker utility as the defender struggles to effectively counter the attack. This results in strategic behaviors that, while similar to previous scenarios, are slightly more advantageous for the attacker.

Despite the mismatch, the sine trigger continues to be advantageous for the attacker, leading to high utility regions in the utility matrices (light orange and yellow zones). This is particularly evident in scenarios with $\alpha_{tr} = 0.5$, where the CDA remains relatively high for both clean and poisoned samples ($CDA_{cb} = 0.959$, $CDA_{cp} = 0.951$), but the ASR also stays notable at 0.109. As α_{tr} increases to 1.0, the defender's performance declines further, with CDA values dropping to $CDA_{cb} = 0.873$ and $CDA_{cp} = 0.783$, while the ASR remains high at 0.113. These outcomes illustrate that a trigger mismatch, though not drastically reducing defender performance due to the similarity between sine and ramp triggers, still weakens the defender's effectiveness and favors the attacker (see Table 14).

Analysis of the equilibrium strategies The equilibrium strategies in the presence of a trigger mismatch show that both players must adjust their approaches due to the incorrect assumption made by the defender. At a low poisoning ratio ($\alpha_{tr} = 0.05$), the attacker maintains a conservative strategy with the equilibrium profile $(\Delta_{tr}, \Delta_{ts}) = (0.01, 0.01)$, while the defender, unaware of the mismatch, continues to apply a high defense power $\Delta_{def} = 0.5$. This results in a utility of zero for both players, indicating a balance where the defender's incorrect assumption does not drastically change the outcome (see Table 11).



(a) $\alpha_{tr} = 0.05$ (b) $\alpha_{tr} = 0.5$ (c) $\alpha_{tr} = 1.0$

Fig. 5 BG_{Min}: u_A with trigger mismatch and different α_{tr}

Table 11 Equilibrium points of BG_{Min}: $\alpha_{\text{tr}} = 0.05$ with trigger mismatch

Profiles	Parameters	Equilibrium
Attacker	$S_A^* = (\Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.01, 0.01)
	$Pr(S_A^*)$	1.0
Defender	$S_D^* = (\Delta_{\text{def}})$	0.5
	$Pr(S_D^*)$	1.0
Utility	$u_A^* = -u_D^* = u^*$	0.0

As the poisoning ratio increases to $\alpha_{\text{tr}} = 0.5$, the attacker begins to exploit the defender's mistake more effectively by adopting a complex mixed strategy, with a notable preference for $(\Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.5, 0.06)$, alongside other strategies such as (0.02, 0.05), (0.2, 0.5), and (0.5, 0.5). This approach reflects the attacker's attempt to capitalize on the defender's uncertainty about the trigger type. In response, the defender, still operating under the assumption of a ramp trigger, employs a mixed strategy across different defense powers, notably selecting $\Delta_{\text{def}} = 0.4$ with a probability of 0.6155 and $\Delta_{\text{def}} = 0.5$ with a probability of 0.2756. The resulting utility slightly favors the defender, with $u^* = -0.1047$, yet the presence of mixed strategies from both sides indicates ongoing uncertainty and a lack of dominance by either player (see Table 12).

At the highest poisoning ratio ($\alpha_{\text{tr}} = 1.0$), the attacker's strategy reverts to a pure strategy Nash equilibrium, favoring minimal trigger powers $(\Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.01, 0.01)$ to maintain the backdoor's persistence while minimizing detection risk. Despite the mismatch in trigger types, the defender's strong defense at $\Delta_{\text{def}} = 0.5$ neutralizes the attack, resulting in zero utility for both players, similar to what is observed in pure trigger cases. While the trigger mismatch offers the attacker some advantage, particularly in sustaining a higher ASR, it does not significantly alter the overall utility compared to scenarios where the trigger types match (see Table 13). The performance metrics confirm this outcome, indicating that the overall dynamics remain balanced even with the mismatch (see Table 14).

Table 12 Equilibrium point of BG_{Min}: $\alpha_{\text{tr}} = 0.5$ with trigger mismatch

Profiles	Parameters	Equilibria			
Attacker	$S_A^* = (\Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.02, 0.05)	(0.2, 0.5)	(0.5, 0.06)	(0.5, 0.5)
	$Pr(S_A^*)$	0.2594	0.0557	0.6400	0.0449
Defender	$S_D^* = (\Delta_{\text{def}})$	0.2	0.3	0.4	0.5
	$Pr(S_D^*)$	0.6155	0.0975	0.0114	0.2756
Utility	$u_A^* = -u_D^* = u^* = -0.1047$				

Table 13 Equilibrium point of BG_{Min}: $\alpha_{\text{tr}} = 1.0$ with trigger mismatch

Profiles	Parameters	Equilibrium
Attacker	$S_A^* = (\Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.01, 0.01)
	$Pr(S_A^*)$	1.0
Defender	$S_D^* = (\Delta_{\text{def}})$	0.5
	$Pr(S_D^*)$	1.0
Utility	$u_A^* = -u_D^* = u^*$	0.0

4.3 Results with BG_{Int}

In this section, we analyze the results obtained from the BG_{Int} game setup, where the attacker has full control over the dataset and can decide both the fraction of samples to poison and the power of the attack.

4.3.1 Sinusoidal trigger

Analysis of the utility matrix In the BG_{Int} game with a sine trigger, the attacker has full access to the dataset and can strategically decide the fraction of samples to poison alongside their attack power. The utility matrix for the attacker in BG_{Int}, as illustrated in Fig. 6, exhibits behavior similar to that observed in the BG_{Min} game, particularly showing low attacker utility (u_A) for columns where the defender's power $\Delta_{\text{def}} > 0.3$. This similarity suggests that increasing the defender's defense power significantly hampers the attacker's ability to execute a successful backdoor attack, especially when the defense power is strong enough to detect and mitigate the effects of the trigger. Additionally, the attacker's utility increases with the poisoning ratio (α_{tr}). Higher utilities are observed for more rows of the $(\Delta_{\text{tr}}, \Delta_{\text{ts}})$ profiles since the attacker has a greater flexibility in balancing strategies regarding the amount of poisoned samples and the trigger power. This flexibility allows the attacker to optimize their strategy more effectively, resulting in more high-utility rows (light orange and yellow) in the utility matrix.

The attacker's utility in the BG_{Int} game with a sine trigger increases with the poisoning ratio α_{tr}, indicating that the more samples the attacker can poison, the greater their potential utility. A higher poisoning ratio improves the likelihood that the backdoor trigger will be effective during the testing phase, thereby enhancing the attack's success rate (ASR). However, as α_{tr} increases excessively (i.e., 0.9 and above), the attack becomes more apparent to the defender, resulting in the appearance of low-utility (blue) rows at the bottom of the utility matrix, where $u_A = 0$. These low-utility strategies should be avoided by the attacker.

Table 14 BG_{Min} with trigger mismatch: performance at the equilibrium

α_{tr}	ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.05	0.131	0.130	0.832	0.805	0.131	0.819	0.0
0.1	0.103	0.098	0.825	0.683	0.101	0.754	0.0
0.2	0.102	0.117	0.971	0.945	0.110	0.958	− 0.1052
0.3	0.099	0.128	0.902	0.858	0.114	0.880	0.0
0.4	0.102	0.105	0.975	0.972	0.104	0.974	− 0.1021
0.5	0.105	0.112	0.959	0.951	0.109	0.955	− 0.1047
0.6	0.102	0.102	0.909	0.924	0.102	0.917	− 0.0996
0.7	0.099	0.110	0.973	0.959	0.105	0.966	− 0.1012
0.8	0.104	0.134	0.956	0.910	0.119	0.933	− 0.1169
0.9	0.090	0.098	0.918	0.883	0.094	0.901	− 0.0943
1.0	0.098	0.128	0.873	0.783	0.113	0.828	0.0

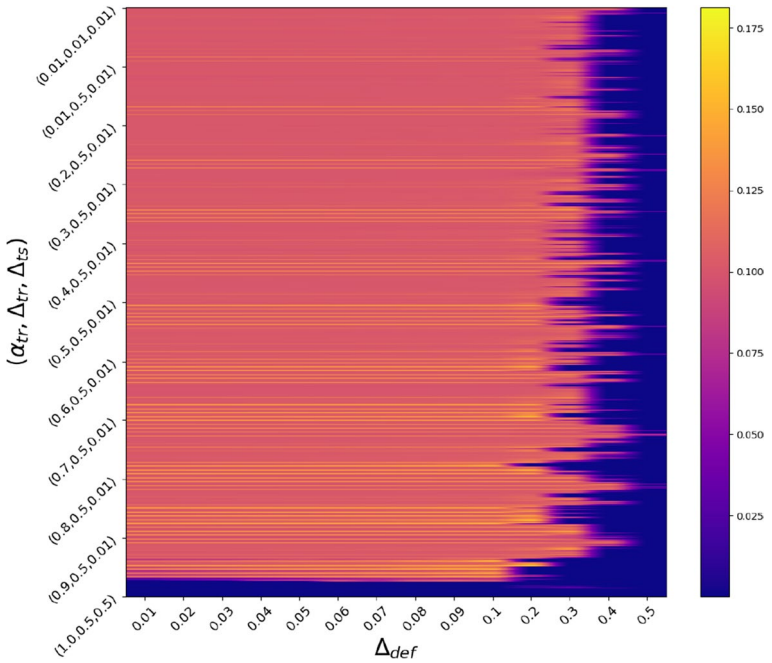


Fig. 6 BG_{Int}: u_A and sine trigger

Unlike in the BG_{Min} game, there is no dominant strategy for either player in BG_{Int}. This absence of a dominant strategy likely stems from the variable α_{tr}, which was fixed in BG_{Min}. Consequently, the utility matrix in BG_{Int} does not contain fully low-utility (purple) columns, leading both players to adopt mixed strategies that involve different power levels of their strategy profiles.

Analysis of the equilibrium strategies The equilibrium analysis in the BG_{Int} game reveals that the attacker employs a mixed strategy with a strong preference for the profile $(\alpha_{tr}, \Delta_{tr}, \Delta_{ts}) = (0.2, 0.2, 0.05)$, selected with a high probability of 0.8504. This choice indicates that the attacker strategically balances the poisoning ratio with moderate trigger powers to optimize the ASR while minimizing the risk of detection. The remaining strategies have significantly lower probabilities, serving as fallback options or less favored strategies to maintain uncertainty and keep the defender off balance. On the other hand, the defender's strategy distribution is more dispersed, showing a cautious approach to selecting defense powers. The defender frequently opts for lower Δ_{def} values, particularly 0.04 and 0.06, which are chosen with probabilities of 0.3914 and 0.3329, respectively. This suggests that even

minimal defense efforts can effectively neutralize the attack under certain conditions, reflecting the defender's intent to maintain a high level of defense across various scenarios (see Table 15).

At equilibrium, the performance metrics show that the attacker achieves a modest ASR of 0.11635, while the defender successfully preserves high CDA values, underscoring the effectiveness of the defender's strategy. This balance results in a utility of $u^* = -0.1078$ for both players, highlighting the competitive nature of the game where neither player can dominate completely. The attacker's dominant strategy allows them to maintain a reasonable ASR, while the defender's mixed strategy effectively mitigates the attack's impact, demonstrating the complex strategic interplay between the two players (see Table 16).

4.3.2 Ramp trigger

Analysis of the utility matrices The utility matrices for the attacker in the BG_{Int} game under the ramp trigger scenario demonstrate behaviors similar to the ones observed with the sine trigger. As shown in Fig. 7, the attacker does not have a clearly dominant strategy, reflecting the complexity of the interaction between the attacker's choices and the defender's countermeasures. The utility for both the attacker and the defender at equilibrium is $u^* = -0.1059$, suggesting a near-equilibrium state where neither player can significantly improve their position unilaterally (see Table 17).

Analysis of the equilibrium strategies Under the ramp trigger, the attacker's mixed strategy is more varied, including three profiles: $(\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.05, 0.05, 0.02)$, (0.5, 0.5, 0.1), and (0.8, 0.4, 0.5). These profiles are chosen with probabilities of 0.0666, 0.7529, and 0.1805, respectively. This distribution indicates that the attacker might adopt a higher trigger power when paired with a significant poisoning ratio, reflecting a more aggressive strategy compared to the sine trigger scenario. The defender's strategy set includes a preference for $\Delta_{\text{def}} = 0.4$, which is chosen with the highest probability, reflecting an optimal balance between resource allocation and defensive effectiveness.

The equilibrium performance metrics for the ramp trigger indicate an ASR of 0.111 and a CDA of 0.942, slightly lower than the sine trigger case, but still reflecting an effective defense (see Table 18).

4.3.3 Trigger mismatch

Analysis of the utility matrices In the trigger mismatch scenario, where the attacker deploys a sine trigger and the defender mistakenly counters with a ramp trigger, the dynamics of the BG_{Int} game reveal notable variations in strategy effectiveness compared to matching trigger cases.

The attacker's utility matrix (Fig. 8) demonstrates that despite the defender's misjudgment, the attacker maintains high utility across several strategy profiles, indicating the robustness of the sine trigger. This resilience is particularly evident when the defender's defense power (Δ_{def}) is not optimally aligned with the actual trigger, allowing the attacker to sustain an advantage similar to that in matched trigger scenarios. The flexibility afforded by the variable poisoning ratio α_{tr} in the BG_{Int} setup further enhances the attacker's ability to

Table 15 Equilibrium point of BG_{Int} with sine trigger

Profiles	Parameters	Equilibria				
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.2, 0.2, 0.05)	(0.3, 0.4, 0.5)	(0.9, 0.09, 0.3)	(0.9, 0.09, 0.5)	-
	$Pr(S_A^*)$	0.8504	0.1225	0.0127	0.0144	-
Defender	$S_D^* = (\Delta_{\text{def}})$	0.04	0.06	0.3	0.4	0.5
	$Pr(S_D^*)$	0.3914	0.3329	0.0116	0.0014	0.2627
Utility		$u_A^* = -u_D^* = u^* = -0.1078$				

Table 16 BG_{Int} with sine trigger: performance at the equilibrium

ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.1088	0.1239	0.9617	0.9515	0.11635	0.9566	− 0.1078

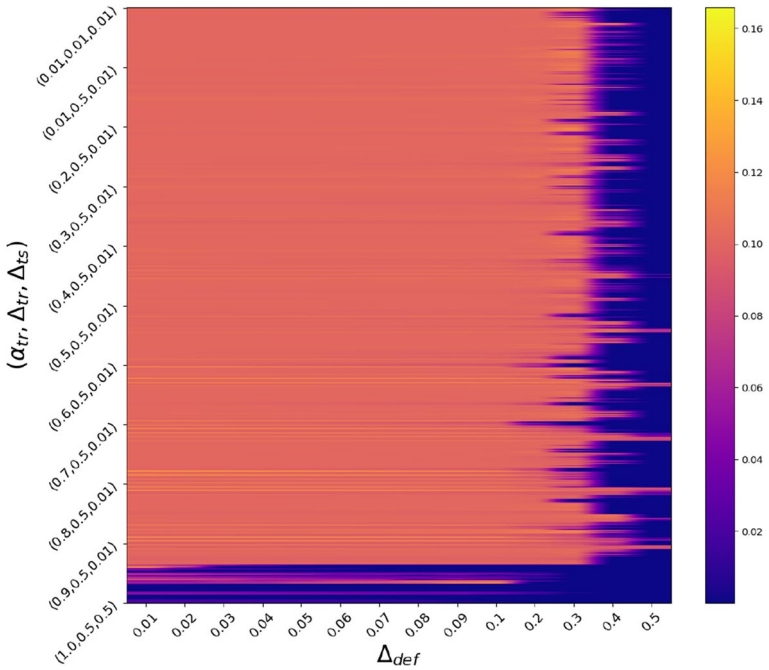


Fig. 7 BG_{Int}: u_A and ramp trigger

Table 17 Equilibrium point of BG_{Int} with ramp trigger

Profiles	Parameters	Equilibria			
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.05, 0.05, 0.02)	(0.5, 0.5, 0.1)	(0.8, 0.4, 0.5)	-
	$Pr(S_A^*)$	0.0666	0.7529	0.1805	-
Defender	$S_D^* = (\Delta_{\text{def}})$	0.2	0.3	0.4	0.5
	$Pr(S_D^*)$	0.0152	0.0438	0.7595	0.1815
Utility	$u_A^* = -u_D^* = u^* = -0.1059$				

Table 18 BG_{Int} with ramp trigger: performance at the equilibrium

ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.108	0.114	0.941	0.943	0.111	0.942	− 0.1059

optimize their strategy, as reflected in the frequent light orange/yellow zones within the utility matrix in Fig. 8. In contrast to the BG_{Min} scenario, where the fixed α_{tr} constrained the attacker's options, the BG_{Int} game underscores the critical role of accurate trigger identification; the defender's lack of knowledge about the exact trigger leads to suboptimal defense strategies, providing the attacker with more opportunities to exploit the mismatch.

Analysis of the equilibrium strategies In the trigger mismatch scenario for the BG_{Int} game, where the attacker uses a sine trigger while the defender assumes a ramp trigger, the attacker's strategy heavily favors the profile $(\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.8, 0.5, 0.4)$, selected with a high probability of 0.8652. This choice likely reflects the attacker's intent to exploit the mismatch by increasing both the poisoning ratio and trigger powers to maximize the impact. The defender, on the other hand, adopts a more distributed approach with defense power levels $\Delta_{\text{def}} = 0.3$, $\Delta_{\text{def}} = 0.4$, and $\Delta_{\text{def}} = 0.5$, with probabilities of 0.4984, 0.31, and 0.1916, respectively. This strategy distribution indicates the defender's uncertainty about the exact nature of the attack, leading to a varied defense approach that aims to balance the potential threat while minimizing the impact on CDA (see Table 19).

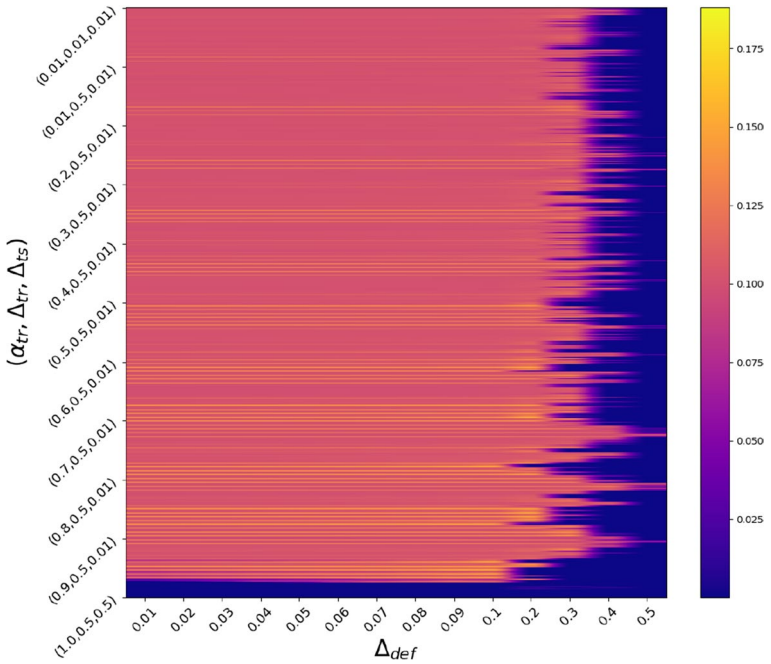


Fig. 8 BG_{Int}: u_A with trigger mismatch

Table 19 Equilibrium point of BG_{Int} with trigger mismatch

Profiles	Parameters	Equilibria		
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.8, 0.4, 0.3)	(0.8, 0.4, 0.5)	(0.8, 0.5, 0.4)
	$Pr(S_A^*)$	0.1021	0.0327	0.8652
Defender	$S_D^* = (\Delta_{\text{def}})$	0.3	0.4	0.5
	$Pr(S_D^*)$	0.4984	0.31	0.1916
Utility		$u_A^* = -u_D^* = u^* = -0.1193$		

The performance metrics at equilibrium, as shown in Table 20, reveal an ASR of 0.12025 and a CDA of 0.944, with a utility value of $u^* = -0.1193$ for both players. These results suggest that the trigger mismatch scenario introduces additional challenges for the defender, slightly increasing the attack's effectiveness compared to uniform trigger scenarios. The mismatch complicates the defender's ability to predict the attacker's actions accurately, necessitating a more cautious and varied defense strategy to maintain effective protection against the attack.

4.4 Results with BG_{Max}

In this section, we present the results from the BG_{Max} game, our most comprehensive game setup, where the defender's decision-making incorporates the probability of defending against an incoming sample, denoted as α_{def}. This setup extends the framework of the BG_{Int} game, providing the defender with greater flexibility to balance the likelihood of an attack and the strength of their defense.

4.4.1 Sinusoidal trigger

Analysis of the utility matrix The utility matrix for the BG_{Max} game under the sinusoidal trigger scenario shows a pattern consistent with that observed in the BG_{Int} game. However, the repetition across different defender decision ratios (α_{def}) becomes more pronounced as α_{def} increases (see Fig. 9). The attacker's utility decreases significantly in the lower rows and columns where Δ_{def} and α_{def} are high, reflecting a more conspicuous attack that is easily detected by the defender. Consequently, this results in lower utility scores for the attacker in these regions, as indicated by the dark purple areas in the matrix, where

Table 20 BG_{Int} with trigger mismatch: performance at the equilibrium

ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.1008	0.1397	0.973	0.915	0.12025	0.944	− 0.1193

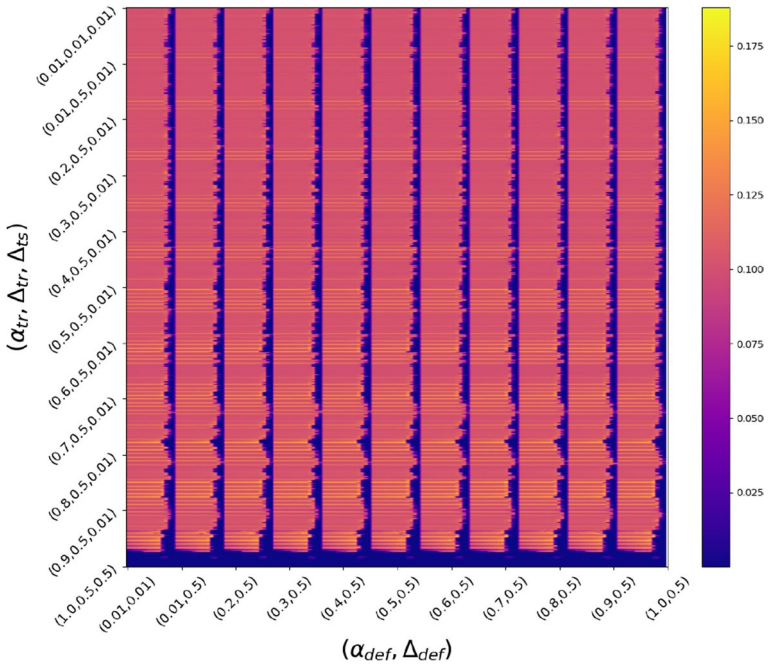


Fig. 9 BG_{Max}: u_A and sine trigger

$u_A = 0$. The absence of a dominant strategy in this scenario leads both players to adopt mixed strategies at the Nash equilibrium.

Analysis of the equilibrium strategies In the sine trigger scenario, the attacker employs various strategies with significant probabilities assigned to several profiles, particularly $(\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.4, 0.2, 0.09)$ and $(0.7,\ 0.5,\ 0.06)$, which are selected with probabilities of 0.2142 and 0.212, respectively (see Table 21). The defender, meanwhile, distributes their strategy across multiple profiles, with a preference for higher α_{def} values. The strategies (0.8, 0.5) and (0.9, 0.5) are chosen most frequently, reflecting the defender's focus on maintaining strong defense as α_{def} increases. The performance metrics at equilibrium, as shown in Table 22, indicate an ASR of 0.10775 and CDA values of 0.9034 and 0.8864 for clean and poisoned data, respectively. The equilibrium utility for both players is $u^* = -0.0636$, suggesting that while the defender maintains effective defense, the attack's subtlety is key to its limited success.

4.4.2 Ramp trigger

Analysis of the utility matrix The utility matrix for the BG_{Max} game under the ramp trigger scenario reveals

Table 21 Equilibrium points of BG_{Max} with sine trigger

Profiles	Parameters	Equilibria				
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.4,0.2,0.09)	(0.7,0.5,0.02)	(0.7,0.5,0.06)	(0.7,0.5,0.09)	(0.7,0.5,0.2)
	$Pr(S_A^*)$	0.2142	0.0042	0.212	0.1519	0.1838
Defender	$S_D^* = (\alpha_{\text{def}}, \Delta_{\text{def}})$	(0.05,0.5)	(0.1,0.5)	(0.2,0.5)	(0.4,0.5)	(0.6,0.5)
	$Pr(S_D^*)$	0.0334	0.1097	0.0484	0.0167	0.1001
Utility		$u_A^* = -u_D^* = u^* = -0.0636$				
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.7,0.5,0.3)	(0.8,0.4,0.1)	(0.8,0.5,0.02)	(0.8,0.5,0.03)	
	$Pr(S_A^*)$	0.0736	0.0011	0.0154	0.1438	
Defender	$S_D^* = (\alpha_{\text{def}}, \Delta_{\text{def}})$	(0.7,0.5)	(0.8,0.5)	(0.9,0.5)	(1,0.5)	
	$Pr(S_D^*)$	0.1307	0.2439	0.2013	0.1158	
Utility		$u_A^* = -u_D^* = u^* = -0.0636$				

Table 22 BG_{Max} with sine trigger: performance at the equilibrium

ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.1075	0.108	0.9034	0.8864	0.10775	0.8949	− 0.0636

more dense orange areas compared to the sine trigger, indicating that the ramp trigger conveys less information in poisoned samples (see Fig. 10). This results in a lower maximum utility for the attacker, as evidenced by the matrix's structure, which leads to a more cautious defensive strategy from the defender. The lack of dominant strategy in this scenario compels both players to adopt mixed strategies, focused on optimizing their outcomes while minimizing the risk of detection.

Analysis of the equilibrium strategies In the ramp trigger scenario, the attacker's equilibrium strategies include profiles such as $(\alpha_{tr}, \Delta_{tr}, \Delta_{ts}) = (0.5, 0.5, 0.08)$, which is heavily favored with a probability of 0.5555 (see Table 23). The defender's strategy distribution shifts towards moderate defense levels, with $\alpha_{def} = 0.3$ being the most probable at 0.3447. The performance metrics at equilibrium, presented in Table 24, show CDA_{cb} and CDA_{cp} values of 0.917 and 0.914 for clean and poisoned data, respectively, with an ASR of 0.09705 and a utility of $u^* = -0.0966$. These results highlight the ramp trigger's less aggressive nature, leading to a more balanced scenario where neither player can significantly improve their outcomes.

4.4.3 Trigger mismatch

Analysis of the utility matrix In the trigger mismatch scenario, where the attacker uses a sine trigger while the defender deploys a ramp trigger, the utility dynamics shift significantly, introducing additional complexity into the game.

Although the behavior of the utilities for the same strategy sets remains consistent with scenarios involving uniform triggers, the mismatch complicates the defender's ability to effectively counter the attack. This complexity is reflected in the attacker's utility matrix (Fig. 11), where more strategy profiles yield high utility for the attacker due to the defender's lack of precise knowledge about the trigger in use. The increased frequency of orange and yellow zones in the utility matrix suggests that the attacker benefits from the defender's uncertainty, allowing them to exploit the mismatch more effectively.

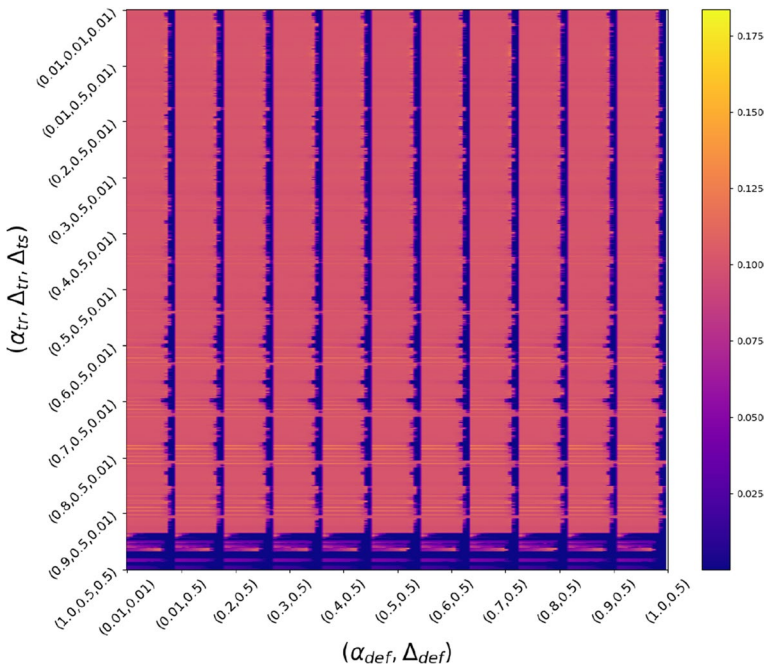


Fig. 10 BG_{Max}: u_A and ramp trigger

Table 23 Equilibrium points of BG_{Max} with ramp trigger

Profiles	Parameters	Equilibria				
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.5,0.5,0.03)	(0.5,0.5,0.06)	(0.5,0.5,0.08)	(0.6,0.5,0.2)	(0.7,0.5,0.5)
	$Pr(S_A^*)$	0.0072	0.0128	0.5555	0.0189	0.1036
Defender	$S_D^* = (\alpha_{\text{def}}, \Delta_{\text{def}})$	(0.05,0.5)	(0.1,0.5)	(0.2,0.5)	(0.3,0.5)	(0.4,0.5)
	$Pr(S_D^*)$	0.0178	0.1636	0.299	0.3447	0.0249
Utility	$u_A^* = -u_D^* = u^* = -0.0966$					
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.8,0.4,0.04)	(0.8,0.4,0.08)	(0.8,0.4,0.02)	(0.8,0.4,0.4)	
	$Pr(S_A^*)$	0.0184	0.1384	0.1361	0.0091	
Defender	$S_D^* = (\alpha_{\text{def}}, \Delta_{\text{def}})$	(0.5,0.5)	(0.6,0.5)	(0.7,0.5)	(1,0.5)	
	$Pr(S_D^*)$	0.0356	0.0081	0.0924	0.0139	
Utility	$u_A^* = -u_D^* = u^* = -0.0966$					

Table 24 BG_{Max} with ramp trigger: performance at the equilibrium

ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.0951	0.099	0.917	0.914	0.09705	0.9155	− 0.0966

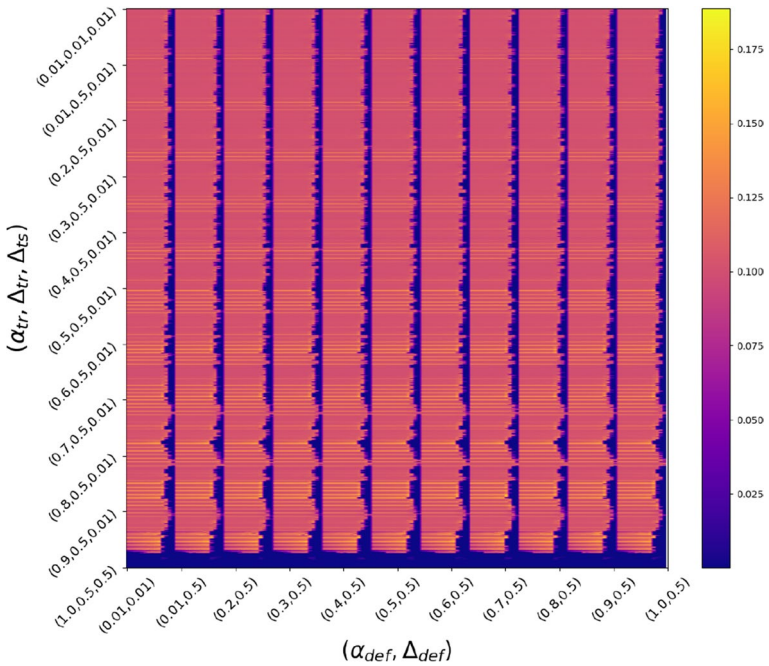


Fig. 11 BG_{Max}: u_A with trigger mismatch

This scenario underscores the importance of the defender's ability to correctly anticipate or identify the type of trigger used by the attacker, as misjudgment leads to suboptimal defense strategies and a broader distribution of high-utility strategies for the attacker.

Analysis of the equilibrium strategies In the trigger mismatch scenario with BG_{Max}, the attacker shows a strong preference for the strategy $(\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}}) = (0.7, 0.5, 0.05)$, with a dominant probability of 0.7227 (see Table 25), likely reflecting an attempt to maximize the attack success rate (ASR) while maintaining a relatively low trigger power to evade detection. Conversely, the defender's strategy is more distributed across various profiles, with the most probable being $(\alpha_{\text{def}}, \Delta_{\text{def}}) = (1, 0.5)$, selected with a probability of 0.2111. This broader distribution

Table 25 Equilibrium point of BG_{Max} with trigger mismatch

Profiles	Parameters	Equilibria				
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.7,0.5,0.05)	(0.7,0.5,0.1)	(0.8,0.4,0.05)	(0.8,0.4,0.1)	(0.8,0.5,0.01)
	$Pr(S_A^*)$	0.7227	0.0024	0.0361	0.0129	0.0050
Defender	$S_D^* = (\alpha_{\text{def}}, \Delta_{\text{def}})$	(0.1,0.5)	(0.2,0.5)	(0.3,0.5)	(0.4,0.5)	(0.5,0.5)
	$Pr(S_D^*)$	0.0176	0.1366	0.0209	0.1174	0.0741
Utility		$u_A^* = -u_D^* = u^* = -0.0953$				
Attacker	$S_A^* = (\alpha_{\text{tr}}, \Delta_{\text{tr}}, \Delta_{\text{ts}})$	(0.8,0.5,0.02)	(0.8,0.5,0.05)	(0.8,0.5,0.07)	(0.8,0.5,0.4)	
	$Pr(S_A^*)$	0.0067	0.0129	0.0058	0.1805	
Defender	$S_D^* = (\alpha_{\text{def}}, \Delta_{\text{def}})$	(0.7,0.5)	(0.8,0.5)	(0.9,0.5)	(1,0.5)	
	$Pr(S_D^*)$	0.0276	0.0118	0.0163	0.2111	
Utility	$u_A^* = -u_D^* = u^* = -0.0953$					

Table 26 BG_{Max} with trigger mismatch: performance at the equilibrium

ASR_{cb}	ASR_{cp}	CDA_{cb}	CDA_{cp}	***ASR***	***CDA***	u^*
0.0915	0.112	0.935	0.8962	0.1018	0.9156	− 0.0953

indicates the defender's difficulty in effectively countering an unexpected trigger type. The performance metrics, as shown in Table 26, demonstrate that the defender's performance is compromised compared to uniform trigger cases, with CDA values dropping to $CDA_{\text{cb}} = 0.935$ and $CDA_{\text{cp}} = 0.8962$, and an ASR of 0.1018. The utility at equilibrium is $u^* = -0.0953$, highlighting the advantage gained by the attacker in the mismatched trigger scenario due to the defender's increased challenge in adapting their strategy.

5 Conclusions and future works

In this paper, we proposed a novel game-theoretic framework to model the interaction between an attacker and a defender in the context of a DNN backdoor attack. Our framework introduced a utility function that integrates clean data accuracy (CDA) and attack success rate (ASR), formulating the backdoor attack as a two-player zero-sum game, and provided flexibility with respect to the level of control afforded to each player. Through numerical simulations, we demonstrated the effectiveness of the proposed framework and found insightful equilibrium strategies, thus evaluating player's performances at the equilibrium.

We explored three different game setups with varying levels of control: BG_{Min}, BG_{Int}, and BG_{Max}. In BG_{Min}, the attacker controlled the trigger power during training and testing, while the defender controlled the trigger removal power during testing. BG_{Int} extended this by allowing the attacker to control the poisoning ratio during training. BG_{Max} further allowed the defender to decide whether to apply the defense to an input sample, adding a decision probability into their strategy set. These setups were chosen to reflect different real-world scenarios, ranging from minimal to maximum strategic complexity, thus providing a comprehensive analysis of attacker-defender dynamics.

A key finding was the paradox faced by the attacker, where increasing the attack power or poisoning ratio improved the attack's success but also made the attack more detectable and the trigger easier to estimate by the defender. This led to the attacker having to balance between delivering sufficient information to the DNN to learn the backdoor and avoiding revealing too much information to the defender. As the attacker's freedom increased (from BG_{Min} to BG_{Max}), the strategies employed became more sophisticated, often involving mixed strategies to maintain unpredictability. This increased the complexity of the game and required the defender to adapt by also employing mixed strategies, particularly in scenarios where no single strategy was dominant.

In all cases, the defender aimed to maximize the clean data accuracy, while minimizing the attack success rate. The defender's strategies involved using high trigger removal power when the attack was more apparent and balancing between different levels of defense power to counter the attacker's mixed strategies. The effectiveness of the defender's strategies was evident in maintaining a high clean data accuracy, even in the face of sophisticated attacks. Other important findings include the observation that fully utilizing one's capacity is a suboptimal strategy for either attacker or defender when maximizing their utilities. The attacker must find a balance between

inducing errors and minimizing information conveyed to the defender, while the defender must minimize attack risks while preserving benign samples.

Future research could extend this framework in several key directions. One significant area for expansion is multi-agent settings, such as scenarios with multiple attackers or the involvement of third parties. For instance, in federated learning, where attackers might strike during the same training round or communicate with each other, the dynamics could evolve into more complex forms like sequential games, Bayesian games, or games with incomplete information. These variations would add layers of complexity to the model, requiring substantial modifications to the game definition and framework, but they would also provide a more accurate reflection of real-world conditions. Further extensions could explore dynamic strategies, where attackers and defenders adapt over time based on ongoing observations, offering deeper insights into practical applications. Additionally, investigating the level of information exchanged between players could quantify strategic advantages from an information-theoretic perspective. Incorporating more varied attack and defense mechanisms, especially in contexts like transfer learning and federated learning, would enhance the framework's robustness and generalizability. Testing the framework on larger datasets and more complex models, such as those in autonomous driving and healthcare, would be essential for assessing scalability and practicality. Finally, exploring collaborative defense approaches, where multiple defense techniques share information to counteract a common attacker, could refine the game-theoretic approach, improving the security of deep neural networks across various domains.

In future work, we also plan to explore non-zero-sum game formulations, which better capture the complexities of real-world adversarial interactions. For example, in federated learning, attackers might aim not only to disrupt a system but also to avoid detection, leading to scenarios where both parties incur non-opposing losses or gains. Additionally, incorporating third-party entities or externalities could necessitate more sophisticated models, such as cooperative or Bayesian games, to account for variable total payoffs. These extensions will provide a more comprehensive understanding of the strategic interplay between attackers and defenders in diverse adversarial settings.

Acknowledgements
Not applicable.

Authors' contributions
Kassem proposed and developed the idea, supervised the research, designed and conducted the experiments, analyzed the data, and wrote the manuscript. Quentin contributed to the development and review of the idea, the execution, analysis, and interpretation of the experimental work, and reviewed the manuscript. Wassim and Teddy assisted in reviewing the manuscript.

Funding
This work is partially supported by the CYBAILE industrial chair, led by Inserm with support from the Brittany Regional Council. Additionally, this publication is partly funded by resources from ANR/AID under the Chaire SAIDA ANR-20-CHIA-0011 project.

Availability of data and materials
The datasets generated and/or analysed during the current study are available in the MNIST repository, http://yann.lecun.com/exdb/mnist/.

Code availability
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Author details
[1]Institut National de la Santé et de la Recherche Médicale (INSERM), Brest, France. [2]Thales Digital Identity & Security (DIS), La Ciotat, France. [3]Institut National de Recherche en Informatique et en Automatique (INRIA), Rennes, France. [4]Technology Innovation Institute (TII), Masdar City, Abu Dhabi, United Arab Emirates.

Received: 14 June 2024 Accepted: 20 September 2024
Published online: 15 October 2024

References

1. A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, D. Andina, Deep Learning for Computer Vision: A Brief Review. Intell Neuroscience **2018**(1687-5265), 13 (2018) https://doi.org/10.1155/2018/7068349
2. S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu, A survey of deep learning techniques for autonomous driving. J. Field Robot. **37**(3), 362–386 (2020)
3. J.B. Heaton, N.G. Polson, J.H. Witte, Deep learning for finance: Deep portfolios. Appl. Stoch. Model. Bus. Ind. **33**(1), 3–12 (2017)
4. R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, Deep learning for healthcare: Review, opportunities and challenges. Brief. Bioinform. **19**(6), 1236–1246 (2018)
5. P.P. Shinde, S. Shah, in *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, A review of machine learning and deep learning applications (IEEE, 2018), pp. 1–6
6. B. Wu, Z. Zhu, L. Liu, Q. Liu, Z. He, S. Lyu, Attacks in adversarial machine learning: A systematic survey from the life-cycle perspective (2024). https://arxiv.org/abs/2302.09457
7. N.M. Gürel, X. Qi, L. Rimanic, C. Zhang, B. Li, in *International Conference on Machine Learning*, Knowledge enhanced machine learning pipeline against diverse adversarial attacks (PMLR, 2021), pp. 3976–3987
8. Q. Le Roux, E. Bourbao, Y. Teglia, K. Kallas, A comprehensive survey on backdoor attacks and their defenses in face recognition systems. IEEE Access **12**, 47433–47468 (2024). https://doi.org/10.1109/ACCESS.2024.3382584
9. O.Y. Al-Jarrah, P.D. Yoo, S. Muhaidat, G.K. Karagiannidis, K. Taha, Efficient machine learning for big data: A review. Big Data Res. **2**(3), 87–93 (2015)
10. Y. Gao, B.G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, H. Kim, Backdoor attacks and countermeasures on deep learning: A comprehensive review (2020). arXiv preprint arXiv:2007.10760

11. W. Guo, B. Tondi, M. Barni, An overview of backdoor attacks against deep neural networks and possible defences (2021). arXiv preprint arXiv:2111.08429
12. Y. Li, Y. Jiang, Z. Li, ST. Xia, Backdoor learning: A survey. IEEE Transactions on Neural Networks and Learning Systems **35**(1), 5–22 (2024) https://doi.org/10.1109/TNNLS.2022.3182979
13. X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning (2017). arXiv preprint arXiv:1712.05526
14. A. Schwarzschild, M. Goldblum, A. Gupta, J.P. Dickerson, T. Goldstein, in *International Conference on Machine Learning*, Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks (PMLR, 2021), pp. 9389–9398
15. T. Zheng, H. Lan, B. Li, B. Be careful with pypi packages: You may unconsciously spread backdoor model weights. In: Conference on Machine Learning and Systems (2023). https://api.semanticscholar.org/CorpusID:270796221. https://www.semanticscholar.org/paper/Be-Careful-with-PyPI-Packages%3A-You-May-Spread-Model-Zheng-Lan/176c85ef19131059d4f4f517f6333d67a0ff8b8f
16. J. Dumford, W. Scheirer, in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, Backdooring convolutional neural networks via targeted weight perturbations (IEEE, 2020), pp. 1–9
17. H. Chen, C. Fu, J. Zhao, F. Koushanfar, Proflip: Targeted trojan attack with progressive bit flips. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7698–7707 (2021). https://doi.org/10.1109/ICCV48922.2021.00762. https://ieeexplore.ieee.org/document/9709910
18. K. Kurita, P. Michel, G. Neubig, Weight poisoning attacks on pre-trained models (2020). arXiv preprint arXiv:2004.06660
19. S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen, T. Chen, Backdoor attacks against transfer learning with pre-trained deep learning models. IEEE Transactions on Services Computing **15**(3), 1526–1539 (2022) https://doi.org/10.1109/TSC.2020.3000900. https://ieeexplore.ieee.org/document/9112322
20. M. Barni, K. Kallas, B. Tondi, in *2019 IEEE International Conference on Image Processing (ICIP)*, A new backdoor attack in cnns by training set corruption without label poisoning (IEEE, 2019), pp. 101–105
21. Y. Liu, Y. Xie, A. Srivastava, in *2017 IEEE International Conference on Computer Design (ICCD)*, Neural trojans (IEEE, 2017), pp. 45–48
22. T. Gu, K. Liu, B. Dolan-Gavitt, S. Garg, BadNets: Evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)
23. A. Turner, D. Tsipras, A. Madry, Label-consistent backdoor attacks (2019). arXiv preprint arXiv:1912.02771
24. B. Wu, S. Wei, M. Zhu, M. Zheng, Z. Zhu, M. Zhang, H. Chen, D. Yuan, L. Liu, Q. Liu, Defenses in Adversarial Machine Learning: A Survey (2023). https://arxiv.org/abs/2312.08890
25. L. Sun, Natural backdoor attack on text data (2020). arXiv preprint arXiv:2006.16176
26. X. Sheng, Z. Han, P. Li, X. Chang, A survey on backdoor attack and defense in natural language processing (2022). arXiv preprint arXiv:2211.11958
27. Y. Kong, J. Zhang, Adversarial audio: A new information hiding method and backdoor for dnn-based speech recognition models (2019). arXiv preprint arXiv:1904.03829
28. A. Bhalerao, K. Kallas, B. Tondi, M. Barni, in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, Luminance-based video backdoor attack against anti-spoofing rebroadcast detection (2019), pp. 1–6. https://doi.org/10.1109/MMSP.2019.8901711
29. Y. Li, Y. Li, B. Wu, L. Li, R. He, S. Lyu, Invisible backdoor attack with sample-specific triggers. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16443–16452 (2021). https://doi.org/10.1109/ICCV48922.2021.01615. https://ieeexplore.ieee.org/document/9711191
30. B. Wang, X. Cao, J. Jia, NZ. Gong, On Certifying Robustness against Backdoor Attacks via Randomized Smoothing (2020). https://arxiv.org/abs/2002.11750
31. J. Jia, Z. Yuan, D. Sahabandu, L. Niu, A. Rajabi, B. Ramasubramanian, B. Li, R. Poovendran, Fedgame: a game-theoretic defense against backdoor attacksin federated learning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23 (Curran Associates Inc., Red Hook, NY, USA 2024), p. 22. https://doi.org/10.5555/3666122.3668432. https://dl.acm.org/doi/abs/10.5555/3666122.3668432. Accessed 30 Sept 2024
32. W. Liu, S. Chawla, in *2009 IEEE International Conference on Data Mining Workshops*, A game theoretical model for adversarial learning (2009), pp. 25–30. https://doi.org/10.1109/ICDMW.2009.9
33. M. Kantarcıoğlu, B. Xi, C. Clifton, Classifier evaluation and attribute selection against active adversaries. Data Min. Knowl. Disc. **22**(1), 291–335 (2011)
34. R. Zhang, Q. Zhu, in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, A game-theoretic analysis of label flipping attacks on distributed support vector machines (2017), pp. 1–6. https://doi.org/10.1109/CISS.2017.7926118
35. J. Nash, Equilibrium points in n-person games. Proc. Natl. Acad. Sci. **36**(1), 48–49 (1950)
36. M. Jagielski, G. Severi, N. Pousette Harger, A. Oprea, in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, Subpopulation data poisoning attacks (2021), pp. 3104–3122
37. Q. Le Roux, K. Kallas, T. Furon, Restore: Exploring a black-box defense against dnn backdoors using rare event simulation. In: 2024 IEEE Conference on Secureand Trustworthy Machine Learning (SaTML), (IEEE Computer Society, Los Alamitos, 2024). p. 286–308. https://doi.org/10.1109/SaTML59370.2024.00021. https://doi.ieeecomputersociety.org/10.1109/SaTML59370.2024.00021. https://www.computer.org/csdl/proceedings-article/satml/2024/495000a286/1WPEDoaiNig. Accessed 30 Sept 2024
38. Y. Liu, W-C. Lee, G. Tao, S. Ma, Y. Aafer, X. Zhang, Abs: Scanning neural networks for back-doors by artificial brain stimulation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. CCS '19 (Association for Computing Machinery, New York, 2019), p. 1265–1282. https://doi.org/10.1145/3319535.3363216. https://dl.acm.org/doi/10.1145/3319535.3363216
39. W. Guo, L. Wang, Y. Xu, X. Xing, M. Du, D. Song, Tabor: Towards inspecting and eliminating trojan backdoors in deep neural networks. In: 2020 IEEE International Conference on Data Mining (ICDM), (2020), p. 162–171. https://doi.org/10.1109/ICDM50108.2020.00025. https://ieeexplore.ieee.org/document/9338311
40. J. Von Neumann, O. Morgenstern, *Theory of games and economic behavior* (Princeton University Press, 2007)
41. J.C. Burguillo, *Game Theory* (Springer International Publishing, Cham, 2018), pp. 101–135. https://doi.org/10.1007/978-3-319-69898-4_7
42. M.J. Osborne, A. Rubinstein, *A Course in Game Theory* (MIT Press, 1994)
43. J. Neumann, Zur theorie der gesellschaftsspiele. Mathematische Annalen **100**(1), 295–320 (1928) https://doi.org/10.1007/BF01448847. https://link.springer.com/article/10.1007/BF01448847. Accessed 30 Sept 2024
44. V. Chvatal, *Linear programming* (Macmillan, 1983)
45. Y. LeCun, C. Cortes, The mnist database of handwritten digits. (2005). https://api.semanticscholar.org/CorpusID:60282629. Accessed 30 Sept 2024

Publisher's Note