A Game-Theoretic Analysis of Adversarial Classification

Lemonia Dritsoula, Patrick Loiseau, and John Musacchio

***Abstract*—Attack detection is usually approached as a classification problem. However, standard classification tools often perform poorly because an adaptive attacker can shape his attacks in response to the algorithm. This has led to the recent interest in developing methods for *adversarial classification*, but to the best of our knowledge, there have been very few prior studies that take into account the attacker's tradeoff between adapting to the classifier being used against him with his desire to maintain the efficacy of his attack. Including this effect is key to derive solutions that perform well in practice.**

In this investigation we model the interaction as a game between a defender who chooses a classifier to distinguish between attacks and normal behavior based on a set of observed features and an attacker who chooses his attack features (class 1 data). Normal behavior (class 0 data) is random and exogenous. The attacker's objective balances the benefit from attacks and the cost of being detected while the defender's objective balances the benefit of a correct attack detection and the cost of false alarm. We provide an efficient algorithm to compute all Nash equilibria and a compact characterization of the possible forms of a Nash equilibrium that reveals intuitive messages on how to perform classification in the presence of an attacker. We also explore qualitatively and quantitatively the impact of the non-attacker and underlying parameters on the equilibrium strategies.

***Index Terms*—adversarial classification, game theory, security, Nash equilibrium, threshold strategies, randomization**

I. Introduction

Classification is one of the most used tools from machine learning. In its simplest instance, a classification algorithm trains a model from a set of labeled data samples of two different classes (class 0 and class 1) and then uses this model to predict the class of new data samples. Many classification algorithms (Support Vector Machines, Logistic Regression, Naive Bayes, etc.) were developed over the past decades and successfully used in applications ranging from computer vision to biology or marketing [4], [44], [50].

One of the most prominent applications of classification is security [53], where a defender typically aims to classify a system's usage into normal/non-attack (class 0) or malicious/attack (class 1). In this framework, *attacks* can range from spams received at a user's inbox (class 1) that must be distinguished from regular emails (class 0) to more serious attacks such as DoS or malicious infiltrations on a server (class 1) that must be distinguished from benign traffic (class 0). Although standard classification algorithms have been used to perform such tasks relatively successfully for some time, recent experimental studies showed that a smart adaptive attacker can easily shape his attacks to render those algorithms inefficient [37], [46], [51], [55]. This leads to the crucial question of how to perform *adversarial classification*, that is classification in a setting where (part of) the data is not i.i.d. but rather generated by an adversary trying to fool the classifier.

In the last decade, a significant body of work from the machine learning and security communities appeared on adversarial classification (see Section I-A). The literature defines two types of attacks: poisoning and evasion attacks. In poisoning attacks, the attacker can alter the training set. The adversarial classification literature mainly analyzes the vulnerability of standard algorithms in this case and sometimes proposes more robust algorithms. Most of these solutions, however, are either too simplistic in that they do not account for the attacker's adaptive nature, or too pessimistic because they rely on the worst-case (zero-sum) assumption that the attacker and defender have opposite objectives; leading to sub-optimal performance in both cases. In evasion attacks, the classifier is fixed and the attacker simply attempts to reverse-engineer it to find an attack that suits his goal while being classified in class 0. Interestingly, there, the literature recently proposed to use randomized classifiers as a defense; but without any formalism to justify the need for randomization and to optimize the distribution of classifiers used for defense. Overall, the problem of finding a classifier that works optimally against an adaptive attacker with a realistic (nonzero-sum) objective remains open.

In parallel, a set of works appeared on applying game theory to security scenarios in order to compute optimal defenses against an adaptive adversary, in particular in the contexts of intrusion detection [2] and defense resource allocation [49] (see Section I-A). In most games studied there, there exists a unique Nash equilibrium in mixed strategies, which justifies randomization and provides a framework to compute the defense distribution. However, none of these works studied a game with a payoff that captures the full complexity of the adversarial classification problem.

In this paper, we use a game-theoretic approach to tackle the question of how to perform classification in an adversarial setting where an adaptive attacker whose objective is not opposite to the defender's generates the data of class 1. Specifically, we propose to model the system as a game between an attacker and a defender, where the attacker chooses his attack patterns (corresponding to class 1 data) and the defender chooses a classification strategy. In our game, normal

Lemonia Dritsoula was with the University of California, Santa Cruz (UCSC), USA and is now at Google. Patrick Loiseau is with EURECOM, France and MPI-SWS, Germany. John Musacchio is with the Department of Technology Management at UCSC. E-mail: lenia@soe.ucsc.edu, patrick.loiseau@eurecom.fr, johnm@soe.ucsc.edu. This work was supported by AFOSR grant FA9550-09-1-0049 and NSF grants CNS-0910711 and CNS-0953884 and we acknowledge funding from the Alexander von Humboldt Foundation.

usage (class 0 data) is random and exogenous. The attacker's objective balances the benefit from attacks and the cost of being detected while the defender's objective balances the benefit of a correct attack detection and the cost of false alarm. Then, we give a complete analysis of the game's Nash equilibria that provides both an algorithm to compute all Nash equilibria and a characterization of the possible forms of a Nash equilibrium. Our characterization is very compact and reveals several important and intuitive messages on classification in the presence of an attacker.

i) First, we find that the defender must randomize the classification rule.
ii) Second, the defender should mix between classifiers only from a small subset of simple classifiers that correspond to applying a threshold on the attacker's benefit from the attack. If the attack pattern is a multi-dimensional quantity (i.e., the classification is based on several features), this is in contrast with known algorithms such as logistic regression which have a predefined shape of the boundary independently of the attacker's goal. We also find that the weight assigned to each threshold classifier is mainly proportional to the marginal reward increase at that point.
iii) Third, the attacker should essentially mimic the distribution of the normal behavior but only on a subset of the support with patterns that yield the highest payoff.
iv) Finally, our results allow us to analyze the investment tradeoffs that a strategic defender is facing when she needs to decide whether or not to acquire more data about the attacker (e.g., investing in a new sensor).

We provide numerical experiments that exemplify these results and their applications. Overall, our results provide a first step towards building classification algorithms that work well in general adversarial settings.

A. Related Work

a) Adversarial classification: The problem of adversarial classification has inspired a lot of research from the machine learning and security communities (see surveys in [7], [23]).

In a seminal paper, Dalvi et al. [15] tackle the problem of classifying a malicious intruder in the presence of an innocent user using a setting close to ours where the malicious intruder can perturb his behavior to confuse the classifier. However, they compute only the best response, which means that each player can adapt only once. Instead, we consider a fully adaptive attacker and defender and compute the Nash equilibrium. A number of papers (Globerson and Roweis [22], Zhou et al. [56], [57]) study a similar type of attacks and attempt to propose robust classifiers but always using a worst-case assumption which is overly pessimistic in practice.

To the best of our knowledge, the only works that analyze adversarial classification using nonzero-sum games are the studies by Brückner et al. [9]–[11], and the studies by Lisý et al. [28] and Samusevich [43]. The setting of Brückner et al. [9]–[11], however, is quite different from ours. They restrict the classifier to particular forms (e.g., logistic regression, where the defender's choice is the set of weights) and identify conditions under which a unique pure Nash equilibrium exists. In contrast, we do not restrict the classifiers a priori and we derive the set of classifiers used at equilibrium (which, as we show, depends on the attacker's utility and can therefore not be fixed without considering this); and we focus on mixed strategy Nash equilibria which are the only ones that make sense in most instances of our model. The setting of Lisý et al. [28] is closer to ours. They assume that the attacker and the defender both choose a threshold in $[0, 1]$ and that the attacker is detected if his chosen threshold is higher than the defender's. Using nonzero-sum payoffs, they give several properties of the Nash equilibrium (in mixed strategies) and propose a method based on discretization to compute an approximate equilibrium. Samusevich [43] derives extensions based on the same model, notably to take into account multiple types of attackers and bounded rationality. This model, however, abstracts away the complexity of multi-feature classification by assuming that players choose a threshold and that the classification result is computed through a ROC curve. In contrast, our model starts with all possible classifiers in the multi-feature case and shows that well-defined threshold classifiers are sufficient. Also, using a discrete model, we propose an efficient procedure to compute the exact Nash equilibrium.

All the aforementioned papers study poisoning attacks where the attacker can alter the training set. The study of evasion attacks was pioneered by Lowd and Meek [30] in the case of linear classifiers and extended by Nelson et al. [38] for convex-inducing classifiers. A recent study by Vorobeychik and Li [27], [54] recognizes the inefficiency of deterministic classifiers in this context. The authors compare the efficiency between two classifiers and investigate under which assumptions it is better to deterministically select a single classifier or uniformly randomize between them. In this paper, we formally justify the need for randomization using game theory and we consider a more general form of interaction (nonzero-sum game). We also start by considering an exhaustive set of all possible classifiers and analytically derive the subset of classifiers that the defender uses (threshold classifiers on the attacker's reward) and the distribution on this subset.

Some forms of thresholds are implemented ad hoc in most spam filtering algorithms to balance false alarm costs and detection gains, using standard deterministic classifiers. Our work provides a formal framework that takes into account the existence of strategic attackers willing to change how they attack to evade detection. Then, randomized classifiers are derived as an optimal and stable solution; no player has an economic benefit to unilaterally deviate. Our analysis shows that the defender should use threshold classifiers on the attack reward, contrasting with known algorithms like logistic regression which has a predefined shape of boundary independent of the attacker's goal.

In parallel to theoretical research aimed at proposing robust classifiers, several empirical works have studied the vulnerability of standard machine learning algorithms to various kinds of attacks [37], [46], [51], [55]. In particular, Sommer and Paxson [46] identified reasons why machine learning algorithms do not work well in practical adversarial settings and found that the most common pitfall is that attackers adjust

their activity in practice to avoid detection.

b) Game theory and intrusion detection: Researchers in the "game theory for security" community have also tackled the problem of detecting an attacker using game models, see surveys in [2], [14], [32]–[34], [41].

Chen and Leneutre [12] address the intrusion detection problem in heterogeneous networks consisting of nodes with different non-correlated security assets. Our model is similar in that different attack vectors can be thought of as distributions over targets of different values. The authors of [12], however, assume that the detection probability is the same for any attack and defense strategies, yielding the utility function to be the sum of the utilities on each target. On the contrary, we involve different detection and false alarm rates, which makes the set-up more realistic for adversarial classification, but makes the analysis more challenging. We also assume asymmetric information, since the defender is not aware of whether she faces an attacker or a non-attacker. Other works on game theory for intrusion detection (Alpcan and Başar [1], Liu et al. [29]) have similar restrictive assumptions as [12] that differentiate them from our work. Lye and Wing [33] have also investigated a security problem with multiple targets. They analyze a stochastic game between an attacker and administrator and compute the best response strategies of the players using a non-linear program, while we analyze the Nash equilibria of a classification game, which is a stronger solution concept.

The games mentioned above are nonzero-sum games. A number of other nonzero-sum games have been discussed in the literature under the term "security games", see [26]; but they share the restriction of [12] that the payoff is the sum of the payoffs on each targets, which does not model the problem of adversarial classification well. Let us finally mention that many applications of nonzero-sum games consider Stackelberg equilibria in which the defender chooses first its mixed action [26], [49]. In our model, we consider a simultaneous move game and find the Nash equilibria. We find that our nonzero-sum game is strategically equivalent to a zero-sum game, and thus we can analyze our nonzero-sum game using a zero-sum game.

Barni and Tondi [5], [6] use a game-theoretic approach to solve a problem of source identification. Their setting differs from ours in that the attacker chooses a distribution of attacks on top of which randomization is exogenously added, whereas we suppose that the attacker directly chooses his attack vector (using a mixed strategy). More importantly, they consider a zero-sum game and derive an asymptotic Nash equilibrium with numerical computations, while we focus on a nonzero-sum game and derive closed-form expressions for the equilibria. Stamm et al. [47] also use game theory to find optimal defense strategies in the context of multimedia forensics. They only derive best responses for the defender given a fixed attack distribution though, and do not characterize the Nash equilibria.

The classification game we investigate is similar in nature to the inspection game, a multi-stage game between a customs inspector and a smuggler, proposed and studied by Dresher [16] and Maschler [35]. Avenhaus et al. [3] find the equilibrium of the general nonzero-sum game by using an auxiliary zero-sum game in which the inspectee chooses a violation procedure and the inspector chooses a statistical test with a given false alarm probability. We do not separate the general nonzero-sum game into two games but show the equivalence to a zero-sum game and provide structure to the equilibrium strategies of a single-shot simultaneous-move game.

A subset of our results appeared in earlier versions of this paper [17], [18], in a simplified setting. Indeed, in [17], [18] we consider that the attack is characterized by a scalar quantity rather than a vector and impose the use of threshold strategies for the defender. Instead, here, we allow more flexibility by considering attack vectors of arbitrary dimension and considering the set of all possible classifiers and we prove that, at Nash equilibrium, the defender uses only threshold strategies on the attacker's reward (which is one of our main result). We also extend [17], [18] by considering all possible Nash equilibria and associated defender/attacker strategy form, by identifying when the Nash equilibrium is unique, and by showing numerically how our results can be applied to analyzing the variations of the equilibrium strategies with the model parameters and the trade-off in investing in a new sensor to increase the features used for classification.

The remainder of the paper is organized as follows. In Section II, the classification game we consider is presented. In Section III we justify the selection of threshold strategies for the defender and the proportionality of the attacker's equilibrium strategy to the non-attacker's distribution (Theorem 1). Section IV provides a Nash equilibrium analysis that gives insights on the structure of the players' equilibrium strategies (Theorem 2). Section V contains the simulation results, and the paper concludes with remarks in Section VI. To improve the flow of the paper, longer or more technical proofs are relegated to our technical report [19].

Notational Conventions: Throughout the paper, we will use the following notational conventions. All vectors are assumed to be column vectors and are denoted by bold lowercase letters (e.g., $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$). The inner product of two vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ of size n is denoted by $\boldsymbol{\alpha} \cdot \boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i \beta_i$. For matrices we use capital greek letters (e.g., Λ). We denote matrix elements in row i and column j by $\Lambda(i,j)$. We use the *prime* sign ($'$) for transpose of matrices and vectors. We use "$\min \boldsymbol{\alpha}$" to denote the minimum element of a vector $\boldsymbol{\alpha}$, and "minimize" when we minimize a specific expression over some constraints. The indicator function is denoted by $\mathbb{1}_{\text{cond}}$; it is equal to 1 if "cond" holds and is equal to 0 otherwise. The column vector of ones of length N is denoted by $\mathbf{1}_N$ and the matrix of ones of dimensions $N \times M$ is denoted by $1_{N \times M}$. The norm of a vector $\boldsymbol{x}$ of length N, denoted by $\|\boldsymbol{x}\|$, always refers to the L_1-norm, i.e, $\|\boldsymbol{x}\| = |x_1| + |x_2| + \ldots + |x_N|$, while the cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. The probability given to strategy s is denoted by α_s.

II. The Classification Game

In this section, we present our game-theoretic model of adversarial classification.

We consider a strategic situation between a *defender* and an agent that may either be an *attacker* with probability p or a *non-attacker* with probability $1-p$. Throughout the paper, we use the generic term non-attacker to designate a "normal user" of the system at stake, that is a non-strategic user without any particular adversarial objective (we will specify later the behavior of a non-attacker). The defender seeks to classify the agent as class 0 (non-attacker) or class 1 (attacker). The strategic attacker seeks to exploit the uncertainty of the defender (about the type of the agent) by attacking in such a way as to avoid being classified as an attacker. This scenario encompasses many applications. For instance, in spam classification, the spammer (the attacker) might change the frequency of words included in an email to evade spam filters. In that case, the non-attacker is the sender of a regular email. In another setting, the owner of fraudulent twitter accounts (the attacker) might try to acquire more followers or publish more posts, so that he will get misclassified as a normal user (the non-attacker here) [48], [51].

Formally, the agent selects an *attack vector* v in $\mathcal{V}$, where $\mathcal{V}$ is the set of all possible attack vectors. The attack vector contains all features used by the defender for classification; e.g., average number of followers, number of retweets, and others (in social networks fraud), number of initiated connections (in portscanner detection [24]), path on a graph among nodes in a network or number of accesses to different targets (in intrusion detection). Therefore, although we use the term "attack vector," our setting is not restricted to pure attack scenarios but instead covers any setting in which a defender needs to detect a malicious user, who seeks to evade detection by gaming. The defender selects a *classifier* in $\mathcal{C}$, where $\mathcal{C} \subseteq 2^{|\mathcal{V}|}$ is the set of all possible classifiers. A classifier corresponds to a classification rule that determines the class to which the agent is assigned upon observing his attack vector:

Definition 1 (Classifier)**.** *A classifier c is a function $c : \mathcal{V} \to \{0,1\}$, with*

$$c(v) = \begin{cases} 1 & \textit{for "attacker" classification,} \\ 0 & \textit{for "non-attacker" classification.} \end{cases}$$

We assume that $\mathcal{V}$ is finite, thus $\mathcal{C}$ is also a finite set.

If the agent is a non-attacker, he picks $v \in \mathcal{V}$ according to a distribution $P_{\mathrm{N}}(\cdot)$ over $\mathcal{V}$, known to both players. If the agent is an attacker, the choice v is strategic: the attacker seeks to maximize the payoff function

$$U^{\mathrm{A}}(v,c) = R(v) - c_{\mathrm{d}} \mathbb{1}_{c(v)=1}, \tag{1}$$

where $R : \mathcal{V} \to \mathbb{R}_+$ is the reward function, and c_{d} is the cost in case of detection. We refer to $R(v)$ as the "reward" (to the attacker) for the attack vector v, which is granted to the attacker even in case of detection. In contrast, his "payoff" is the reward minus the cost if detected. One can view the reward term as the immediate benefit from the attack, while the detection cost can be interpreted as the attacker's "opportunity cost" from losing the opportunity to extract value in future attacks after having been exposed. There are many real world scenarios in which the reward is granted to the attacker even upon detection. In most such scenarios, classification does not occur real-time leaving some exploitation window for the attacker. For instance, consider online app stores that contain malware. By the time these apps are classified as malicious and removed from the stores, the owners of the apps (attackers) will already have benefited, e.g., from online purchases, from compromising users' information, paid ads, and others.

The defender's payoff has two additive components. The first is the expected loss to the attacker. When the attacker is present, the loss to the defender is assumed to be minus the gain of the attacker, $-U^{\mathrm{A}}(v,c)$. Recall that the attacker's utility $U^{\mathrm{A}}(v,c)$ is composed of a reward term $R(v)$ minus a detection cost term. Thus the defender is in a sense earning a detection reward that matches the attacker's detection cost. This "reward" can be seen as the future costs the defender avoids by detecting the attacker now. Since the defender interacts with an attacker with chance p, the expected loss to the attacker is $-pU^{\mathrm{A}}(v,c)$. The second component captures the expected loss due to false alarms. Since the non-attacker is present with chance $1-p$, the expected false alarm cost is $1-p$ times the chance that a non-attacker would pick a v that gets classified as an attacker. Finally, the whole payoff function is scaled by the constant $1/p$ for the convenience of having the term $U^{\mathrm{A}}(v,c)$ appear unscaled in the payoff. Note that scaling a player's payoff function by a constant has no strategic effect on a game since the player retains the same preferences among outcomes. The resulting payoff function is

$$U^{\mathrm{D}}(v,c) = -U^{\mathrm{A}}(v,c) - \frac{1-p}{p} c_{\mathrm{fa}} \sum_{v' \in \mathcal{V}} P_N(v') \mathbb{1}_{c(v')=1}, \tag{2}$$

where c_{fa} is a constant that captures the cost of false alarms. Note that for simplicity of the exposition, we assume that the defender has a cost $R(v)$ (in the $U^{\mathrm{A}}(v,c)$ term) upon attack vector v, equal to the reward to the attacker. Our results, however, can be straightforwardly extended to the case where an attack vector v yields a cost to the defender $D(v) \neq R(v)$. Indeed, this would correspond to a payoff $\hat{U}^{\mathrm{D}}(v,c) = U^{\mathrm{D}}(v,c) + R(v) - D(v)$ which has no strategic effect since, for any $v \in \mathcal{V}$, maximizing $\hat{U}^{\mathrm{D}}$ with respect to c is equivalent to maximizing U^{D}. Note though, that the payoff of the defender will be changed.

To summarize our model, we define the following game.

Definition 2 (Classification game $\mathcal{G}$)**.** *The classification game $\mathcal{G} = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N, R(\cdot))$ is the two-player game between the attacker and the defender, where the strategy space of the attacker is $\mathcal{V}$, the strategy space of the defender is $\mathcal{C} \subseteq 2^{\mathcal{V}}$, and the payoffs are given by* (1) *and* (2)*, parameterized by*

- $p \in [0,1]$: *probability that the agent is an attacker;*
- $c_d \in \mathbb{R}_+$: *cost of detection;*
- $c_{fa} \in \mathbb{R}_+$: *cost of false alarm;*
- $P_N : \mathcal{V} \to [0,1]$: *probability measure that describes the non-attacker's distribution on $\mathcal{V}$;*
- $R : \mathcal{V} \to \mathbb{R}_+$: *the reward function.*

Throughout the paper, we analyze this game as a simultaneous, complete information game. In particular we assume that the payoffs' parameters are common knowledge. We are also able to formulate our game in normal (matrix) form since

the non-attacker is not strategic. At time 0, the attacker selects the attack vector and the defender selects the classifier. The uncertainty is coming from the presence or not of a strategic player. The expected utility of the defender incorporates this uncertainty through the probability p. The game would differ significantly if we supposed that there were two or more different types of strategic attackers. In such a setting we would not be able to reduce our model to a matrix game.

A. Model discussion and limitations

Before analyzing the game defined above, let us note that, as every model, it makes simplifying assumptions that limit its applicability and there are interesting aspects of security problems that it does not capture. Nevertheless, through its simplicity, we believe our model provides useful insights about the structure of Nash equilibria and the driving factors and intuition behind the players' equilibrium strategies that can be useful in some application scenarios. These insights would be difficult to obtain through a more complex, mathematically less tractable model. Moreover, the analysis of our simple model yields original and non-trivial mathematical results and derivations that can be useful to study more complex models. We leave this as future work but we clarify here the main limitations of our present model, the scenarios in which our assumptions may hold (or not) and some of the ways in which our results could be useful as a basis for further studies.

One of the main assumptions in our model is that the attacker is granted the reward $R(v)$ even in case of detection, and that the detection cost is the same for the attacker and for the defender. This assumption is technically important as we will see because it gives the "almost zero-sum" nature of the game that supports our analysis; but it limits the potential applicability of our model. For the reader to get a better idea of the implication of this assumption, we provide here three example scenarios, two where the assumption is reasonable and one where it is not.

Consider first the task of classifying an incoming email as spam or not. If the email is marked as spam, then the user never sees it so the attacker is granted the reward only upon no detection. Such a scenario cannot be captured by our model, but there are other examples in which the assumption make sense, such as the following two. Consider the problem of classifying fake news. There is an immediate reward to the publisher who spreads out the fake news (popularity, re-shares, advertising money). If the publisher gets classified as spreading out fake news, then the publisher loses credibility and/or ability to post more news. The detection cost can be seen as the lost opportunity cost from future posts of news (fake or real) which is also the detection bonus for the defender. Finally, consider the problem of classifying malware in apps. Most apps nowadays monetize ads shown to users but malicious apps can trick users to click on ads or gain important personal information of the user who has installed the app. When a malicious app is detected, some harm has already been done that cannot be reversed. The detection cost for the attacker would be the lost opportunity cost from being banned to show future ads. This cost can be seen as the detection bonus for the defender because in the place of this bad actor, another legitimate app can show ads with some expected lifetime reward. (Essentially, the impressions that would be allocated to this app can be diverted to be shown to other similar apps.)

Our model also assumes that parameter c_d is a constant independent of v, i.e., the defender gets a constant bonus independent of the attack for detecting the attack. This assumption is instrumental in obtaining the results that the defender only mixes amongst threshold strategies on the attacker's reward. Again, this assumption is reasonable in many cases but not in others. For the fake news or malware apps scenarios above, it is reasonable to assume that this cost is independent from the actual attack intensity, since it can be seen as lost opportunity cost from monetizing ads in the future. For cases such as audits from IRS though, there are usually fines imposed to tax fraudsters which makes the cost symmetric (whatever the fraudster pays, the IRS wins), but the fine usually depends on the amount misreported. Building on our results for the simpler case, however, we believe that it would be possible to study the game where c_d depends on v. One would need to find a ranking of attack vectors (depending on both functions R and c_d) such that an equivalent of our Lemma 4 holds.

Finally, our model is specified as a strategic-form game in which players choose their actions once, without knowing how their opponent has played. While this is a simplification of reality, the framework does not prevent us from considering players that take into account the future ramifications of the immediate outcome. For instance, attackers may perceive an opportunity cost of being less able to profit from future attacks by being detected now. As in models of economic competition and market entry, players can incorporate these opportunity costs into their payoff functions that describe their preferences over the outcomes of the "one-shot" game at hand [52]. That said, there are phenomena in repeated strategic interactions that are exposed only when the repeated interaction is modeled explicitly, such as in repeated games in which cooperative outcomes are enforced by the threat of future punishment [21]. If one were to develop and analyze an adversarial classification game that explicitly models the dynamics of an attacker and defender interacting on multiple occasions over an extended period, that model would need the analysis of the one-shot game to construct the needed value functions (see e.g., [20]). Thus our present results, in particular through the exponential reduction of the strategy space of the defender (Theorem 1) can be seen as a stepping-stone for constructing such a dynamic game model.

III. Justification of threshold strategies

In this section, we show that in equilibrium, the defender's strategy space can be reduced to threshold classifiers on the attacker reward. The sufficiency of threshold classifiers is a useful result as it allows us to compute the Nash equilibria of the game analytically and efficiently while simultaneously providing intuition about the players' equilibrium strategies. In particular, the number of classifiers can be as large as $2^{|\mathcal{V}|}$ while the number of threshold classifiers is of size $|\mathcal{V}|+1$, so

this is a great reduction in the defender's strategy space. We first introduce a number of useful definitions for the analysis of our game. Then in Section III-A, we show that the defender's strategy space can be reduced to the set of threshold classifiers and in Section III-B, we show that the attacker's strategy space can also be reduced.

We will be interested in mixed strategy equilibria, in which the attacker randomizes across multiple attack vectors with a probability distribution $\boldsymbol{\alpha}$ on $\mathcal{V}$ and the defender randomizes across multiple classifiers with a distribution $\boldsymbol{\beta}$ on $\mathcal{C}$. The expected attacker and defender payoffs are then given by

$$U^{\mathrm{A}}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}} \alpha_v U^{\mathrm{A}}(v,c)\beta_c, \tag{3}$$

$$U^{\mathrm{D}}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}} \alpha_v U^{\mathrm{D}}(v,c)\beta_c. \tag{4}$$

Note that a pure strategy is a special case of mixed strategies in which that particular pure strategy is selected with probability 1 and every other strategy with probability 0.[1]

Recall the definition of Nash equilibrium [21]:

Definition 3 (Nash equilibrium)**.** *The pair of probability measures $(\boldsymbol{\alpha},\boldsymbol{\beta})$ on $\mathcal{V}$ and $\mathcal{C}$ respectively is a Nash equilibrium (NE) of game $\mathcal{G}$ if each player's mixed strategy is a best response to the other player's mixed strategy, i.e.,*

$$U^A(\boldsymbol{\alpha},\boldsymbol{\beta}) \geq U^A(\hat{\boldsymbol{\alpha}},\boldsymbol{\beta}), \tag{5}$$

for every probability distribution $\hat{\boldsymbol{\alpha}}$ over $\mathcal{V}$, and

$$U^D(\boldsymbol{\alpha},\boldsymbol{\beta}) \geq U^D(\boldsymbol{\alpha},\hat{\boldsymbol{\beta}}) \tag{6}$$

for every probability distribution $\hat{\boldsymbol{\beta}}$ over $\mathcal{C}$.

We define the notion of best-response equivalent games in the same way as in [40]:

Definition 4. *Two games are* **best-response equivalent** *if the sets of best response strategies of a player in both games coincide for any strategy of the other player.*

Note that for best-response equivalent games, the strategy spaces of each player in both games need to be the same.

We now define the reduced strategy space for the defender that consists of threshold rules on all possible attack rewards.

Definition 5 (Set of threshold classifiers)**.**

$$\mathcal{C}^T = \{c\in\mathcal{C} : c(v) = \mathbb{1}_{R(v)\geq t}, \forall v\in\mathcal{V} \text{ for some } t\in\mathbb{R}\}.$$

When using a threshold classifier, the defender compares what the attack reward would have been from the observed attack vector to a threshold instead of computing a mapping from any possible attack vector to a detection probability. We assume that $\mathcal{C}^T \subseteq \mathcal{C}$, which holds for any reasonable $\mathcal{C}$, in particular for $\mathcal{C} = 2^{\mathcal{V}}$. The outcome of the analysis of the paper is that in most cases the defender should not use a single optimal value of t. Instead, the threshold should be chosen randomly from a particular range of values with a particular distribution. This range and distribution are found using the analysis of this paper.

We also define the probability of detection function as the probability of class 1 classification (or detection) given the attack vector v and the defender's strategy $\boldsymbol{\beta}$.

Definition 6 (Probability of detection function)**.** *The probability of detection for an attack vector v and defender's strategy $\boldsymbol{\beta}$ is defined as*

$$\pi_d^{\boldsymbol{\beta}}(v) = \sum_{c\in\mathcal{C}} \beta_c \mathbb{1}_{c(v)=1}, \quad \forall v\in\mathcal{V}. \tag{7}$$

A. Defender's reduced strategy space

Threshold strategies are simple and intuitive, but their optimality is not guaranteed a priori. Yet, in the classification game, we show that threshold strategies are optimal in a sense formalized in the following theorem.

Theorem 1. *For any NE $(\boldsymbol{\alpha},\boldsymbol{\beta})$ of $\mathcal{G} = (\mathcal{V},\mathcal{C},p,c_d,c_{fa},P_N,R(\cdot))$, there exists a NE of $\mathcal{G}^T = (\mathcal{V},\mathcal{C}^T,p,c_d,c_{fa},P_N,R(\cdot))$ with the same $\boldsymbol{\alpha}$ and equilibrium payoff pair and the same π_d in the support of the non-attacker's distribution.*

Theorem 1 shows in particular that, when restricted to using only threshold classifiers, the defender achieves the same equilibrium payoff. Hence, although there may exist Nash equilibria where the defender uses other classifiers, he does not lose anything by using only threshold classifiers.

At a high level, the intuition behind Theorem 1 is as follows. First, we can show that the utilities depend on the defender's strategy $\boldsymbol{\beta}$ only through the probability of detection function $\pi_{\mathrm{d}}^{\boldsymbol{\beta}}$. Then, we show that, at any NE, $\pi_{\mathrm{d}}^{\boldsymbol{\beta}}$ is non-decreasing in the attacker's reward (i.e., higher rewarding vectors have a higher probability of being detected). Finally, we show that any probability of detection function that is non-decreasing in the attacker's reward can be achieved by a mix of threshold classifiers. The proof of Theorem 1 is provided at the end of this section. We first establish a series of lemmas that give information about the game structure and will be used in the proof of Theorem 1.

Lemma 1. *For any strategy profile $(\boldsymbol{\alpha},\boldsymbol{\beta})$ of $\mathcal{G} = (\mathcal{V},\mathcal{C},p,c_d,c_{fa},P_N,R(\cdot))$, the expected payoffs of the players depend on $\boldsymbol{\beta}$ only through the probability of detection function $\pi_d^{\boldsymbol{\beta}}(\cdot)$:*

$$U^A(\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{v\in\mathcal{V}} \Big(\alpha_v R(v) - c_d\alpha_v \pi_d^{\boldsymbol{\beta}}(v)\Big), \tag{8}$$

$$U^D(\boldsymbol{\alpha},\boldsymbol{\beta}) = -U^A(\boldsymbol{\alpha},\boldsymbol{\beta}) - \frac{1-p}{p} c_{fa} \sum_{v'\in\mathcal{V}} \Big(P_N(v')\pi_d^{\boldsymbol{\beta}}(v')\Big). \tag{9}$$

By abuse of notation, we denote the probability of detection function by π_{d}, instead of $\pi_{\mathrm{d}}^{\boldsymbol{\beta}}$, when it brings no ambiguity.

Lemma 2. *For any function $f:\mathcal{V}\to[0,1]$, there exists a probability measure $\boldsymbol{\beta}$ over $\mathcal{C} = 2^{|\mathcal{V}|}$ s.t. $\pi_d^{\beta}(v) = f(v), \forall v\in\mathcal{V}$.*

[1] For most instances of interest no pure-strategy equilibrium exists. If one player chooses deterministically a pure strategy, the opponent would switch to a strategy to either evade detection completely or to guarantee detecting the attacker.

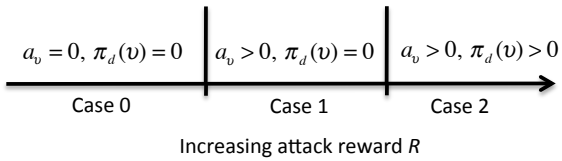

Fig. 1. Three cases for strategies selected in NE (see Lemma 3).

Proof. *Let f be an arbitrary function from $\mathcal{V}$ to $[0,1]$. Without loss of generality, we reindex strategies v such that f is non-decreasing, i.e., $\forall v_i, v_j \in \mathcal{V}$, with $i < j$, $f(v_i) \leq f(v_j)$. Starting with the attack vector v_1 with the lowest value of f, we assign probability $\beta_{c_1} = f(v_1)$ to the classifier $c_1 \in \mathcal{C}$ with $c_1(v) = 1, \forall v \in \mathcal{V}$. We then assign $\beta_{c_2} = f(v_2) - f(v_1)$ to the classifier c_2 with $c_2(v) = 1, \forall v \in \mathcal{V} \setminus \{v_1\}$. We continue this process until we reach the last vector $v_{|\mathcal{V}|}$. We assign probability $f(v_{|\mathcal{V}|}) - f(v_{|\mathcal{V}|-1})$ to the classifier that classifies only $v_{|\mathcal{V}|}$ as coming from an attacker. The remaining weight $1 - f(v_{|\mathcal{V}|})$ (if positive) is given to the classifier that never classifies the agent as an attacker. The strategy $\boldsymbol{\beta}$ derived with the above procedure is guaranteed by construction to have elements in $[0,1]$ with unit sum. Moreover $\pi_d^{\boldsymbol{\beta}}(v_1) = \beta_{c_1} = f(v_1)$ since v_1 is classified as coming from an attacker only by c_1, $\pi_d^{\boldsymbol{\beta}}(v_2) = \beta_{c_1} + \beta_{c_2} = f(v_2)$ since v_2 is classified as coming from an attacker only by c_1 and c_2, and so on until $\pi_d^{\boldsymbol{\beta}}(v_{|\mathcal{V}|}) = \sum_{i=1}^{|\mathcal{V}|} \beta_{c_i} = f(v_{|\mathcal{V}|})$ since $v_{|\mathcal{V}|}$ is classified as coming from an attacker by all classifiers c_1 through $c_{v_{|\mathcal{V}|}}$. Thus we have constructed a valid probability measure $\boldsymbol{\beta}$ over $\mathcal{C}$, with $\pi_d^{\boldsymbol{\beta}}(v) = f(v), \forall v \in \mathcal{V}$.* □

Without loss of generality, we now order the attack vectors in increasing attacker reward, i.e., $R(v_i) \leq R(v_{i+1}),\ \forall i \in \{1, \ldots, |\mathcal{V}| - 1\}$. The following lemma, illustrated in Fig. 1, establishes results on the players supports.

Lemma 3. *If $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE of $\mathcal{G} = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N, R(\cdot))$ that yields a probability of detection function π_d, then $\forall v \in \mathcal{V}$ such that $P_N(v) > 0$, one of the following three cases hold.*

- *Case 0: $\alpha_v = 0$ and $\pi_d(v) = 0$,*
- *Case 1: $\alpha_v > 0$ and $\pi_d(v) = 0$,*
- *Case 2: $\alpha_v > 0$ and $\pi_d(v) > 0$.*

Furthermore $R(v_0) \leq R(v_1) < R(v_2)$, for any strategies v_0, v_1, v_2 in Cases 0, 1, and 2 respectively.

Proof. *Suppose that $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE and that there exists $v^* \in \mathcal{V}$ with $P_N(v^*) > 0$ such that $\alpha_{v^*} = 0$ and $\pi_d^{\boldsymbol{\beta}}(v^*) > 0$. Let $\hat{\boldsymbol{\beta}}$ be a mixed strategy of the defender assigning zero detection probability on v^* and leaving the probability of detection unchanged for other attack vectors, i.e., such that $\pi_d^{\hat{\boldsymbol{\beta}}}(v^*) = 0$ and $\pi_d^{\hat{\boldsymbol{\beta}}}(v) = \pi_d^{\boldsymbol{\beta}}(v)$, for all $v \neq v^*$. By Lemma 2, strategy $\hat{\boldsymbol{\beta}}$ exists. By Lemma 1, we have*

$$U^D(\boldsymbol{\alpha}, \hat{\boldsymbol{\beta}}) = U^D(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \frac{1-p}{p} c_{fa} P_N(v^*) \pi_d^{\boldsymbol{\beta}}(v^*) > U^D(\boldsymbol{\alpha}, \boldsymbol{\beta}),$$

which contradicts the fact that $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE.

We now show that $R(v_0) \leq R(v_1) < R(v_2), \forall v_0, v_1, v_2$ in Cases 0, 1, and 2 respectively. Since both pure strategies v_1, v_2 are included in the attacker's equilibrium mixed strategy, $U^A(v_1, \boldsymbol{\beta}) = U^A(v_2, \boldsymbol{\beta})$. Since $\pi_d(v_1) = 0$ and $\pi_d(v_2) > 0$,

$$R(v_1) - c_d \cdot 0 = R(v_2) - c_d \pi_d(v_2) \Rightarrow R(v_1) < R(v_2).$$

Moreover, since $\alpha_{v_0} = 0, \alpha_{v_1} > 0$, $U^A(v_0, \boldsymbol{\beta}) \leq U^A(v_1, \boldsymbol{\beta})$. Since $\pi_d(v_0) = \pi_d(v_1) = 0$, this implies $R(v_0) - c_d \cdot 0 \leq R(v_1) - c_d \cdot 0$, hence $R(v_0) \leq R(v_1)$. □

Lemma 3 shows that, under some assumptions about the non-attacker's distribution, in NE: (1) the defender is never classifying as attacker upon seeing an attack vector that the attacker never uses, and (2) the attacker randomizes only amongst the most rewarding attack vectors and the defender randomizes only amongst classifiers that classify as attacker upon seeing the most rewarding attack vectors. This is illustrated in Section V, using numerical experiments.

We can also show the following corollary.

Corollary 1. *If $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE of $\mathcal{G} = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N, R(\cdot))$, for all v_i, v_j in Case 1, $R(v_i) = R(v_j)$.*

Proof. *Let v_1, v_2 be in Case 1. Then $\alpha_{v_1} > 0, \alpha_{v_2} > 0$ and*

$$\pi_d(v_1) = \pi_d(v_2) = 0. \tag{10}$$

Since the attacker mixes among both pure strategies v_1, v_2, these give the same expected utility to the attacker, we have:

$$\begin{aligned} &U^A(v_1, \boldsymbol{\beta}) = U^A(v_2, \boldsymbol{\beta}) \\ &\quad \Rightarrow R(v_1) - c_d \cdot 0 = R(v_2) - c_d \cdot 0, \quad \textit{(using (10))} \end{aligned}$$

hence $R(v_1) = R(v_2)$. □

What we show next is that, under certain assumptions on the non-attacker's behavior, vectors of higher attacker reward have higher or equal probability of getting detected.

Lemma 4. *Let $v_1, v_2 \in \mathcal{V}$ be such that $P_N(v_1), P_N(v_2) > 0$. In any NE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of $\mathcal{G} = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N, R(\cdot))$ that yields a probability of detection function π_d, we have $R(v_1) \leq R(v_2) \Rightarrow \pi_d(v_1) \leq \pi_d(v_2)$.*

Hence, if $P_N(v) > 0, \forall v \in \mathcal{V}$, $\pi_d(v)$ is non-decreasing in the attack reward $R(v)$.

Proof. *Let $v_1, v_2 \in \mathcal{V}$ be such that $P_N(v_1), P_N(v_2) > 0$ and $R(v_1) \leq R(v_2)$. By Lemma 3, v_1, v_2 are in Case 0, 1, or 2. If both v_1, v_2 are in either Case 0 or 1, then $\pi_d(v_1) = \pi_d(v_2) = 0$ so that $\pi_d(v_1) \leq \pi_d(v_2)$ holds. Similarly the result holds if v_1, v_2 are in Cases 0 or 1 and 2 respectively, since $\pi_d(v_1) = 0$ and $\pi_d(v_2) > 0$.*

The only remaining case is when both v_1 and v_2 are in Case 2. By Lemma 3, we have $\alpha_{v_1} > 0, \alpha_{v_2} > 0$. Since the attacker mixes among both v_1, v_2, then $U^A(v_1, \boldsymbol{\beta}) = U^A(v_2, \boldsymbol{\beta})$. But, from (8) we have

$$U^A(v_2, \boldsymbol{\beta}) = U^A(v_1, \boldsymbol{\beta}) + R(v_2) - R(v_1) + c_d\left(\pi_d(v_1) - \pi_d(v_2)\right),$$

so that $R(v_2) \geq R(v_1)$ implies $\pi_d(v_1) \leq \pi_d(v_2)$. □

Recall that Lemma 2 (and its proof) give a way to construct, for any probability of detection function targeted, a defender's strategy $\boldsymbol{\beta}$ that does yield this probability of detection function. The first step in the proof of Lemma 2 was to reindex attack vectors so that they have non-decreasing detection probability.

By Lemma 4, vectors ranked in non-decreasing reward already satisfy this property (if $P_{\mathrm{N}}(v) > 0, \forall v \in \mathcal{V}$). We can thus skip the step of reindexing and describe the nature of classifiers $c \in \mathcal{C}$ that are given positive weight. Classifier c_1 detects all attack vectors and is equivalent to a threshold classifier with threshold equal to the reward of the vector with the smallest reward (or smallest detection probability). Classifier c_2 detects all vectors except the one with the smallest reward and is equivalent to a threshold classifier with threshold equal to the second smallest attack reward, and so on until we reach classifier $c_{|\mathcal{V}|}$ that detects only the attack vector with highest reward (threshold equal to the highest reward). The remaining weight (if any) is given to classifier $c_{|\mathcal{V}|+1}$ that always classifies the agent as a non-attacker, which is a threshold classifier with threshold larger than the highest reward $R(v_{|\mathcal{V}|})$. The above procedure leads to the following corollary of Lemma 2:

Corollary 2. *For any NE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of $\mathcal{G} = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N, R(\cdot))$ that results in a non-decreasing probability of detection $\pi_d^{\boldsymbol{\beta}}$ there exists a NE $(\boldsymbol{\alpha}, \hat{\boldsymbol{\beta}})$ of $\mathcal{G}$, where $\hat{\boldsymbol{\beta}}_c = 0, \forall c \in \mathcal{C} - \mathcal{C}^T$ and $\pi_d^{\boldsymbol{\beta}}(v) = \pi_d^{\hat{\boldsymbol{\beta}}}(v),\ \forall v \in \mathcal{V}$.*

We now give the proof of Theorem 1.

Proof (Proof of Theorem 1)**.** *Let $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ be a NE of $\mathcal{G} = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N, R(\cdot))$, that results in a probability of detection function $\pi_d^{\boldsymbol{\beta}}$.*

Step 1. First assume that $P_N(v) > 0$, for all $v \in \mathcal{V}$. By Lemma 4, $\pi_d^{\boldsymbol{\beta}}$ is non-decreasing hence we can directly apply Corollary 2, which gives a probability $\hat{\boldsymbol{\beta}}$ over $\mathcal{C}$ with positive weight only on $\mathcal{C}^T$ such that $(\boldsymbol{\alpha}, \hat{\boldsymbol{\beta}})$ is a NE of $\mathcal{G}$. Therefore $(\boldsymbol{\alpha}, \boldsymbol{\beta}^T)$ is also NE of $\mathcal{G}^T = (\mathcal{V}, \mathcal{C}^T, p, c_d, c_{fa}, P_N, R(\cdot))$, where $\boldsymbol{\beta}^T$ is a probability measure over $\mathcal{C}^T$ with $\beta_c^T = \hat{\beta}_c$, for all $c \in \mathcal{C}^T$ and $\pi_d^{\boldsymbol{\beta}^T}(v) = \pi_d^{\hat{\boldsymbol{\beta}}}(v) = \pi_d^{\boldsymbol{\beta}}(v)$, for all $v \in \mathcal{V}$.

Step 2. Second, assume that there exists a unique $v^* \in \mathcal{V}$ with $P_N(v^*) = 0$. Note that, if $\alpha_{v^*} > 0$, then $\pi_d(v^*) = 1$. Indeed, suppose $\pi_d(v^*) < 1$. Consider $\hat{\boldsymbol{\beta}}$ that results in $\pi_d^{\hat{\boldsymbol{\beta}}}(v^*) = 1$ and $\pi_d^{\hat{\boldsymbol{\beta}}}(v) = \pi_d^{\boldsymbol{\beta}}(v)$, for all $v \neq v^*$. By Lemma 2, such $\hat{\boldsymbol{\beta}}$ exists. By Lemma 1,

$$U^D(\boldsymbol{\alpha}, \hat{\boldsymbol{\beta}}) = U^D(\boldsymbol{\alpha}, \boldsymbol{\beta}) + c_d \left(1 - \pi_d^{\boldsymbol{\beta}}(v^*)\right) > U^D(\boldsymbol{\alpha}, \boldsymbol{\beta}),$$

which contradicts the fact that $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE. We distinguish two sub-cases:

(a) v^* is not the highest reward vector, i.e., there exists $\hat{v} \in \mathcal{V}$, with $R(\hat{v}) > R(v^*)$. In that case, $\alpha_{v^*} = 0$. Indeed, suppose that $\alpha_{v^*} > 0$. By the previous analysis, $\pi_d(v^*) = 1$. By Lemma 1 we have

$$\begin{aligned} U^A(\hat{v}, \boldsymbol{\beta}) &= U^A(v^*, \boldsymbol{\beta}) + R(\hat{v}) - R(v^*) + c_d \cdot (1 - \pi_d(\hat{v})), \\ &> U^A(v^*, \boldsymbol{\beta}), \end{aligned}$$

since $R(\hat{v}) > R(v^*)$ and $\pi_d(\hat{v}) \leq 1$. Contradiction. Let $\tilde{\boldsymbol{\beta}}$ be such that $\pi_d^{\tilde{\boldsymbol{\beta}}}(v^*) = \pi_d^{\boldsymbol{\beta}}(v^*_{next})$ (where v^*_{next} is the next rewarding strategy after v^*) and $\pi_d^{\tilde{\boldsymbol{\beta}}}(v) = \pi_d^{\boldsymbol{\beta}}(v)$, for all $v \neq v^*$. By Lemma 2, such $\tilde{\boldsymbol{\beta}}$ exists. Since $\alpha_{v^*} = 0$, $(\boldsymbol{\alpha}, \tilde{\boldsymbol{\beta}})$ is still a NE of $\mathcal{G}$, with the same pair of payoffs as $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, but with probability of detection function $\pi_d^{\tilde{\boldsymbol{\beta}}}$ non-decreasing in the attack reward. We can then apply Corollary 2 and conclude in the same way as in Step 1; and $\pi_d^{\boldsymbol{\beta}^T}(v)$ and $\pi_d^{\boldsymbol{\beta}}(v)$ will differ only for $v = v^*$.

(b) v^* is the highest reward vector, i.e., for all $v \in \mathcal{V}, R(v^*) > R(v)$. In that case, either $\alpha_{v^*} > 0$ and then $\pi_d(v^*) = 1$ so that π_d is non-decreasing in the attack reward and we conclude as in Step 1; or $\alpha_{v^*} = 0$ and then we define $\tilde{\boldsymbol{\beta}}$ such that $\pi_d^{\tilde{\boldsymbol{\beta}}}(v^*) = 1$ and $\pi_d^{\tilde{\boldsymbol{\beta}}}(v) = \pi_d^{\boldsymbol{\beta}}(v)$ for all $v \neq v^*$ and conclude as in Step 2(a).

Step 3. Finally, the procedure above generalizes straightforwardly if there exist several attack vectors with $P_N(v) = 0$, hence concluding the proof. □

B. Reduced attacker's strategy space and equilibrium structure

We now turn to the attacker's equilibrium strategy. We first show Lemma 5, which gives the attacker's equilibrium strategy for most attack vectors. Then we show that this result, together with the reduction of the defender's strategy space, allow us to reduce the attacker's strategy space as well.

Lemma 5. *If $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE of $\mathcal{G} = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N, R(\cdot))$, then for all $v \in \mathcal{V}$ such that $0 < \pi_d(v) < 1$,*

$$\alpha_v = \frac{1-p}{p} \frac{c_{fa}}{c_d} P_N(v). \tag{11}$$

Proof. *Consider $v_i \in \mathcal{V}$ with $\pi_d(v_i) \in (0, 1)$. Since $\pi_d(v_i) \neq 0$, there exists $c_i \in \mathcal{C}$ s.t. $c_i(v_i) = 1$ with $\beta_{c_i} > 0$. Since $\pi_d(v_i) \neq 1$, there exists $c^* \in \mathcal{C}$ s.t. $c^*(v_i) = 0$ with $\beta_{c^*} > 0$. Now suppose that $c_i(v) = c^*(v), \forall v \in \mathcal{V} - \{v_i\}$. This is without loss of generality. Indeed, even if we cannot find in the support of $\boldsymbol{\beta}$ two such classifiers, we can construct another defender's strategy $\tilde{\boldsymbol{\beta}}$ that contains two such classifiers and has the same probability of detection function. We do that using a construction similar to the one in the proof of Lemma 2. If other vectors have the same reward as v_i, we include separately each classifier that detects those vectors as attacks. Since $\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\beta}}$ have the same probability of detection function, $(\boldsymbol{\alpha}, \tilde{\boldsymbol{\beta}})$ is also a NE.*

Finally, since $\beta_{c_i} > 0, \beta_{c^*} > 0$, $U^D(\boldsymbol{\alpha}, c_i) = U^D(\boldsymbol{\alpha}, c^*)$, that is

$$\begin{aligned} &\sum_{v \in \mathcal{V}} \alpha_v \big(R(v) - c_d \mathbb{1}_{c_i(v)=1}\big) + \frac{1-p}{p} c_{fa} \sum_{v' \in \mathcal{V}} P_N(v') \mathbb{1}_{c_i(v')=1} \\ &= \sum_{v \in \mathcal{V}} \alpha_v \big(R(v) - c_d \mathbb{1}_{c^*(v)=1}\big) + \frac{1-p}{p} c_{fa} \sum_{v' \in \mathcal{V}} P_N(v') \mathbb{1}_{c^*(v')=1}. \end{aligned}$$

This yields

$$-c_d \alpha_{v_i} + \frac{1-p}{p} c_{fa} P_N(v_i) = 0,$$

which immediately gives the result. □

Lemma 5 shows that for attack vectors v with some uncertainty of getting detected, the attacker mixes proportionally to the non-attacker's distribution. Note that the result clearly also applies to any Nash equilibrium of $\mathcal{G}^T$. Yet, the attacker's strategy space could be large and complex. For instance, consider a game in which there are M different targets and

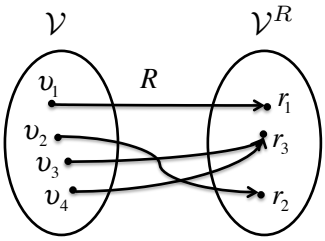

Fig. 2. Attacker's initial and reduced strategy spaces.

the attacker chooses at each time step from 1 to N a target to attack in $\{0, \ldots, M\}$ (counting the no-attack case). An attack v is a sequence of N attacks and $\mathcal{V}$ is the set of all such sequences which is of cardinality $(M+1)^N$. If the defender's classification rule is a threshold on the attack reward, however, permutations of attacks that yield the same attack reward will have the same probability of getting detected. Hence, as we explain below, they can be collapsed into a single strategy. For instance, if only the total number of times each target is hit (and not the order of the attacks) matters for the reward of the attacker, there are only $\binom{N+M+1}{N}$ combinations of attack rewards, and then at most that many unique values of the attacker's reward. Hence, we can exploit the fact that the defender uses only threshold classifiers in $\mathcal{G}^T$ and reduce the cardinality (and complexity) of the attacker's strategy space. As N or M increase, the benefits from this reduction become more profound.

The intuition behind the reduction in the attacker's strategy space comes from the following observation: If the attacker includes in his equilibrium support one attack vector of a certain reward, he should include all vectors of the same reward since all of them will have the same probability of detection. By Lemma 5, the attacker's equilibrium weight on each one is proportional to the non-attacker's distribution. Since the defender's classification is based on the reward of the attack vectors (and not on the actual vector), a game in which the attacker mixes on attack rewards (instead of attack vectors) does not influence the defender's equilibrium strategy. Furthermore, such a game does not give any more or less freedom to the attacker, but reduces the complexity (cardinality) of the strategy space of the attacker.

We now formalize the reduction of the game $\mathcal{G}^T$ to a game $\mathcal{G}^{R,T}$. The new attacker's strategy space is defined as the set of all images of the reward function $R : \mathcal{V} \to \mathbb{R}_+$ (see an illustration in Fig. 2):

Definition 7 (Reduced strategy space of the attacker)**.**

$$\mathcal{V}^R = \{r \in \mathbb{R}_+ : r = R(v), \textit{ for some } v \in \mathcal{V}\}. \tag{12}$$

Note that, although $\mathcal{V}^R$ is not rigorously a subset of $\mathcal{V}$, $\mathcal{V}^R$ is a reduced strategy space in the sense that R is clearly a surjection from $\mathcal{V}$ to $\mathcal{V}^R$. The non-attacker's probability measure can be similarly reduced to a probability measure that describes the non-attacker's distribution on $\mathcal{V}^R$ with

$$P_{\mathrm{N}}^R(r) = \sum_{v'} P_{\mathrm{N}}(v' \in \mathcal{V}) \mathbb{1}_{R(v')=r}. \tag{13}$$

Finally, since in $\mathcal{G}^T$, $\boldsymbol{\beta}$ is a probability on $\mathcal{C}^T$, we have that $c(v)$ is the same for all v with the same reward and that any two attack vectors with the same reward have the same probability of detection. Hence, by abuse of notation, we can define $c(r)$ as $c(v)$ for any v such that $R(v) = r$, and we can define the probability of detection function as a function of the reward by $\pi_{\mathrm{d}}(r) := \pi_{\mathrm{d}}(v)$, where $r = R(v)$. The reduced game $\mathcal{G}^{R,T} = (\mathcal{V}^R, \mathcal{C}^T, p, c_{\mathrm{d}}, c_{\mathrm{fa}}, P_{\mathrm{N}}^R)$ is then defined as the game between the attacker choosing $r \in \mathcal{V}^R$ and the defender choosing $c \in \mathcal{C}^T$ where the utilities adapt (1)-(2) in the obvious way:

$$U^{\mathrm{A}}(r, c) = r - c_{\mathrm{d}} \mathbb{1}_{c(r)=1},$$
$$U^{\mathrm{D}}(r, c) = -U^{\mathrm{A}}(r, c) - \frac{1-p}{p} c_{\mathrm{fa}} \sum_{r' \in \mathcal{V}^R} P_N^R(r') \mathbb{1}_{c(r')=1}.$$

The expected utilities in mixed strategies also adapt (8)-(9) in the obvious way.

The following proposition formalizes the relationship between the NE of $\mathcal{G}^T$ and $\mathcal{G}^{R,T}$.

Proposition 1. *If $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE of $\mathcal{G}^T = (\mathcal{V}, \mathcal{C}^T, p, c_d, c_{fa}, P_N, R(\cdot))$, then $(\boldsymbol{\alpha}^*, \boldsymbol{\beta})$ is a NE of $\mathcal{G}^{R,T} = (\mathcal{V}^R, \mathcal{C}^T, p, c_d, c_{fa}, P_N^R)$ with the same equilibrium payoff pair where $\alpha^*_{r_i} = \sum_{v_j \in \mathcal{V}, R(v_j)=r_i} \alpha_{v_j}$, $\forall r_i \in \mathcal{V}^R$.*

The proof of Proposition 1 can be found in our technical report [19]. The importance of Proposition 1 comes from the fact that it is easier to compute the NE $(\boldsymbol{\alpha}^*, \boldsymbol{\beta})$ of $\mathcal{G}^{R,T}$, in which the cost matrix of the attacker consists of non-identical rows. This NE is given in Theorem 2 in Section IV. However, from the NE $(\boldsymbol{\alpha}^*, \boldsymbol{\beta})$ of $\mathcal{G}^{R,T}$, we can easily recover a NE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of $\mathcal{G}^T$ as follows. By Proposition 1, $\boldsymbol{\beta}$ is unchanged. Given $\boldsymbol{\alpha}^*$ on $\mathcal{V}^R$ we compute the attacker's strategy $\boldsymbol{\alpha}$ over $\mathcal{V}$, as follows: For $r_i \in \mathcal{V}^R$ with $\pi_{\mathrm{d}}(r_i) \in (0, 1)$, α_{v_i} is given by (11) $\forall v_i \in \mathcal{V}$ with $R(v_i) = r_i$, by Lemma 5. For r_i with $\pi_{\mathrm{d}}(r_i) \in \{0, 1\}$, any possible combination of weights is possible, as long as $\sum_{v_j \in \mathcal{V}, R(v_j)=r_i} \alpha_{v_j} = \alpha^*_{r_i}$. Hence, although we reduce the attacker's strategy space, we provide a roadmap to get all NE of $\mathcal{G}^T$ as well. Note that if all attack strategies in $\mathcal{V}$ yield distinct attack reward, then $|\mathcal{V}| = |\mathcal{V}^R|$ and there is no reduction in the attacker's strategy space.

IV. Nash Equilibrium Analysis

Our goal in this section is to characterize the structure of the NE of the classification game. It is known that every finite game (finite number of players with finite number of strategies for each player) has a mixed-strategy NE [36]. Our game is finite, thus it admits a NE in mixed strategies. However, finding the Nash equilibrium has a high computational complexity in the general case [13].

We consider the game $\mathcal{G}^{R,T} = (\mathcal{V}^R, \mathcal{C}^T, p, c_{\mathrm{d}}, c_{\mathrm{fa}}, P_{\mathrm{N}}^R)$, in which the attacker's strategy space consists of distinct attack rewards $r \in \mathcal{V}^R$, the defender's strategy space consists of threshold classifiers $c \in \mathcal{C}^T$, and P_{N}^R is the non-attacker's probability measure on $\mathcal{V}^R$ given by (13). We denote by r_i, $i \in \{1, \ldots, |\mathcal{V}^R|\}$ the elements of $\mathcal{V}^R$ and, without loss of generality, we assume that they are ranked in increasing order,

i.e., $r_i < r_{i+1}$ for all $i \in \{1,\dots,|\mathcal{V}^R|-1\}$. Similarly, classifier c_i corresponds to a threshold classifier with threshold equal to the attacker reward r_i. Recall (see Definition 5) that a threshold classifier with threshold t classifies as attacker if the reward is $r \geq t$. Hence, c_1 corresponds to the "always classify as attacker" strategy. By definition, $\mathcal{C}^T$ also includes the "always classify as non-attacker" strategy, which is denoted by $c_{|\mathcal{V}^R|+1}$ and corresponds to a threshold $r_{|\mathcal{V}^R|}+\delta$ for any $\delta > 0$. Hence we have $|\mathcal{C}^T| = |\mathcal{V}^R| + 1$.

We can express the payoff functions of the players in compact form as matrices. We define $\tilde{\Lambda}$ to be the cost matrix of the attacker, with $\tilde{\Lambda}(i,j) = c_d \mathbb{1}_{r_i \geq r_j} - r_i$, $i \in \{1,\dots,|\mathcal{V}^R|\}$, $j \in \{1,\dots,|\mathcal{V}^R+1|\}$:

$$\tilde{\Lambda} = c_{\mathrm{d}} \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & 1 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \ddots & \vdots & \vdots \\ \vdots & & & \ddots & 0 & \vdots \\ 1 & \cdots & \cdots & \dots & 1 & 0 \end{pmatrix} - \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{|\mathcal{V}^R|-1} \\ r_{|\mathcal{V}^R|} \end{pmatrix} \cdot \mathbf{1}'_{|\mathcal{V}^R|+1}.$$

Recall that we use the *prime* sign ($'$) for transpose of matrices and vectors. There are $|\mathcal{V}^R|$ rows (strategies) for the attacker, ranked by increasing reward. Certain computations are simplified by using a matrix with only positive entries. Therefore, we define $\Lambda \triangleq \tilde{\Lambda} + (r_{|\mathcal{V}^R|} + \epsilon) \cdot 1_{|\mathcal{V}^R| \times |\mathcal{V}^R|+1}$, where $\epsilon > 0$. Adding a constant to the players' payoff does not affect their best responses, and hence does not change the equilibrium strategies. Thus, from here on, we will use matrix Λ to define the players' payoffs.

In the following, the pair $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ will denote the probability measures of the attacker and defender on $\mathcal{V}^R, \mathcal{C}^T$ respectively. The attacker's expected cost is given by $\boldsymbol{\alpha}'\Lambda\boldsymbol{\beta}$. The defender's expected payoff is given by $\boldsymbol{\alpha}'\Lambda^{\mathrm{eq}}\boldsymbol{\beta}$, with

$$\Lambda^{\mathrm{eq}} = \Lambda - \mathbf{1}_{|\mathcal{V}^R|} \cdot \boldsymbol{\mu}', \tag{14}$$

where $\boldsymbol{\mu}$ represents the false alarm penalty vector for the defender with elements $\mu_i = \dfrac{1-p}{p} c_{\mathrm{fa}} \sum_{k \geq i} P_{\mathrm{N}}^R(r_k)$. We assume that $\boldsymbol{\mu}$ is a strictly decreasing vector (component-wise): $\mu_i > \mu_{i+1}, \forall i \in \{1,\dots,|\mathcal{V}^R|\}$. This assumption is equivalent to assuming that the non-attacker gives some positive weight to all strategies $r \in \mathcal{V}^R$, i.e., $P_{\mathrm{N}}^R(r_i) > 0, \forall i \in \{1,\dots,|\mathcal{V}^R|\}$. Even if this property does not hold, we can still describe how both players behave, as shown in Theorem 1 in Section III.

It is easy to show that $\mathcal{G}^{R,T} = (\mathcal{V}^R, \mathcal{C}^T, p, c_{\mathrm{d}}, c_{\mathrm{fa}}, P_{\mathrm{N}}^R)$ is best-response equivalent (see Definition 4) to a zero-sum game, in which the attacker's cost is given by $\boldsymbol{\alpha}'\Lambda^{\mathrm{eq}}\boldsymbol{\beta}$. Indeed, the two games have the same players with the same strategy spaces. Vector $\boldsymbol{\mu}$ depends only on the non-attacker's distribution and is fixed. Adding constants to the columns of the cost matrix of the attacker (row player) in the original classification game yields the cost matrix of the attacker of the new game without changing the Nash equilibria of the game. Indeed, the defender's payoff matrix is unchanged, and, for any given $\boldsymbol{\beta}$, minimizing $\boldsymbol{\alpha}'\Lambda\boldsymbol{\beta}$ and minimizing $\boldsymbol{\alpha}'\Lambda^{\mathrm{eq}}\boldsymbol{\beta}$ with $\Lambda^{\mathrm{eq}}(i,j) = \Lambda(i,j) - \mu_j$ give the same minimizing strategy for the attacker. Thus the two games generate the same sets of best response functions and have the same equilibrium strategies.

Note that the best-response equivalence of our game to a zero-sum game guarantees that in all NE the defender's expected payoff will be the same (and equal to the value of the zero-sum game), but the attacker's payoff is not always the same in each equilibrium of the original nonzero-sum game. Indeed, equilibria of our original game could give different payoffs to the attacker after transforming his cost matrix Λ^{eq} (adding constants to the columns) back to Λ.

There exist polynomial algorithms to compute the Nash equilibria in zero-sum games via a transformation to a linear program (LP) [25]. These algorithms, however, do not provide structure on the equilibrium strategies. In the remaining of this section, we aim to give more intuition on the players' NE strategies than the solution derived via a linear programming toolbox. Along the way, we also provide an algorithm to compute the NE, which can be faster than solving the LP.

Our results can be summarized in the following theorem.

Theorem 2. *Algorithm 1 finds all NE of the classification game $\mathcal{G}^{R,T}$. Moreover, if $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE, then, there exists $k \in \{1,\dots,|\mathcal{V}^R|\}$ such that*

$$\boldsymbol{\beta} = (0,\dots,0,\beta_k,\dots,\beta_{|\mathcal{V}^R|},\beta_{|\mathcal{V}^R|+1}),$$
$$\boldsymbol{\alpha} = (0,\dots,0,\alpha_k,\dots,\alpha_{|\mathcal{V}^R|}),$$

where

$$\beta_i = \frac{r_i - r_{i-1}}{c_d}, \qquad \forall i \in \{k+1,\dots,|\mathcal{V}^R|\}, \tag{15}$$

$$\alpha_i = \frac{1-p}{p}\frac{c_{fa}}{c_d} P_N^R(r_i), \quad \forall i \in \{k+1,\dots,|\mathcal{V}^R|-1\}, \tag{16}$$

and $\beta_k, \beta_{|\mathcal{V}^R|+1} \geq 0$ and $\alpha_k, \alpha_{|\mathcal{V}^R|} \geq 0$ are such that

(i) $\beta_k \in (0, \frac{r_k - r_{k-1}}{c_d}), \beta_{|\mathcal{V}^R|+1} = 0$, *and* α_k *satisfies* (16), $\alpha_{|\mathcal{V}^R|} > 0$*; or*

(ii) $\beta_k = 0, \beta_{|\mathcal{V}^R|+1} > 0$, *and* $\alpha_k \in (0, \frac{1-p}{p}\frac{c_{fa}}{c_d} P_N^R(r_k))$, $\alpha_{|\mathcal{V}^R|}$ *satisfies* (16)*; or*

(iii) $\beta_k = 0, \beta_{|\mathcal{V}^R|+1} = 0$, *and* $\alpha_k \in [0, \min(\frac{1-p}{p}\frac{c_{fa}}{c_d} P_N^R(r_k), 1 - \sum_{i=k+1}^{|\mathcal{V}^R|-1} \alpha_i)]$, $\alpha_{|\mathcal{V}^R|} \geq 0$*; or*

(iv) $\beta_k \in [0, \frac{r_k - r_{k-1}}{c_d}], \beta_{|\mathcal{V}^R|+1} \geq 0$, *and* $\alpha_k, \alpha_{|\mathcal{V}^R|}$ *satisfy* (16).

The rest of this section is dedicated to proving Theorem 2. As a direct consequence of best-response equivalence of our game to a zero-sum game, we have the following result.

Lemma 6. *In NE, the defender's strategy $\boldsymbol{\beta}$ solves the following linear program (LP):*

$$\begin{aligned} \underset{\boldsymbol{\beta}, z}{\textit{maximize}} \quad & -\boldsymbol{\mu}'\boldsymbol{\beta} + z \\ \textit{subject to} \quad & z\mathbf{1}_{|\mathcal{V}^R|} \leq \Lambda\boldsymbol{\beta} \\ & \mathbf{1}'_{|\mathcal{V}^R|+1} \cdot \boldsymbol{\beta} = 1, \ \boldsymbol{\beta} \geq \mathbf{0}. \end{aligned} \tag{17}$$

Proof. *If $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE of $\mathcal{G}^{R,T}$ with attacker's cost matrix Λ, then $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE of the zero-sum, best-response equivalent game with cost matrix Λ^{eq}. Therefore, $\boldsymbol{\beta}$ maximizes $\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}'\Lambda^{eq}\boldsymbol{\beta} = \min[\Lambda\boldsymbol{\beta}] - \boldsymbol{\mu}'\boldsymbol{\beta}$. Transforming this optimization problem to an LP we get the program* (17). □

An important consequence of Lemma 6, is that the defender's strategy $\boldsymbol{\beta}$ in Nash equilibrium maximizes her minimum payoff. Thus playing β gives the defender the robustness property that her expected payoff will not be any worse than her expected Nash equilibrium payoff regardless of what the attacker chooses to play.

We now define the main entities used throughout the section.

Definition 8. *A* **polyhedron** *is the solution set of a finite number of linear inequality constraints. An inequality constraint is* **tight** *if it holds as an equality; otherwise, it is* **loose**. *A point $\boldsymbol{x} = (x_1, \ldots, x_{|\mathcal{V}^R|+1})$ of a polyhedron is said to be* **extreme** *if there is no $\boldsymbol{x}'$ whose set of tight constraints is a strict superset of the set of tight constraints of $\boldsymbol{x}$. For an n-dimensional linear program, a point is called a* **basic solution**, *if n linearly independent constraints are tight for that point. A* **feasible solution** *to a linear program is a solution that satisfies all constraints. A point is a* **basic feasible solution**, *iff it is a basic solution that is also feasible. Two distinct basic feasible solutions to an n-dimensional linear program are* **adjacent** *if we can find $n-1$ linear independent constraints that are tight at both of them. We say that a point $\boldsymbol{x}$ of a polyhedron* **corresponds** *to strategy $\boldsymbol{\beta}$ (or strategy $\boldsymbol{\beta}$ corresponds to $\boldsymbol{x}$), if $\boldsymbol{\beta} = \boldsymbol{x}/\|\boldsymbol{x}\|$.*

Extreme point and basic feasible solution are equivalent terms [31, Chapter 2.5] and we will use them interchangeably.

A. Form of optimal extreme points

In this section, we gain intuition on the structure of the defender's NE strategy $\boldsymbol{\beta}$. We first show the following lemma.

Lemma 7. *Any NE strategy $\boldsymbol{\beta}$ of the defender corresponds to an extreme point or a convex combination of extreme points of the polyhedron defined by*

$$P: \ \Lambda \boldsymbol{x} \geq \mathbf{1}_{|\mathcal{V}^R|}, \boldsymbol{x}_{|\mathcal{V}^R|+1} \geq \mathbf{0}. \tag{18}$$

The proof of Lemma 7 can be found in our technical report [19]. We call the first type of constraints "inequality constraints" and the second type "positivity constraints." There are $|\mathcal{V}^R|$ inequality constraints and $|\mathcal{V}^R|+1$ positivity constraints. Writing down the inequality constraints, we get

$$\begin{aligned}
c_{\mathrm{d}} \cdot x_1 + \left(r_{|\mathcal{V}^R|} - r_1 + \epsilon\right) \cdot \|\boldsymbol{x}\| &\geq 1 \\
c_{\mathrm{d}} \cdot (x_1 + x_2) + \left(r_{|\mathcal{V}^R|} - r_2 + \epsilon\right) \cdot \|\boldsymbol{x}\| &\geq 1 \\
\vdots & \\
c_{\mathrm{d}} \cdot (x_1 + x_2 + \ldots + x_{|\mathcal{V}^R|}) + \qquad \epsilon \qquad \cdot \|\boldsymbol{x}\| &\geq 1.
\end{aligned}$$

Searching for an extreme point of the polyhedron P defined in (18) is computationally straightforward and there are known algorithms that provide polynomial complexity. Our goal is to provide an algorithm that can run faster (though still polynomially) and in parallel to provide intuition on the structure of the extreme points. The main method used is to reduce the search space by eliminating suboptimal non-extreme points.

Combining properties of basic feasible solutions of an LP and of the structure of the defender's LP, we show the following lemma describing the set of tight inequality constraints.

Lemma 8. *At an extreme point $\boldsymbol{x}$ that corresponds to a defender's NE strategy $\boldsymbol{\beta}$, there exists exactly one contiguous block (of indices) of inequality constraints that are tight and the last inequality constraint is tight.*

The proof of Lemma 8 can be found in our technical report [19]. We define s as the index of the first (starting) tight inequality constraints. The lemma states that all inequality constraints from s to $|\mathcal{V}^R|$ are tight. We can now state the result that describes the form of optimal extreme points of the defender's LP. Its proof is in our technical report [19].

Lemma 9. *Any extreme point $\boldsymbol{x}$ of polyhedron P that corresponds to a defender's NE strategies is of one of the following types:*

$$\textit{Type I: } \boldsymbol{x} = \big(0, \ldots, 0, x_{s_1} \geq 0, x_{s_1+1} > 0, \ldots, x_{|\mathcal{V}^R|} > 0, 0\big)',$$
$$\textit{Type II: } \boldsymbol{x} = \big(0, \ldots, 0, x_{s_2+1} > 0, \ldots, x_{|\mathcal{V}^R|} > 0, x_{|\mathcal{V}^R|+1} \geq 0\big)',$$

for some $s_1, s_2 \in \{1, \ldots, |\mathcal{V}^R|\}$, with $x_i = \dfrac{r_i - r_{i-1}}{c_d}\|\boldsymbol{x}\|$ for all $i \in \{s_j + 1, \ldots, |\mathcal{V}^R|\}$ ($j \in \{1, 2\}$). Moreover, there exist at most one extreme point of type I and two adjacent extreme points of type II that correspond to a defender's NE strategies.

B. Form of players' equilibrium strategy

With Lemma 9 that describes the form of the possible extreme points of polyhedron P defining the constraints of the defender's LP, we can now prove our main result, Theorem 2. The complete proof can be found in our technical report [19]. It essentially works by enumerating all of the possible combinations of optimal basic feasible solutions allowed by Lemma 9 to get the possible forms of $\boldsymbol{\beta}$ (i.e., all possible solutions of the defender's LP which we consider the primal) and using the complementary slackness condition from Linear Programming to get the possible forms of $\boldsymbol{\alpha}$ (i.e., all possible solutions of the dual LP).

Theorem 2 provides both an algorithm (Algorithm 1) that finds all NE and a compact characterization of the restricted number of possible forms that a NE can have. Interestingly, we observe that the defender assigns a weight to a reward r_i that is positive and proportional to the marginal reward increase at that point, on a support that goes until the highest reward $r_{|\mathcal{V}^R|}$. This is somewhat counter-intuitive as it implies that the defender includes at NE with positive weights classifiers that almost never classify as attacker even for a high reward and even if the probability that a non-attacker uses this reward is arbitrarily small. We also recover in Theorem 2 the fact that the attackers mimics the non-attacker's distribution (proportionally) on a support that corresponds to the defender's support.

For readers familiar with Linear Programming, let us finally remark that the result in Theorem 2 (and its proof) is in accordance with the relationship between degeneracy and multiplicity of the primal and the dual optimal solutions [45, p. 144] (a degenerate optimal solution is a solution of size n where more than n constraints are tight). If there exists a unique non-degenerate optimal solution of the primal (defender's LP)

then the optimal solution to the dual problem (the attacker's LP) is also unique and non-degenerate (cases (i) and (ii) of Theorem 2). If the primal has multiple solutions with at least one non-degenerate then the optimal solution to the dual is unique and degenerate (case (iv) of Theorem 2). If the primal optimal solution is unique and degenerate, then the dual has multiple optimal solutions (case (iv) of Theorem 2). Another benefit of having the solution of the game corresponding to the solution of an LP is that there are well studied methods for analyzing the sensitivity of an LP to parameters (see for instance [8]). These methods can be used to study how sensitive the game solution is to parameter perturbations.

Algorithm 1: How to compute the NE $(\boldsymbol{\alpha}, \boldsymbol{\beta})$

for $type = 1, 2$ **do**
  construct $\boldsymbol{\beta}$ for $s \in \{1, \ldots, |\mathcal{V}^R|\}$ using Algorithm 2
  find $(\boldsymbol{\beta}_{1,2}, s^*_{1,2})$ that maximize defender's payoff $U^D_{1,2}$
if $U^D_1 > U^D_2$ **then**
  $\boldsymbol{\beta} \leftarrow$ compute-$\beta(s^*_1, 1)$; $\boldsymbol{\alpha} \leftarrow$ compute-$\alpha(s^*_1)$
if $U^D_1 < U^D_2$ **then**
  if s^*_2 *is unique* **then**
    $\boldsymbol{\beta} \leftarrow$ compute-$\beta(s^*_2, 2)$; $\boldsymbol{\alpha} \leftarrow$ compute-$\alpha(s^*_2)$
  else
    // denote s^*_{2a} and $s^*_{2b}=s^*_{2a}+1$ the 2 solutions
    $\boldsymbol{\beta}_a \leftarrow$ compute-$\beta(s^*_{2a}, 2)$;
    $\boldsymbol{\beta}_b \leftarrow$ compute-$\beta(s^*_{2b}, 2)$
    $\boldsymbol{\beta} \leftarrow$ convex hull of $\boldsymbol{\beta}_a, \boldsymbol{\beta}_b$
    $\boldsymbol{\alpha} \leftarrow$ compute-$\alpha(s^*_{2b})$
if $U^D_1 = U^D_2$ **then**
  if s^*_2 *is unique* **then**
    // $s^*_2 = s^*_1$
    $\boldsymbol{\beta}_1 \leftarrow$ compute-$\beta(s^*_1, 1)$;
    $\boldsymbol{\beta}_2 \leftarrow$ compute-$\beta(s^*_1, 2)$
    if $\boldsymbol{\beta}_1 \neq \boldsymbol{\beta}_2$ **then**
      $\boldsymbol{\beta} \leftarrow$ convex hull of $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$
    else
      $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}_1$
    $\boldsymbol{\alpha} \leftarrow$ compute-$\alpha(s^*_1)$
  else
    // denote s^*_{2a} and $s^*_{2b}=s^*_{2a}+1$ the 2 solutions
    // the type 1 and type 2a solutions are identical
    $\boldsymbol{\beta}_a \leftarrow$ compute-$\beta(s^*_{2a}, 1)$;
    $\boldsymbol{\beta}_b \leftarrow$ compute-$\beta(s^*_{2b}, 2)$
    $\boldsymbol{\beta} \leftarrow$ convex hull of $\boldsymbol{\beta}_a, \boldsymbol{\beta}_b$
    $\boldsymbol{\alpha} \leftarrow$ compute-$\alpha(s^*_{2b})$

V. Numerical Results

In this section, we numerically study several instances of our model and observe the behavior of the players in NE. In particular, we explore the strategic attacker's exploitation of the knowledge he has of the non-attacker's distribution (noise).

Algorithm 2: Compute-$\beta(s, type)$

for $i = 1$ **to** $s - 1$ **do**
  $\beta_i \leftarrow 0$
for $i = s + 1$ **to** $|\mathcal{V}^R|$ **do**
  $\beta_i \leftarrow \dfrac{r_i - r_{i-1}}{c_{\mathrm{d}}}$
$remainder \leftarrow \mathbf{1}_{type=1}(1 - \sum_{s+1}^{|\mathcal{V}^R|} \beta_i)$
if *type=1* **then**
  $\beta_s \leftarrow remainder$, $\beta_{|\mathcal{V}^R|+1} \leftarrow 0$
if *type=2* **then**
  $\beta_s \leftarrow 0$, $\beta_{|\mathcal{V}^R|+1} \leftarrow remainder$
// $U^D(\boldsymbol{\beta})$ is the defender's NE payoff
$U^D_{type} \leftarrow \min[\Lambda\boldsymbol{\beta}] - \boldsymbol{\mu}'\boldsymbol{\beta}$

Algorithm 3: Compute-$\alpha(\boldsymbol{\beta}, k)$

// $\boldsymbol{\beta}$ is the *set* of all convex combinations if multiple
for $i = 1$ **to** $k - 1$ **do**
  $\alpha_i \leftarrow 0$
for $i = k + 1$ **to** $|\mathcal{V}^R| - 1$ **do**
  $\alpha_i \leftarrow \dfrac{1-p}{p}\dfrac{c_{\mathrm{fa}}}{c_{\mathrm{d}}} P^R_N(r_i)$
if $\beta_k > 0$ *for some* $\boldsymbol{\beta}$ **then**
  $\alpha_k \leftarrow \dfrac{1-p}{p}\dfrac{c_{\mathrm{fa}}}{c_{\mathrm{d}}} P^R_N(r_k)$
  $\alpha_{|\mathcal{V}^R|} \leftarrow 1 - \sum_{i=k}^{|\mathcal{V}^R|-1} \alpha_i$
else if $\beta_{|\mathcal{V}^R|+1} > 0$ *for some* $\boldsymbol{\beta}$ **then**
  $\alpha_{|\mathcal{V}^R|} \leftarrow \dfrac{1-p}{p}\dfrac{c_{\mathrm{da}}}{c_{\mathrm{d}}} P^R_N(r_{|\mathcal{V}^R|})$
  $\alpha_k \leftarrow 1 - \sum_{i=k+1}^{|\mathcal{V}^R|} \alpha_i$;
else if $\beta_k = \beta_{|\mathcal{V}^R|+1} = 0$ **then**
  for $i = 1$ **to** $k - 1$ *and* $i = k + 1$ **to** $|\mathcal{V}^R| - 1$ **do**
    $\alpha^1_i \leftarrow \alpha_i$, $\alpha^2_i \leftarrow \alpha_i$
  $\alpha^1_k \leftarrow 0$
  $\alpha^2_k \leftarrow \min\left(\frac{1-p}{p}\frac{c_{\mathrm{fa}}}{c_{\mathrm{d}}} P^R_N(r_k), 1 - \sum_{i=k+1}^{|\mathcal{V}^R|-1} \alpha_i\right)$
  $\alpha^{1,2}_{|\mathcal{V}^R|} \leftarrow 1 - \sum_{i=k}^{|\mathcal{V}^R|-1} \alpha^{1,2}_i$
  $\boldsymbol{\alpha} \leftarrow$ convex hull of $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2$

A. Single-feature-based classification

We first consider $\mathcal{G}^{R,T} = (\mathcal{V}^R, \mathcal{C}^T, p, c_{\mathrm{d}}, c_{\mathrm{fa}}, P^{\mathrm{N}})$, in which classification is based on a single feature, i.e., there is a single target of interest and the defender observes how often this target is (or attempted to be) compromised. The attacker's strategy space consists of $N + 1$ attack rewards $r_0, \ldots, r_N$, with $r_i = i \cdot c_{\mathrm{a}}$ representing the attack reward when the target is compromised i times. The defender's strategy space $\mathcal{C}^T$ consists of all threshold classifiers on r_i. The attacker bears a cost of c_{d} upon detection. The false alarm penalty for the defender when she mistakenly classifies the non-attacker as an attacker is expressed by the factor c_{fa}.

The non-attacker, typically a normal user (or noise from the point of view of defender looking to do attacker detection), accesses the target i times with binomial distribution: attack reward r_i is the outcome of i successes over N trials with probability of success θ_0. His behavior results in a distribution

$$P_{\mathrm{N}}^{R} = \binom{N}{k} \theta_0^k (1-\theta_0)^{N-k}. \tag{19}$$

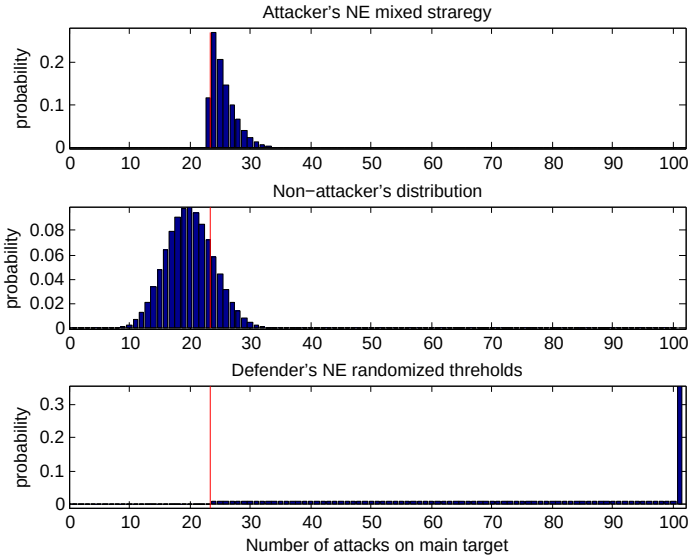

Fig. 3. Equilibrium distributions.

Figure 3 illustrates the equilibrium mixing distributions of the players, and the probability distribution of non-attackers, for the following particular choice of parameters: $c_{\mathrm{a}} = 1$, $c_{\mathrm{d}} = 120$, $p = 0.2$, $c_{\mathrm{fa}} = 140$, $\mathcal{V}^R = \{r_0, \ldots r_i, \ldots, r_{100}\}$, with $r_i = ic_{\mathrm{a}}$, and $P_{\mathrm{N}}^{R}(r_i)$ given by (19) with $\theta_0 = 0.2$. This example turns out to illustrate all of the major structural findings of the players' equilibrium distributions proved in Theorem 2. In particular:

i) The attacker uses a truncated, scaled version of the distribution of the non-attacker, but only on a subset of the support (the interior of the defender's support). Moreover, his strategy space comprises of actions that yield the highest payoffs.
ii) The defender uses a set of contiguous strategies (thresholds on the attack reward) that consist of the most rewarding attack vectors. This is in contrast with known algorithms such as logistic regression which have a predefined shape of the boundary independently of the attacker's goal.
iii) The weight assigned to each threshold is proportional to the marginal reward increase at that point. If the marginal reward increase is constant, the defender randomizes uniformly among strategies in the interior of her support.

Note that this equilibrium is calculated easily by using the results from Section IV, without any need for a complex program. For the defender, the weight given to each strategy is constant in the interior of the support, equal to $\beta_i = \frac{r_i - r_{i-1}}{c_{\mathrm{d}}} = \frac{c_{\mathrm{a}}}{c_{\mathrm{d}}} = \frac{1}{120}$.

In the following, we see how changes in the parameters affect the players' equilibrium payoffs. First we study how the equilibrium payoffs change as c_{a}, the cost to the defender of a single attack (or conversely reward to the attacker), increases and c_d, the cost to attacker of a detection event, is fixed. For instance, consider the real world problem of online click fraud which is prominent in pay-per-click (PPC) online advertising. Owners of websites that post the ads are paid an amount of money determined by how many visitors to the sites click on the ads. Malicious owners of sites could choose to use low-priced key words to attract less attention from the fraud classifiers of the ad-network, or be more aggressive and pick high-priced keywords. If the expected reward is very high, the attacker might as well attack with full strength (e.g., choose high-priced keywords) to get the revenue generated by the ads, and risk immediate detection. As we see in Fig. 4, when c_{a} rises, the attacker's strategy becomes more and more concentrated on attacking the target with more frequency, since the expected cost of being detected becomes relatively less important. In fact, in the extreme case where c_{a} is much larger than c_{d}, the attacker always attacks the maximum number of times even though he is always detected in the process. Increasing c_{a} relative to c_{d} gives the attacker more power to earn reward at the defender's cost, without giving the defender any corresponding increase in the ability to "fight back." Thus one should expect that as c_{a} rises the equilibrium payoff to the attacker should rise and that of the defender should fall.

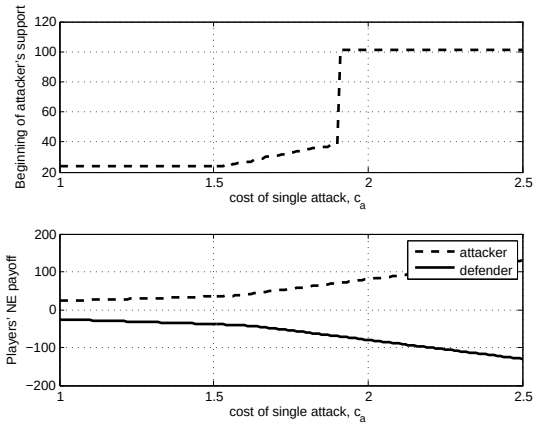

Fig. 4. Attacker's support lower bound (top) and equilibrium payoffs of attacker and defender (bottom) as c_{a}, the cost of a single attack, is varied.

In Figure 5 we observe the impact of the false alarm penalty parameter. In real world scenarios, the false alarm penalty incurred to the defender can differ a lot. For example, credit card companies care more about classifying a real user as fraudster than a mail provider who classifies an email as spam or not spam. As the false alarm penalty increases, the defender reduces false alarms by concentrating her distribution of threshold choice on higher values. Conversely, the attacker can exploit the higher thresholds by using attack distributions more concentrated on attack strategies that yield higher reward. Thus raising c_{fa} increases attacker payoff and decreases defender payoff in equilibrium.

Observe that in Fig. 5 the attacker's expected payoff has

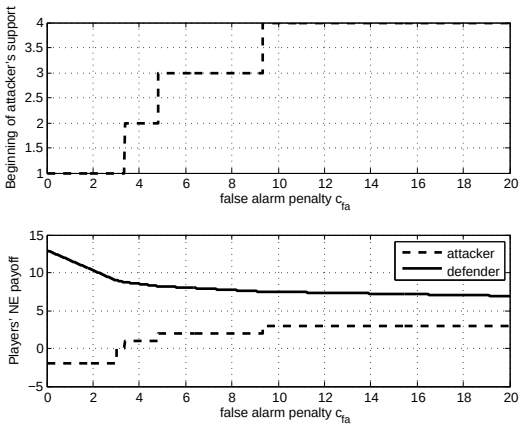

Fig. 5. Attacker's support lower bound (top) and equilibrium payoffs of attacker and defender (bottom) as c_{fa}, the cost of false alarm, is varied.

a staircase nature. This is an effect of the discrete optimal values of the starting index s of the contiguous block of tight inequality constraints: for some region of parameters, the same s is optimal. For the same optimal s, the support of the defender's strategy stays the same and the detection benefit, hence the attacker's equilibrium payoff is the same. The defender's staircase nature is slightly affected by the false alarm penalty factor in her payoff function. The above result suggests that in practice, the attacker is not so sensitive to false alarm penalty variations. Thus, even a few different buckets of estimated false alarm penalty costs to the defender would be sufficient for the attacker to compute his optimal strategy. Even if the underlying parameter cost fluctuates by a small amount, the attacker's strategy might remain the same.

B. Multiple Features: optimal investment defender strategies

In the previous experiments, classification was based on a single feature. In a security environment, one important decision of the defender (or network administrator) is to decide whether incorporating an additional feature would improve attacker classification, and improve it enough to justify any additional cost in collecting that data.

The feature vector of the defender, upon which the classification is based, may consist of multiple features. For example, the feature vector of Twitter's classification algorithms (which differentiate fraudulent accounts from legitimate ones), might consist of features such as the number of followers, average number of retweets per tweet, number of mentions, country of origin, and others. Before the defender decides to collect more features for her classification purposes, she should be aware that the most successful feature is the one that remains stealthy from the attacker. If she invests a lot into acquiring features that can be easily learned by the attacker, then the window of expected high reward (see Scenario 2 in Fig. 6) might be too small to compensate for the increased expenses of the feature acquisition and maintenance.

In this experiment we suppose that the defender observes which servers in her network a user accesses, and one of these servers is known to all parties to be particularly valuable. The observation of how many times this "valuable server" is accessed we designate as feature 1. A possible second feature the defender can use in classification is inspired by the literature of detecting portscanners [24]. Jung et al. found from empirical data a distinction between benign and malicious portscanners in terms of the proportions of the connections that were successfully established. In particular they define a metric, called **inactive-pct**, as the ratio of the number of hosts to which failed connections are made versus the number of hosts to which successful connections are made. Jung et al. found that benign users (e.g., web search engines) have a low **inactive-pct** ($< 80\%$), as a larger fraction of connections will be successfully established. On the contrary, malicious portscanners often have a high **inactive-pct** presumably because they initiate a lot of connections to detect vulnerabilities and terminate them immediately.

In our experiment, the attacker decides how many times to access the "valuable server" over a window of $N = 2$ time slots. Moreover, the attacker explores the other ports of the network, in parallel, looking for other targets of opportunity. He can access these other ports with a low or high **inactive-pct**. Scanning with a low **inactive-pct** is less efficient at finding targets, so it is less rewarding for the attacker. In this experiment, the attacker chooses to attack the "valuable server" either 0, 1, or 2 times, and also chooses whether to have a low or high **inactive-pct**. Thus there are $3 \times 2 = 6$ attack vectors. The reward for each is set to be $c_{\text{a}} = 1$ times the number of attacks of the "valuable server" plus $c_{\text{low}} = 2$ if **inactive-pct** is chosen low and $c_{\text{high}} = 4.1$ if **inactive-pct** is chosen high. We also suppose that $p = 0.2$ (the prior probability a user is an attacker), $\theta_0 = 0.3$ (the non-attacker's frequency of access to the "valuable server"), and $\theta^{\text{low}} = 0.8$ (the non-attacker's probability to have a low **inactive-pct**), $c_{\text{d}} = 1$ (the cost of detection), and $c_{\text{fa}} = 1$ (the cost of false alarm). The experiment consists of four scenarios:

1. The defender only observes feature 1, the number of times an agent accesses the "valuable server," and the attacker knows only feature 1 is being used in the classifier. Consequently, the attacker only uses the high **inactive-pct** strategy vectors since choosing a low **inactive-pct** only reduces reward without changing detection probability.
2. The attacker continues to play the equilibrium strategy of scenario 1 while the defender now has access to feature 2, **inactive-pct**, and uses this to optimize her classifier. The defender assumes, correctly for now, that the attacker keeps the same strategy as in scenario 1. The attacker continues to use the equilibrium strategy of scenario 1 because he does not "know" that the defender has access to feature 2.
3. The defender continues to use the equilibrium strategy of scenario 2, but now the attacker knows that the defender changed classifiers to use both features and optimized it against a scenario 1 attacker.
4. Both features are used by the defender, and it is common knowledge of both players that both features are being used.

As we see in Fig. 6, when the attacker does not know that the defender classifies him based on two features (scenario 2), the defender's payoff increases from scenario 1 in which the defender only classified the agent based on a single feature.

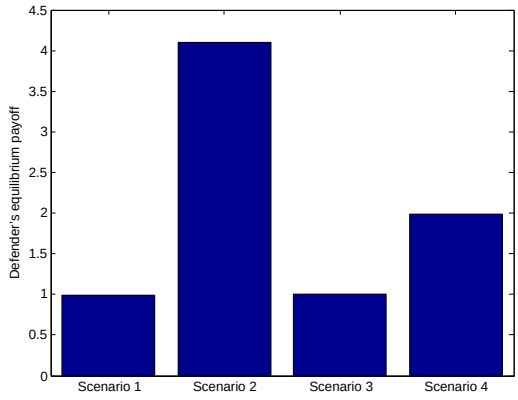

Fig. 6. Scenario 1: Defender does not differentiate between low and high "inactive-pct". Scenario 2: Attacker mistakenly believes he is only classified based on a single feature. Scenario 3: Defender mistakenly believes that attacker believes he is classified based on a single feature. Scenario 4: Both players know that two sensors (features) determine classification.

When the attacker finds out that he is getting classified on multiple features, then the defender's value decreases from scenario 2, since the attacker is smart enough to realize why he got detected. Comparing scenarios 1 and 4, we see that in this experiment the defender's NE payoff increases when she has access to two features, but the benefit is greatly diminished by the attacker's counter response to the new classifier.

VI. Concluding Remarks

In this paper, we propose and analyze a new game-theoretic model of adversarial classification. Contrarily to most previous research, our model takes into account the key differences between the objectives of the attacker and defender using a nonzero-sum game, yet it is simple enough to yield analytical results that bring intuitive insights into the structure of the Nash equilibrium and how classification should be performed against this type of adversary. In particular we show that, to remain stealthy while maximizing reward, the attacker mixes amongst attack vectors with a distribution proportional to the non-attacker's distribution (i.e., to normal behavior) but on a reduced support of attack vectors with highest rewards. The defender, on the other hand, mixes between classifiers that correspond to applying a threshold on the attacker's reward. This result intuitively shows that, in a strategic setting, the classifier should combine features according to the attacker's reward function (i.e., by looking at the reward a given feature/attack vector gives). Hence, using a standard classifier with a fixed shape of the decision boundary (such as logistic regression with a linear boundary) will necessarily be suboptimal regardless of how the parameters of the model are trained if the reward function does not have the predefined shape. Our results on the defender's equilibrium strategy also show the need for randomization (mixed strategy) and show that the weight assigned to each threshold is mainly proportional to the marginal reward increase at that point.

An important element in the tractability of our model is the special structure of the payoffs that makes our game best-response equivalent to a zero-sum game and hence allows us to use Linear Programming tools to search for all Nash equilibria. Surprisingly, although this equivalence seems straightforward, the literature in security games is largely looking at zero-sum games. We believe that models that better capture realistic scenarios in which the defender and attacker have different tradeoffs in their objective functions but are still computationally equivalent to zero-sum games could be studied using an approach similar to ours.

A major assumption of most game theoretic models is that players are rational – simply meaning that each player has preferences over the possible outcomes of the game that can be represented by assigning each outcome a payoff which each player tries to maximize in expectation. In many games such as ours, the solution concept that is most natural is that of a mixed strategy Nash equilibrium. However there are always questions of whether players will actually play these mixed strategies, and indeed this has been an area that the luminaries of game theory have given considerable thought over the years (see Rubinstein [42] for a review). As Rubinstein points out, while there are cases in which mixed strategies may be seen as an abstraction of interacting with a population of players, or decision making with private information, there are cases in which players have an incentive to randomize. It is the case in our game as both attacker and defender have an incentive to be unpredictable. Another domain in which agents are motivated to be unpredictable is sport, where empirical studies have shown players to randomize according to Nash equilibrium distributions [39]. The players likely are not calculating the Nash equilibrium analytically but instead are learning about the frequencies in which opponents play certain actions.

Finally, even if one doubts whether the attacker will behave "rationally," the defender has a compelling reason to consider following the mixed strategy prescribed by this work. The reason is that the defender's equilibrium mixed strategy minimizes the maximum cost the defender can suffer from the attacker, however the attacker chooses to play. This includes the possibility of an attacker that attacks in some non-strategic way, such as according to an automated script that is not finely tuned to be the optimal attack for a particular defender. On the other hand, a defender might wish to exploit the predictability of some non-strategic attackers by employing a multi-stage approach to first find the attackers who follow a consistent pattern before applying the approach described in this paper.

Overall, our paper makes a step towards building better attack detection systems using classification techniques that take into account the objective of the attacker to optimize the defense. Our results show that game theory is a valuable tool to tackle adversarial classification problems in settings that are not worst-case as it provides a formal way to justify the need for randomization and to optimize the defense distribution for a given attacker's objective. In future work, we plan to extend our results to more complex adversarial settings.

Acknowledgement

We would like to thank the anonymous reviewers and the associate editor for carefully reading of our manuscript and for providing insightful comments and suggestions.

REFERENCES

[1] T. Alpcan and T. Başar. A game theoretic approach to decision and analysis in network intrusion detection. In *Proceedings of IEEE CDC*, pages 2595–2600, 2003.

[2] T. Alpcan and T. Başar. *Network Security: A Decision and Game-Theoretic Approach*. Cambridge University Press, 2010.

[3] R. Avenhaus, B. Von Stengel, and S. Zamir. Chapter 51 inspection games. In R. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 3, pages 1947–1987. Elsevier, 2002.

[4] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[5] M. Barni and B. Tondi. The Source Identification Game: An Information-Theoretic Perspective. *IEEE Transactions on Information Forensics and Security*, 8(3):450–463, 2013.

[6] M. Barni and B. Tondi. Binary hypothesis testing game with training data. *IEEE Transactions on Information Theory*, 60(8):4848–4866, 2014.

[7] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.

[8] S. P. Bradley, A. C. Hax, and T. L. Magnanti. *Applied mathematical programming*. Addison-Wesley, Reading Mass., 1977.

[9] M. Brückner, C. Kanzow, and T. Scheffer. Static prediction games for adversarial learning problems. *Journal of Machine Learning Research*, 13:2617–2654, 2012.

[10] M. Brückner and T. Scheffer. Nash equilibria of static prediction games. In *Proceedings of NIPS*, pages 171–179, 2009.

[11] M. Brückner and T. Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of KDD*, pages 547–555, 2011.

[12] L. Chen and J. Leneutre. A Game Theoretical Framework on Intrusion Detection in Heterogeneous Networks. *IEEE Transactions on Information Forensics and Security*, 4(2):165–178, 2009.

[13] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player nash equilibria. *J. ACM*, 56(3):14:1–14:57, May 2009.

[14] N. Christin. Network Security Games: Combining Game Theory, Behavioral Economics, and Network Measurements. In *Proceedings of GameSec*, pages 4–6, 2011.

[15] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *Proceedings of ACM KDD*, pages 99–108, 2004.

[16] M. Dresher. A sampling Inspection Problem in Arms Control Agreements: a Game-theoretic Analysis. In *Memorandum RM-2972-ARPA, The RAND Corporation*, 1962.

[17] L. Dritsoula, P. Loiseau, and J. Musacchio. A Game-Theoretical Approach for Finding Optimal Strategies in an Intruder Classification Game. In *Proceedings of CDC*, 2012.

[18] L. Dritsoula, P. Loiseau, and J. Musacchio. Computing the Nash Equilibria of Intruder Classification Games. In *Proceedings of GameSec*, 2012.

[19] L. Dritsoula, P. Loiseau, and J. Musacchio. A game-theoretic analysis of adversarial classification. *arXiv:1610.04972*, 2017.

[20] F. Forges. Chapter 6 repeated games of incomplete information: Non-zero-sum. In R. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 1, pages 155–177. Elsevier, 1992.

[21] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

[22] A. Globerson and S. Roweis. Nightmare at test time: Robust learning by feature deletion. In *Proceedings of ICML*, 2006.

[23] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of ACM AISec*, pages 43–58, 2011.

[24] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. In *Proceedings of IEEE S&P*, 2004.

[25] N. Karmarkar. A New Polynomial-time Algorithm for Linear Programming. In *Proceedings of ACM STOC*, pages 302–311, 1984.

[26] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *J. Artif. Int. Res.*, 41(2):297–327, May 2011.

[27] B. Li and Y. Vorobeychik. Scalable optimization of randomized operational decisions in adversarial classification settings. In *Proceedings of AISTATS*, 2015.

[28] V. Lisý, R. Kessl, and T. Pevný. Randomized operating point selection in adversarial classification. In *Proceedings of ECML PKDD*, pages 240–255, 2014.

[29] Y. Liu, C. Comaniciu, and H. Man. A bayesian game approach for intrusion detection in wireless ad hoc networks. In *Proceeding GameNets*, 2006.

[30] D. Lowd and C. Meek. Adversarial learning. In *Proceedings of ACM KDD*, pages 641–647, 2005.

[31] D. Luenberger. *Linear and Nonlinear Programming*. Springer, 2nd edition, 2003.

[32] T. F. Lunt. A Survey of Intrusion Detection Techniques. *Computers and Security*, 12(4):405–418, June 1993.

[33] K.-W. Lye and J. M. Wing. Game strategies in network security. *Int. J. Inf. Secur.*, 4(1-2):71–86, 2005.

[34] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux. Game theory meets network security and privacy. *ACM Comput. Surv.*, 45(3):25:1–25:39, 2013.

[35] M. Maschler. A Price Leadership Method for Solving the Inspectors Non-constant Sum Game. In *Naval Research Logistics Quarterly*, 1966.

[36] J. Nash. *Non-Cooperative Games*. PhD thesis, Princeton University, 1950.

[37] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia. Misleading learners: Co-opting your spam filter. In P. S. Yu and J. J. P. Tsai, editors, *Machine Learning in Cyber Trust: Security, Privacy, and Reliability*. Springer, 2009.

[38] B. Nelson, B. I. P. Rubinstein, L. Huang, A. D. Joseph, S. Lau, S. Lee, S. Rao, A. Tran, and J. D. Tygar. Near optimal evasion of convex-inducing classifiers. In *Proceedings of AISTATS*, 2010.

[39] I. Palacios-Huerta. Professionals play minimax. *The Review of Economic Studies*, 70(2):395–415, 2003.

[40] R. W. Rosenthal. Correlated Equilibria in Some Classes of Two-Person Games. *In International Journal of Game Theory*, 3(3):119–128, 1974.

[41] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A Survey of Game Theory as Applied to Network Security. In *Proceedings of HICSS*, pages 1–10, 2010.

[42] A. Rubinstein. Comments on the Interpretation of Game Theory. *Econometrica*, 59(4):909–924, July 1991.

[43] R. Samusevich. Game theoretic optimization of detecting malicious behavior. Master's thesis, Czech Technical University in Prague, 2016.

[44] N. Sebe, I. Cohen, A. Garg, and T. S. Huang. *Machine Learning in Computer Vision*. Springer, 2005.

[45] G. Sierksma. *Linear and Integer Programming: Theory and Practice*. CRC Press, 2nd edition, 2002.

[46] R. Sommer and V. Paxson. Outside the Closed World: On Using Machine Learning For Network Intrusion Detection. In *Proceedings of IEEE S&P*, 2010.

[47] M. C. Stamm, W. S. Lin, and K. J. R. Liu. Forensics vs. anti-forensics: A decision and game theoretic framework. In *Proceedings of ICASSP*, pages 1749–1752, 2012.

[48] G. Stringhini, G. Wang, M. Egele, C. Kruegel, G. Vigna, H. Zheng, and B. Zhao. Follow the green: Growth and dynamics in twitter follower markets. In *Proceedings of IMC*, 2013.

[49] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[50] A. L. Tarca, V. J. Carey, X.-W. Chen, R. Romero, and S. Drăghici. Machine learning and its applications to biology. *PLoS Comput Biol*, 3(6):e116+, June 2007.

[51] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *Proceedings of USENIX Security*, pages 195–210, 2013.

[52] J. Tirole. *The Theory of Industrial Organization*. The MIT Press, 1988.

[53] J. J. P. Tsai and P. S. Yu, editors. *Machine Learning in Cyber Trust: Security, Privacy, and Reliability*. Springer, 2009.

[54] Y. Vorobeychik and B. Li. Optimal randomized classification in adversarial settings. In *Proceedings of AAMAS*, 2014.

[55] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *Proceedings of USENIX Security*, pages 239–254, 2014.

[56] Y. Zhou and M. Kantarcioglu. Adversarial learning with bayesian hierarchical mixtures of experts. In *Proceedings of SIAM SDM*, pages 929–937, 2014.

[57] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and B. Xi. Adversarial support vector machine learning. In *Proceedings of KDD*, pages 1059–1067, 2012.