Scalable Optimal Classifiers for Adversarial Settings under Uncertainty

Benjamin Roussillon, Patrick Loiseau

Scalable Optimal Classifiers for Adversarial Settings under Uncertainty*

Benjamin Roussillon[1] and Patrick Loiseau[1]

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG

Abstract. We consider the problem of finding optimal classifiers in an adversarial setting where the class-1 data is generated by an attacker whose objective is not known to the defender—an aspect that is key to realistic applications but has so far been overlooked in the literature. To model this situation, we propose a Bayesian game framework where the defender chooses a classifier with no *a priori* restriction on the set of possible classifiers. The key difficulty in the proposed framework is that the set of possible classifiers is exponential in the set of possible data, which is itself exponential in the number of features used for classification. To counter this, we first show that Bayesian Nash equilibria can be characterized completely via functional threshold classifiers with a small number of parameters. We then show that this low-dimensional characterization enables us to develop a training method to compute provably approximately optimal classifiers in a scalable manner; and to develop a learning algorithm for the online setting with low regret (both independent of the dimension of the set of possible data). We illustrate our results through simulations.

1 Introduction

Detecting attacks such as spam, malware, or fraud is a key part of security. This task is usually approached as a *binary classification* problem where the defender classifies incoming data (login pattern, text features, or other data depending on the application) as legitimate (non-attack, modeled as class 0) or malicious (attack, modeled as class 1) [47, 8].

It is well known that using standard classification algorithms for this task leads to poor performance because attackers are able to avoid detection by adjusting the data that they generate while crafting their attacks [36, 45, 46, 51]. There is a vast literature on adversarial classification (see Section 1.1), but these works often propose ad-hoc defense methods optimized against specific attacks without fully modeling the attacker's adaptiveness. This leads to an arms race between attack and defense papers.

To better take into account the interaction between attacker and defender, several game-theoretic models of adversarial classification have emerged over the

* This work was supported by the French National Research Agency through the "Investissements d'avenir" program (ANR-15-IDEX-02) and through grant ANR-16-TERC0012; by the DGA; by the Alexander von Humboldt Foundation.

last decade (see Section 1.1). Most of them, however, have two crucial limitations. First, they restrict the possible classifiers to a specific set of known parameterized classifiers and assume that the defender only selects those parameters. Second, they assume complete information about the attacker's objective,[1] which is often too strong in practice [50].

In a recent paper, Dritsoula et al. [15] propose a model where the defender can select *any* classifier (i.e., function from the set of data to $\{0,1\}$). A key difficulty lies in the exponential size of the resulting set of classifiers. The authors show that it is possible to restrict it to a small set of *threshold* classifiers on a function that appears in the attacker's payoff. The classifiers identified, however, have no parameter and their solution method is ad-hoc for the restrictive model chosen—with complete information and simplistic payoffs—, hence it cannot extend to more realistic scenarios. In realistic adversarial classification scenarios with uncertainty on the attacker's payoff, this leaves open the questions: *What classifiers shoud the defender use at equilibrium? And how to compute optimal classifiers in a scalable manner?*

In this paper, we answer both questions through the following contributions:

1. We introduce structural extensions to the model of [15] where the defender can choose any function from a set of data $\mathcal{V}$ to $\{0,1\}$ as a classifier: we model the uncertainty of the defender on the attacker's payoff as a Bayesian game and use generalized payoffs (see Sec. 2.3).
2. We characterize the equilibrium of the game and exhibit a set of optimal threshold classifiers depending on a small number of parameters (in number independent of $|\mathcal{V}|$). Our method first uses a classical technique in resource allocation games (see Sec. 1.1) to establish a link between a mixed strategy on the set of classifiers and a 'random classifier' that assigns a probability in $[0,1]$ to every data vector $v \in \mathcal{V}$. The set $\mathcal{V}$, however, is still exponentially large—this is the key challenge in our work. We then show that the 'random classifier' used by the defender at equilibrium has a specific form described with a small number of parameters, and that finding it is equivalent to maximizing a piecewise linear function of the previously mentioned parameters. This low-dimensional characterization has many interesting consequences: it enables using classical stochastic programming and online optimization techniques for efficient learning both online and offline in our game.
3. We show that our parametric expression of equilibrium classifiers allows the defender to train parameters on a labeled dataset with access to only limited information. In particular, our training method, which leverages classical stochastic programming techniques combined with our low-dimensional characterization, produces error bounds independent of $|\mathcal{V}|$ and does not require knowledge of the non-attack distribution. This gives much desired scalability since $|\mathcal{V}|$ is exponential in the number of features and might be large.
4. We illustrate our results through numerical simulations on different games, in particular a credit card fraud game built from the distributions in the publicly available real-world dataset [48] introduced in [12].

[1] with the exception [20], but which considers regression.

5. We also show that our parametric expression of equilibrium classifiers allows the defender to learn in an online setting—where they update the classifier and receive feedback from the classification at each time step—with very little regret (in particular, independent of $|\mathcal{V}|$). Additional illustrations for this setting are provided in [32].

Our results provide a basis for designing provably robust classifiers for adversarial classification problems. Our characterization of equilibrium strategies also emphasizes the potential of randomized operating point methods from [31]: our final classification algorithm can be seen as a randomized operating point on a non-trivial class of optimal threshold classifiers (see discussion below Theorem 1). Interestingly, we find that the set of optimal randomized defenses is of low pseudo-dimension. This highlights recent results by Cullina et al. [11]: in our model, facing adversaries simplifies the learning process as worst-case attacks are predictable while classical learning is chaotic (see discussion below Proposition 1). This is further supported by our finding that the set of optimal threshold classifiers is of VC dimension 1 [43]; hence our result could be interpreted in hindsight as showing that a reduction to classifiers of VC dimension 1 would come at no loss to the defender. Yet, we emphasize that there is no reason *a priori* why this set would be sufficient, it is a consequence of our results.

Due to space constraints, proofs and additional illustration are presented as supplementary material in [32].

1.1 Related work

Adversarial learning: The literature on adversarial learning usually studies two types of attacks: 'poisoning attacks', where the attacker can alter the training set to tamper the classifier's training [13, 18, 2, 27, 21, 52]; and 'evasion attacks', where the attacker tries to reverse engineer a fixed classifier to find a negative instance of minimal cost [33, 35, 29]. This literature, however, does not fully model the attacker's adaptiveness, which often leads to an arms race. In recent years, the adversarial learning research focused on evasion attacks called adversarial examples that affect deep learning algorithms beyond attack detection applications [19, 38, 37]. These works, however, follow the same pattern. [11] extend PAC theory to adversarial settings and show that fundamental learning bounds can be extended to this setting and that the adversarial VC dimension can be either larger or smaller than the standard one.

Game-theoretic models of adversarial classification: A number of game-theoretic models of adversarial classification have been proposed, with various utility functions and hypotheses on the attacker's capabilities. Most of them, however, restrict a priori the possible classifiers: [53, 52] rely on kernel methods; [23] assumes that the defender uses a single type of classifier (though unspecified in the model); [13] focuses on naive Bayes classifiers (and only compute one-stage best responses); [6, 5] constrain the classifier to a specific form and look for the (pure) equilibrium value of the parameters; [28] uses a different model but also

restrict to linear classifiers; [14] restricts the defender to a set of adversarially trained classifiers of different strengths; [29] uses a more general classifier, but restricts for most results to a family of classifiers constructed on a given basis (their model of the attacker is also more constrained than ours); and [31] abstracts away the classifier through a ROC curve (attacker and defender only select thresholds). In contrast, the objective of our work is to derive the optimal form of the classifiers so we do not make any restriction a priori on the classifiers used.

At the exception of [31, 29], the aforementioned papers build deterministic classifiers while recent papers tend to advocate for randomization: [7] introduces random strategies on top of [5] while [39] highlights the importance of randomized attacks and [40] of randomized defenses (albeit without being able to characterize the equilibrium). In our work, we completely characterize the equilibrium and naturally find that it must involve randomized attack and defense strategies.

It is important to understand that these works consider two main types of model. [6, 5, 40] study *adversarial learning* problems where the learning problem is defined even without attackers (e.g., image recognition), whereas [13, 31, 53, 52, 23, 28, 29] study *adversarial classification* where the learning problem is to detect attacks and exists only because there are attackers (e.g., spam filtering). These models lead to different attack methods and defenses. Our work belongs to the second category, of adversarial classification problems.

Security games: Our game has similarities with *security resource allocation* games [9, 24, 3, 34, 16, 1, 42, 4] used in applications such as airport security [41]. These works consider a defender with limited resources (e.g. guards) to be allocated to the defense of critical targets. In these settings, problems are at a relatively low scale and are usually entirely described via loss in case of attack of an undefended target. The challenge is the management of the limited amount of resources, which produces NP-hard problems [26] preventing these models to be transferred to very large scale settings. Our work studies a similar setting applied to classification, where targets would correspond to attack vectors in $\mathcal{V}$. In contrast to the security games literature, we do not impose limited resources (the defender self-restricts its detection to limit false alarm costs whereas in security games resources lead to hard constraints on the possible strategies), which eliminates the combinatorial issue. We are then able to provide a very different characterization of the solutions with applicability to classification as well as to scale to very large sets $\mathcal{V}$ that is never studied in classical security games and is the major challenge in our model.

Exponential zero-sum games: Our game reparametrization with 'randomized classifiers' to reduce the dimension of the set of classifiers from $2^{|\mathcal{V}|}$ to $|\mathcal{V}|$ borrows ideas classical in security games. This technique is also studied for more generic zero-sum games [22]; but with objectives and limitations similar to security games.

2 Model

In this section we present our game-theoretic model. We introduce utility functions from the defender's viewpoint as we focus on optimal classifiers. We then introduce the probability of detection function as a tool to reduce complexity and discuss the model's assumptions and applicability.

2.1 Setting and notation

Consider the following situation. A defender receives data samples that can be either attacks (class 1) or non-attacks (class 0) and wants to predict the class of incoming data. We assume that a data example is represented by a feature vector v that belongs to the same set $\mathcal{V}$ regardless of the class. This vector is typically a simplified representation of the actual attack/non-attack (e.g., spam/non-spam) in a feature space used to perform the classification. We assume that the probability that a data example is an attack, denoted p_a, is fixed.

Vectors corresponding to non-attacks follow a fixed probability distribution P_0 on $\mathcal{V}$ whereas vectors corresponding to attacks are generated by attackers. Attackers choose the vector they generate to maximize a utility function (see below) depending on the classification of the defender. To model the uncertainty of the defender, we assume that strategic attackers are endowed with a type $i \in \{1, \ldots, m\}$ that encodes their utility. The defender does not know the type of the attacker but holds a prior $(p_i)_{i\in\{1,\ldots,m\}}$ on the possible types.

The defender chooses a classifier in $\mathcal{C} = 2^{\mathcal{V}}$, that maps a vector to a predicted class. The defender maximizes a utility function balancing costs/gains in different cases as follows. A *false negative* incurs a loss $U_i^u(v)$ when facing a type-i attacker. A *true positive* incurs a gain $U_i^d(v)$ when facing a type-i attacker. A *false positive* incurs a false alarm cost $C_{\text{fa}}(v)$. A *true negative* incurs no cost. The attacker's gain is the opposite of the defender's for each classification outcome.

Summarizing the above discussion, the utilities of the attacker and defender, when the attacker is of type i, are defined as follows:

$$\begin{aligned} U_i^A(v,c) =& U_i^u(v)\mathbb{1}_{c(v)=0} - U_i^d(v)\mathbb{1}_{c(v)=1}, \\ U_i^D(v,c) =& -p_a U_i^A(v,c) - (1-p_a)\sum_{v'\in\mathcal{V}} C_{\text{fa}}(v')P_0(v')\mathbb{1}_{c(v')=1}. \end{aligned} \tag{1}$$

We assume that $\mathcal{V}$ is finite and all functions of v are arbitrary. Our main result, however, extends to $\mathcal{V}$ compact (details in [32] due to space constraints).

The above primitives define a Bayesian game that we denote by $\mathcal{G}$. Note that we assume that all parameters of the game including p_a, P_0, and the utility functions (but not the attacker's type) are known to both players. (We will discuss later how to relax this assumption.) As we will see, in this game, equilibria exist only in mixed strategy (intuitively, both players have an incentive to be unpredictable). For the defender, a mixed strategy β is a probability distribution on $\mathcal{C}$. A mixed strategy of the attacker is a function $\alpha : \{1, \ldots, m\} \to \Delta(\mathcal{V})$ such that for all $i \in \{1, \ldots, m\}$, $\alpha_\cdot^i$ is a probability distribution over $\mathcal{V}$ chosen by a

type-i attacker. Throughout the paper, we will use the standard solution concept of Bayesian Nash equilibrium, which intuitively prescribes that no player can gain from unilateral deviation.

Definition 1. *(α^*, β^*) is a Bayesian Nash equilibrium (BNE) of the game $\mathcal{G}$ if and only if, for all α, β,*

$$\sum_{i\in\{1,\dots,m\}} p_i U_i^D(\alpha^*, \beta^*) \geq \sum_{i\in\{1,\dots,m\}} p_i U_i^D(\alpha^*, \beta), \text{ and} \tag{2a}$$

$$\sum_{i\in\{1,\dots,m\}} p_i U_i^A(\alpha^*, \beta^*) \geq \sum_{i\in\{1,\dots,m\}} p_i U_i^A(\alpha, \beta^*). \tag{2b}$$

The defender's utility depends on the attacker they face. With the belief the defender holds on the probability of each attacker type, it is natural that the defender tries to maximize their average utility. The equilibrium is also described with the average utility of the different attacker types, but as the actions of different attacker types are unrelated it is equivalent to each type maximizing its own utility.

Finally, for all $i \in \{1, \dots, m\}$, we define $\underline{G}_i = \max_{v\in\mathcal{V}} \left(-U_i^d(v)\right)$ and $\overline{G}_i = \max_{v\in\mathcal{V}} \left(U_i^u(v)\right)$, which respectively represent the minimum possible gain of the attacker (even if all vectors are always detected they can gain this quantity) and their maximum possible gain. Note that all results in our paper assume knowledge of these bounds (even when knowledge of utilities is limited). While finding these intervals is challenging if the utilities are arbitrary, they are easy to find in many applications from reasonable monotonicity assumptions on the utilities, as they simply represent the most damage an undetected/detected attack can cause.

2.2 Preliminary: reduction of dimensionality

A first difficulty of the model we study is the exponential size of $\mathcal{C}$ in $\mathcal{V}$. This issue is commonly found in resource allocation games (similar reparametrizations are found in other games such as dueling algorithms) and circumvented through the use of a probability of allocation function: only the probability that an abstract resource is allocated to a target is considered thus ignoring the actual allocation and removing combinatorial complexity (assuming that one can compute this function at equilibrium). In our case, in the spirit of [15], we define a probability of detection π, for any strategy β of the defender, as $\pi^\beta(v) = \sum_{c\in\mathcal{C}} \beta_c \mathbb{1}_{c(v)=1}$.

This transformation exploits the fact that, as long as a vector is detected, the actual classifier used for the detection is not important. Thus, with this probability of detection function, we can rewrite the payoffs independently of classifiers:

$$\begin{aligned} U_i^A(\alpha, \beta) &= \sum_{v\in\mathcal{V}} \alpha_v^i \left[U_i^u(v) - \pi^\beta(v) \cdot \left(U_i^u(v) + U_i^d(v)\right)\right]; \\ U_i^D(\alpha, \beta) &= -p_a U_i^A(\alpha, \beta) - (1 - p_a) \sum_{v\in\mathcal{V}} C_{\text{fa}}(v) P_0(v) \pi^\beta(v). \end{aligned} \tag{3}$$

Any probability of detection function can be attained through simple threshold classifiers crafted for this function. To see this, consider the set of threshold classifier $c(v) = \mathbb{1}_{\pi^\beta(v) \geq t}$ for some $t \in [0, 1]$. Then, picking a random threshold uniformly on $[0, 1]$ defines a strategy achieving detection probability $\pi^\beta(\cdot)$.

2.3 Model discussion

The main motivating scenarios for our model are detection of malicious behaviors such as spam (in emails, social media, etc.), fraud (e.g., bank or click fraud), or illegal intrusion. In such scenarios, the attacker is the spammer, fraudster or intruder while the non-attacker represents a normal user (e.g., non-spam message). The vector v is a representation of the observed behavior on which the classification is done. For spam filtering, it can be a simplified representation of the messages obtained by extracting features such as number of characteristic words. The distribution P_0 represents the distribution over those features for normal messages (not chosen with any adversarial objective). In our basic model, we assume that it is known by both players. It is reasonable in applications where it can be estimated from observation of a large number of easily obtainable messages (e.g., in social medias they are public). We relax it in Section 3.2 and Section 4 where we show that the defender can learn well without a priori knowledge of P_0, p_a and p_i.

In our model the defender is uncertain of its own utility as soon as they have uncertainty regarding the attacker they face. Although not the most classical setting, it is meaningful and well studied in Bayesian games (see [17]). It is well justified in our case. For instance, if a fraudster manages to get access to sensitive information or to an account, the amount of harm may differ depending on the skills and resources of the fraudster. In these fraud settings it makes sense that the attacker's gain is the defender's loss or a fraction of it (e.g. a bank must reimburse its clients or pay higher insurance fees if it is victim of fraud). We note here that our model is still valid for this last case as we rely on zero-sum min-max properties which are robust to small changes such as multiplying factors.

The interaction between classifier and attacker is often modeled as a Stackelberg game where the attacker observes and reacts to the defender's strategy. We focus on the (Bayesian) Nash equilibrium which makes sense if the attacker cannot have perfect information about the defender's strategy. More generally though, we will see that in our game the defender's strategy at BNE must be min-max; hence, any strategy of the defender in a Stackelberg equilibrium would have the same property. We use the Stackelberg model in the online setting where there would be a bigger difference. Note that this min-max property also yields robustness.

Our payoff function generalizes that of [15] in a practically important way. In their model, a reward $R(v)$ is granted to an attack with vector v regardless of the outcome and a fixed detection cost c_d is paid if the attack is detected. This is unreasonable in many applications such as bank fraud. In our model, the utility in case of detected and undetected attacks are arbitrary unrelated functions of v (which is equivalent to letting the detection cost c_d depend on v). This alone

breaks the ad-hoc method of [15] to compute the equilibrium. We also generalize to a Bayesian game (The complete information game is the case where $m = 1$), and consider training and online learning problems of practical importance.

3 BNE characterization and computation

In this section, we first characterize the equilibrium entirely and exhibit a class of threshold classifiers which are sufficient to define an optimal classifier. Leveraging this characterization, we then show how to compute approximately optimal strategies through training with limited knowledge.

3.1 Equilibrium characterization

Finding a Bayesian Nash equilibrium is often hard in general games. A key property is that our game is essentially zero-sum and can be reduced to a min-max problem. Compiling this with the action space reduction via the probability of detection we are able to completely characterize the BNE.

Using the payoffs defined in (1), we can see that adding the false alarm term to the payoff of the attacker gives an equivalent Bayesian zero-sum game (as this term is independent from the action of the attacker this addition does not change their strategy). This transformation does not change the defender's payoff. This implies that at equilibrium they maximize their minimum average gain and gives the following lemma (whose proof can be found in [32]):

Lemma 1. *Let (α^*, β^*) be a BNE. Then*

$$\beta^* \in \arg\max_{\beta} \min_{\alpha} \sum_i p_i U_i^D(\alpha, \beta). \tag{4}$$

Computing the min-max strategies of Lemma 1 can be done via a classical transformation to a linear program, but this "naive" program would be of size exponential in $|\mathcal{V}|$. Even by expressing it in terms of π^β, the program would remain of size $|\mathcal{V}|$, which may be too large. Instead, we will leverage the min-max property to show that the equilibrium can be described compactly using a small number of parameters $\boldsymbol{G} = (G_1, \cdots, G_m)$ that can be interpreted as the utility of the attacker for each type. Formally, we define:

Definition 2 (Optimal probability of detection). *For any $\boldsymbol{G} \in [\underline{G}_1, \overline{G}_1] \times \ldots \times [\underline{G}_m, \overline{G}_m]$, let*

$$\pi_{\boldsymbol{G}}(v) = \max\left\{0, \max_i \left\{\frac{U_i^u(v) - G_i}{U_i^u(v) + U_i^d(v)}\right\}\right\}, \ \forall v \in \mathcal{V}. \tag{5}$$

As we will see, this quantity is the unique probability of detection that guarantees attacker utility below $\boldsymbol{G}$ while minimizing the false alarms, so it plays a key role in the BNE strategy. In particular, it allows us to express the strategy of the defender as the maximum of a concave function of $\boldsymbol{G}$:

Definition 3 (Minimum gain function U^D). *For all* $\boldsymbol{G} \in [\underline{G}_1, \overline{G}_1] \times ... \times [\underline{G}_m, \overline{G}_m]$, *let* $U^D(\boldsymbol{G}) = -p_a \sum_i p_i G_i - (1 - p_a) \sum_{v \in \mathcal{V}} C_{fa}(v) P_0(v) \pi_{\boldsymbol{G}}(v)$.

This function represents the minimum utility of the defender assuming they use a probability of detection function $\pi_{\boldsymbol{G}}(\cdot)$ for some $\boldsymbol{G}$. It allows us to state our parametrization result which is the main tool we use to prove all our core results.

Proposition 1. *For any* $\boldsymbol{G}_{\max} \in \arg\max_{\boldsymbol{G} \in [\underline{G}_1, \overline{G}_1] \times ... \times [\underline{G}_m, \overline{G}_m]} (U^D(\boldsymbol{G}))$, *any strategy of the defender that yields a probability of detection function* $\pi_{\boldsymbol{G}_{\max}}(v)$ *for all* $v \in \mathcal{V}$ *is a min-max strategy and* $\max_\beta \min_\alpha \sum_i p_i U_i^D(\alpha, \beta) = U^D(\boldsymbol{G}_{\max})$.

A proof of Proposition 1 can be found in [32]. The proof relies on the min-max property of the problem which implies that the defender must maximize their minimum gain. We show that for a given utility profile $\boldsymbol{G}$, the minimum gain of the defender as defined in (4) is at least $U^D(\boldsymbol{G})$. However, the key difficulty is that not all utility profiles $\boldsymbol{G} \in [\underline{G}_1, \overline{G}_1] \times ... \times [\underline{G}_m, \overline{G}_m]$ are feasible and the set of feasible utility profiles needs not be convex due to our Bayesian game and arbitrary functions; hence $U^D(\boldsymbol{G})$ could be meaningless. Our proof bypasses this difficulty by showing that $\pi_{\boldsymbol{G}_{\max}}(\cdot)$ is a min-max strategy in any case and shows as a corollary that $\boldsymbol{G}_{\max}$ is a feasible utility profile.

Proposition 1 states that in order to find the equilibrium strategy, the defender should only find m parameters $(G_1, \cdots, G_m)$, corresponding to the maximum utility that it should let each attacker type gain. From those parameters, the probability of detection function is naturally defined. This has multiple consequences.

First, from this characterization we deduce that one does not need to know all the parameters of the problem to find a good strategy. Finding "good enough" parameters for the utility of the different attacker types allows the defender to fully define its strategy. This is the main tool allowing us to define strategies which can generalize to unknown vectors in Section 3.2. In particular, in Theorem 2 we prove that near-optimal (and even optimal with high probability) classifiers can be computed by training the model on a labeled dataset with very limited information. Note that this is a key difference between our work and security games where the probability of allocation is computed directly using a linear program. There, the lack of a simple expression for the allocation probability prevents the definition of strategies that can generalize. It is also worth noting that unlike linear programs, our method can be generalized to a continuous vector set—we refer to [32] for details about that.

Second, the result from Proposition 1 shows that the presence of strategic adversaries *simplifies* learning in our problem. Indeed, the class of real valued functions $\{\pi_{\boldsymbol{G}}\}$ which contains the optimal strategy is of low pseudo-dimension (e.g., if there exist v_1 (resp. v_0) of class 1 (resp 0) with $U^u(v_0) > U^u(v_1)$ and $U^d(v_0) < U^d(v_1)$, these two points cannot be shattered). This can be explained by the predictable aspect of adversaries acting according to their best-response. On the contrary, when facing non-strategic adversaries the optimal strategy would be a cost-sensitive adaptation of the naive Bayes classifier, which can potentially

be any arbitrary function of $2^{\mathcal{V}}$ (since we make no assumption on P_0). This is noteworthy as such a possibility was hinted at by Cullina et al. [11] who show that, for adversaries who can modify vectors in some neighborhood, the adversarial VC dimension can be either lower or higher than the standard one—i.e., the complexity can either increase or decrease in the presence of adversaries. In our adversarial classification model, the complexity drastically decreases. This suggests that classifiers relying on simply adapting classical training might be inefficient as they do not take into account the fundamental complexity differences between classical and adversarial learning.

With Proposition 1 describing the probability of detection function at equilibrium, we can deduce a characterization in terms of threshold classifiers.

Definition 4 (Generalized threshold classifiers). *For all $\boldsymbol{G} \in \mathbb{R}^m$, define*

$$\mathcal{C}^T_{\boldsymbol{G}} = \{c \in \mathcal{C} : c(v) = \mathbb{1}_{\pi_{\boldsymbol{G}}(v) \geq t}, \forall v \in \mathcal{V} \text{ for some } t \in [0,1]\}.$$

Theorem 1. *There exists $\boldsymbol{G} \in \mathbb{R}^m$ such that the defender can achieve equilibrium payoff using only classifiers from $\mathcal{C}^T_{\boldsymbol{G}}$.*

This theorem settles our first main question: "which classifiers should the defender use at the equilibrium?". These are threshold classifiers on a non-standard function with threshold t representing a probability of detection. A threshold t can be interpreted as classifying a vector as an attack if, even when being detected with probability t, at least one type of attacker gains at least G_i on average. Interestingly, $\mathcal{C}^T_{\boldsymbol{G}}$ has a VC dimension of only 1 as the set comprised of v_1 (resp. v_0) of class 1 (resp. 0) with $\pi_{\boldsymbol{G}}(v_1) < \pi_{\boldsymbol{G}}(v_0)$ cannot be shattered. This strengthens our previous remark on the complexity of adversarial classification. Efficient randomized classification for adversarial settings does not require high capacity classifiers but rather classifiers tailored to the players payoffs. Then, our threshold classifiers may be linear classifier if payoffs are linear as the condition $\pi_{\boldsymbol{G}}(v) \geq t$ can be rewritten as $\max_i \left\{U_i^u(v) - G_i - t(U_i^u(v) + U_i^d(v))\right\} \geq 0$. Thus, in the linear setting, our threshold classifiers correspond to the defender picking a linear classifier for each type of attacker and outputting class 1 if at least one of the linear classifiers outputs it. In general however, linear classifiers may perform sub optimally.

The fact that the defender uses specifically threshold classifiers is noteworthy as there is already a literature on the choice of threshold and on this choice in an adversarial setting as in [31]. However, the random choice of the threshold in our setting is surprisingly simple – it is a threshold on the probability of detection and choosing a threshold uniformly over $[0, 1]$ gives the desired strategy. This emphasizes that randomization is necessary to defend against an adversary but also that the choice of the set of classifiers to use is crucial to obtain good results.

Having characterized the equilibrium, we must now answer our second main question "How can the defender compute optimal strategies in a scalable manner?". Before presenting a scalable training procedure exploiting our equilibrium parametrization to compute an approximate equilibrium (Section 3.2), let us notice that the equilibrium characterization naively leads to a linear programming

solution polynomial in $|\mathcal{V}|$ to compute an exact equilibrium as function U^D is piecewise linear. This is presented in Proposition 2; note that a similar program could be obtained without our equilibrium characterization. We give in [32] a linear program that allows computing the attacker's strategy in time polynomial in $|\mathcal{V}|$.

Proposition 2. *Maximizing $U^D(\boldsymbol{G})$ is equivalent to solving the linear program:*

$$\begin{aligned} \underset{\pi,\boldsymbol{G}}{maximize} \;& -p_a \sum_{i=1}^{m} p_i G_i - (1-p_a) \sum_{v \in \mathcal{V}} C_{fa}(v) P_0(v) \pi_v \\ subject\ to: \; & G_i \geq U_i^u(v) - \pi_v (U_i^u(v) + U_i^d(v)), \forall i, \forall v \\ & \pi_v \leq 1, \forall v. \end{aligned}$$

3.2 Scalable approximate computation

Our previous results allow computing the equilibrium in time polynomial in $|\mathcal{V}|$. Yet, two major challenges remain: (i) $|\mathcal{V}|$ may be too large, in particular it grows exponentially with the number of features k; and (ii) computing the equilibrium requires knowledge of all parameters of the game and in particular of P_0, which can be hard to evaluate. In this section, we propose a training method that solves both issues by leveraging stochastic programming techniques. To do so, we first express $U^D(\boldsymbol{G})$ as an expected value as follows: $U^D(\boldsymbol{G}) = E[U^D(\boldsymbol{G}, \xi)]$ where $U^D(\boldsymbol{G}, \xi) = G_i$ with probability $p_a p_i$ and $U^D(\boldsymbol{G}, \xi) = C_{\text{fa}}(v)\pi_{\boldsymbol{G}}(v)$ with probability $(1-p_a)P_0(v)$ for all $v \in \mathcal{V}$. Leveraging the specific form of this stochastic function, we apply a stochastic programming technique called sample average approximation (SAA) [44, 25, 49, 30] to obtain our training method, Algorithm 1.

Algorithm 1 Sample average approximation

Sample $\xi_1 \ldots, \xi_N$
Define $\tilde{U}^D(\boldsymbol{G}) = 1/N \sum_{i=1}^{N} U^D(\boldsymbol{G}, \xi_i)$
Maximize $\tilde{U}^D(\boldsymbol{G})$ on $[\underline{G}_1, \overline{G}_1] \times ... \times [\underline{G}_m, \overline{G}_m]$

The maximization step in Algorithm 1 can be done exactly through a linear program in the spirit of Proposition 2, in time polynomial in N since $\tilde{U}^D(\boldsymbol{G})$ is piecewise linear. Thus the complexity of this algorithm depends only on the sample size and not on the problem dimension. Additionally, very little information is required: the defender only needs to have access to N samples, which may correspond to a labeled dataset, as well as to the parameters $C_{\text{fa}}(v)$, $U_i^u(v)$, $U_i^d(v)$ for those samples, and $\underline{G}_i$, $\overline{G}_i$. Yet the following theorem shows that Algorithm 1 outputs an very good approximation of the defender's min-max strategy.

Theorem 2. *Let $\hat{\mathcal{S}}$ be the set of maximizers of $\tilde{U}^D(\boldsymbol{G})$ from Algorithm 1 and $p_N = Pr[\hat{\mathcal{S}} \subseteq \arg\max U^D(\boldsymbol{G})]$. We have*

$$\limsup_{N\to\infty} \frac{1}{N} \log(1 - p_N) < 0.$$

A proof of Theorem 2 can be found in [32]. It relies on a strong result for sample average approximation (Theorem 15 of [44]), which fully exploits the structure of our problem as it requires the optimized stochastic function to be piecewise linear and to depend on random variables with finite support (extensions to continuous supports are possible under mild assumptions). This result is then enabled by the polyhedral structure of the problem.

Theorem 2 states that Algorithm 1 will find an exact maximum of $U^D(\boldsymbol{G})$ with probability exponentially close to one (where the randomness is in the draw of the training set from unknown P_0, p_a and p_i). Then, from Theorem 1, this immediately gives an exact min-max strategy of the defender. The rate of the exponential convergence of p_N to 1 is not given by Theorem 2. It is possible to state a stronger result that gives the rate if the problem is "well conditionned"—which roughly means that $\arg\max U^D(\boldsymbol{G})$ is a singleton and the function is not flat around the optimum—, but this is not guaranteed in any instance of our game, and such a result is anyways impractical because it depends on the true optimal value. From the high-probability result of Theorem 2, it is easy to derive that the output of Algorithm 1 is exponentially close to the true optimum since the function is bounded; although the exponential rate may be arbitrarily low if the problem is not well conditioned. In that case, though, worst case bounds show convergence of expected value at least in $N^{-1/2}$ and depending only on $Var[U^D(\boldsymbol{G}_{\max}, \xi)]$ [44].

Theorem 2 combined with Theorem 1 shows that using SAA on top of our equilibrium characterization solves the key difficulties of our problem: we are able to compute an exact min-max strategy for the defender with high probability from a labeled training set without knowledge of P_0, p_a and p_i. It is remarkable that we do not need to estimate P_0 from the training set, this is automatically done within the stochastic approximation procedure. Other stochastic approximation algorithms (e.g., as stochastic gradient descent) could be used but without strong convexity property (which is our case since our function is piecewise linear), they only have convergence guarantees in $N^{-1/2}$.

3.3 Numerical illustration

We performed numerical experiments on different games to illustrate various aspects of our results. In particular, we performed experiments on controlled artificial setups to illustrate the convergence of our training method, the (in)dependence on the number of features, and the form of the equilibrium with multiple attacker types. Due to space constraints, the results are deferred to [32], along with details on the experimental setup for reproducibility (all our code will be made public upon acceptance). We present here the results for a game defined with a real

feature distribution from a credit card fraud dataset [48], to illustrate the form of the equilibrium for simple payoffs.

The dataset [48] contains transactions made by European cardholders in September 2013. A data vector is composed of 31 features: the amount of the transaction (in €) denoted A, the time since the first transaction in the dataset, whether the transaction was malicious (i.e., the label), and 28 anonymized features coming from a PCA. We instantiate our game with this static data set by replacing each attack in the data set by an abstract adaptative attack in our model. For simplicity, we focus only on the amount of the transaction and consider a single attacker type with the following gains: $U^u(v) = A$, $U^d(v) = 0$, and $C_{\text{fa}}(v) = \ell \times A$ for a given $\ell > 0$. This models an attacker that gains the transaction's amount if successful (and the bank loses it), but gains nothing if detected. On the other hand, when a valid transaction is blocked, the bank pays a fraction ℓ of the transaction as false alarm cost. This choice of utility functions is meant to illustrate the equilibrium in a reasonable and simple scenario and not to represent a practical ready-to-implement setting. In the dataset, the fraction of attacks is $p_a = 0.00172$, the maximum transaction is $25,691.16$€ with an average of 88.35€. There are $N = 284,807$ transactions in total.

Figure 1 represents the histogram of valid transaction amounts in $[0, 700]$ (where the majority of transactions occur) and the probability of detection function π_G obtained through our training for different values of ℓ (G_ℓ denotes the parameter trained on the dataset with false alarm cost factor ℓ). When ℓ is small, the defender classifies "aggressively" as fraud by accepting a high false alarm rate. When ℓ increases, the probability of detection functions show that the defender flags as fraud less often. For example, transactions of 700€ are flagged with probability ~ 0.9 by the most aggressive strategy ($\ell = 0.006$) but only with probability ~ 0.1 for the least aggressive strategy ($\ell = 0.074$).

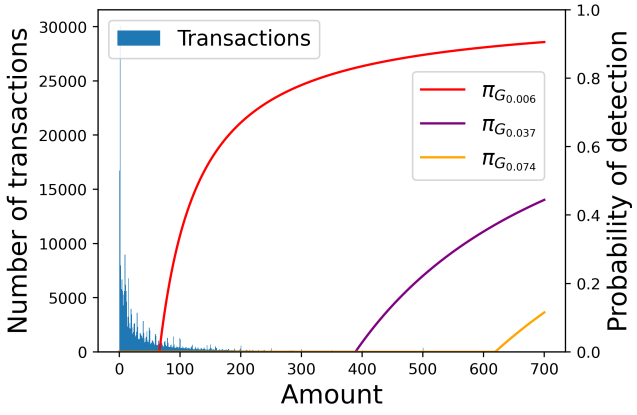

Fig. 1: Empirical distribution of transaction amounts and representation of defender min-max strategies for various l.

The results presented here are computed through our training method in Algorithm 1 and may not be exact. We evaluate the quality of our approximation on games based on artificial distributions (as we only have access to the empirical distribution). The results suggest that the approximation is good even for much smaller training sets as hinted by the theoretical guarantee. Computation times (not exceeding 15min) can be found in [32].

4 Online learning

In the previous section, we showed how the defender can compute an approximate min-max strategy from a training set. Yet, such historical data is not always available. We now show how our low-dimensional characterization of the min-max strategy also allows the defender to learn a good strategy *on-line*, without a priori knowledge of P_0, p_a and p_i, while incurring low loss as captured by the regret.

We consider the following setting. At each time step $t = 1, \dots, T$, the defender chooses a probability of detection function π_t and receives a vector v_t that is classified as an attack with probability $\pi_t(v_t)$. They incur a loss $l(v_t)$ that is $C_{\text{fa}}(v_t)$ in case of false positive and 0 in case of true negative if facing a non-attacker; and $-U_i^d(v_t)$ and $U_i^u(v_t)$ in case of true positive and false negative respectively when facing a type i attacker. We assume that after classification, the defender can observe the type of attack (for convenience, we denote by type $i = 0$ non-attacks) and that they can compute $C_{\text{fa}}(v_t)$ and $U_i^u(v_t), U_i^d(v_t)$ for all i. Finally, as in [10], we assume that attackers act according to best responses to π_t in a Stackelberg fashion, i.e., if the defender faces an attacker of type i at time t we have $v_t \in \arg\max_v \{U_i^u(v)(1 - \pi_t(v)) - U_i^d(v)\pi_t(v)\}$. The defender seeks to minimize the Stackelberg regret:

Definition 5 (Stackelberg regret). *The Stackelberg regret for a sequence of vectors* $(v_1, \dots, v_T)$ *is:* $R(T) = \sum_{t=1}^{T} E_{\pi_t}[l(v_t)] - \min_\pi \sum_{t=1}^{T} E_\pi[l(v_t)]$.

The notion of Stackelberg regret implies that the sequence of vectors depends on the probabilities of detection used. In particular, $\min_\pi \sum_t E_\pi[l(v_t)]$ must be computed using the best response of the attacker to π. It is also key to remember that in our setting, the unknown quantities are P_0, p_a and p_i. The attacker's strategy is assumed to be known as it is best-response to the utilities U_i^u, U_i^d.

It is possible to achieve low regret in T using naively the online gradient descent algorithm of [54]—see [32]—to learn π directly. This gives, however, a bound on the Stackelberg regret of

$$R(T) \leq \frac{D^2\sqrt{T}}{2} + \left(\sqrt{T} - \frac{1}{2}\right) L^2, \tag{6}$$

with $L = \max(\max_v\{C_{\text{fa}}(v)\}, \max_{v,i}\{|U_i^u(v) + U_i^d(v)|\})$ (maximum gradient) and $D^2 = |\mathcal{V}|$ (maximum L_2 distance between two π functions)—see a proof in [32]. This bound is meaningless if the number of features k is large as $|\mathcal{V}| = \Omega(2^k)$. The full strategy π also may not fit into memory.

Building on our characterization of the min-max strategy, we parametrize the defender's strategy by $\boldsymbol{G}$ to propose an alternate learning scheme as Algorithm 2 (where $\Pi_{\mathcal{S}}$ denotes the euclidian projection on a set $\mathcal{S}$).

Algorithm 2 Efficient online gradient descent

Choose $\boldsymbol{G}_1 \in [\underline{G}_1, \overline{G}_1] \times ... \times [\underline{G}_m, \overline{G}_m]$ arbitrarily
for $t = 1, \ldots, T$ **do**
  Predict $\pi_{\boldsymbol{G}_t}$ and receive vector v_t and type i
  if v_t came from a non-attacker **then**
    grad $\in \partial(\pi_{\boldsymbol{G}_t}(v_t) C_{\text{fa}}(v_t))$
  else if v_t came from an attacker of type i **then**
    grad $= e_i$ (i^{th} vector of the canonical base of $\mathbb{R}^m$)
  $\boldsymbol{G}_{t+1} = \Pi_{[\underline{G}_1, \overline{G}_1] \times ... \times [\underline{G}_m, \overline{G}_m]}(\boldsymbol{G}_t - \frac{1}{\sqrt{t}}\text{grad})$

Algorithm 2 exploits the fact that each attacker best responds to the defender's strategy, hence only strategies of the form $\pi_{\boldsymbol{G}}(\cdot)$ are worth using. Thus, instead of learning directly π, the defender learns the parameters $\boldsymbol{G}$. Note that this implies that the defender must be able to evaluate the bounds on the attackers' gain they can impose. Algorithm 2 presents two major advantages: *First*, the defender's strategy is compactly represented with a small number m of parameters, independent of $|\mathcal{V}|$. *Second*, we get a much better regret bound:

Theorem 3. *Algorithm 2 gives Stackelberg regret bound* (6) *with* $L = \max\{1, \max_{v,i}\{\frac{C_{fa}(v)}{U_i^d(v) + U_i^u(v)}\}\}$ *and* $D = ||\overline{\boldsymbol{G}} - \underline{\boldsymbol{G}}||_2$.

Theorem 3 is proved in [32]; the proof leverages our characterization of the min-max strategy with parameters $\boldsymbol{G}$. The result formalizes the intuition that learning $\boldsymbol{G}$ rather than π allows a much smaller regret (D is now independent in $|\mathcal{V}|$). Parameter L^2 now represents the change in false alarm cost one can expect at worst when changing parameters $\boldsymbol{G}$; which is different from L^2 in the naive procedure that corresponded to a gradient wrt π. We performed numerical experiments that illustrate the result of Theorem 3 (in particular the independence in $|\mathcal{V}|$) in [32]. In addition, we observe that $\boldsymbol{G}_t$ converges towards $\boldsymbol{G}_{\max}$.

5 Concluding remarks

We provided a low-dimensional characterization of the min-max strategy in adversarial classification games with general payoffs and showed that this characterization enables efficient training and online learning in practice. Our characterization also allows extending our results to continuous (compact) sets of data $\mathcal{V}$—see the details in [32].

We considered here only strategic attackers. It is possible to extend the model to include non-strategic attackers that follow fixed strategy, through a redefinition of the false alarm cost that preserves the game structure and allows all our results

to be transferred. This can model attacks that are the result of a fixed algorithm. Attacks that are the result of an adaptive algorithm are outside the scope of the current work, but we note that for a wide class of adaptive algorithm this may be modeled in the long run through a utility function.

In the paper, we considered only strategic attackers maximizing their utility. It is possible, however, to extend the model to include non-strategic attackers that follow fixed strategy, through a redefinition of the false alarm cost that preserves the game structure and allows all our results to be transferred. This can be useful for instance to model attacks that are the result of a fixed algorithm. Modeling attacks that are the result of an adaptive algorithm may be more complex and is outside the scope of the current work, but we note that for a wide class of adaptive algorithm this may be well modeled in the long run through a utility function. (e.g. when attacks are the results of a fixed algorithms) in the analysis with minimal changes assuming that the benefits of detecting non-strategic attackers never outweighs the false alarm costs. This allows modeling non-strategic attackers by considering the difference between the previous two quantities as false alarm cost (which is then strictly positive). Non-strategic attackers with shifting strategies are, however, beyond the scope of this paper.

Bibliography

[1] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of EC*, pages 61–78, 2015.

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

[3] Branislav Bošanský, Viliam Lisý, Michal Jakob, and Michal Pěchouček. Computing time-dependent policies for patrolling games with mobile targets. In *Proceedings of AAMAS*, pages 989–996, 2011.

[4] Matthew Brown, Arunesh Sinha, Aaron Schlenker, and Milind Tambe. One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 425–431, 2016.

[5] Michael Brückner, Christian Kanzow, and Tobias Scheffer. Static prediction games for adversarial learning problems. *Journal of Machine Learning Research*, 13:2617–2654, 2012.

[6] Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of ACM SIGKDD*, pages 547–555, 2011.

[7] Samuel Rota Bulò, Battista Biggio, Ignazio Pillai, Marcello Pelillo, and Fabio Roli. Randomized prediction games for adversarial machine learning. *IEEE transactions on neural networks and learning systems*, 28(11):2466–2478, 2016.

[8] Godwin Caruana and Maozhen Li. A survey of emerging approaches to spam filtering. *ACM Computing Surveys*, 44(2):9:1–9:27, 2012.

[9] Lin Chen and Jean Leneutre. A game theoretical framework on intrusion detection in heterogeneous networks. *IEEE Transactions on Information Forensics and Security*, 4(2):165–178, 2009.

[10] Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers. In *Proceedings of NIPS*, 2020.

[11] Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of adversaries. In *Advances in Neural Information Processing Systems (NIPS)*, volume 31, pages 230–241, 2018.

[12] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166. IEEE, 2015.

[13] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of ACM KDD*, pages 99–108, 2004.

[14] Prithviraj Dasgupta, Joseph B Collins, and Michael McCarrick. Improving costs and robustness of machine learning classifiers against adversarial attacks via self play of repeated bayesian games. In *The Thirty-Third International Flairs Conference*, 2020.

[15] Lemonia Dritsoula, Patrick Loiseau, and John Musacchio. A game-theoretic analysis of adversarial classification. *IEEE Transactions on Information Forensics and Security*, 12(12):3094–3109, December 2017.

[16] Fei Fang, Albert Xin Jiang, and Milind Tambe. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *Proceedings of AAMAS*, pages 957–964, 2013.

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

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

[19] Ian Goodfellow, Jon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. arXiv:1412.6572.

[20] Michael Großhans, Christoph Sawade, Michael Brückner, and Tobias Scheffer. Bayesian games for adversarial regression problems. In *Proceedings of ICML*, pages III–55–III–63, 2013.

[21] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of ACM AISec*, pages 43–58, 2011.

[22] Nicole Immorlica, Adam Tauman Kalai, Brendan Lucier, Ankur Moitra, Andrew Postlewaite, and Moshe Tennenholtz. Dueling algorithms. In *Proceedings of STOC*, pages 215–224, 2011.

[23] Murat Kantarcioglu, Bowei Xi, and Chris Clifton. Classifier evaluation and attribute selection against active adversaries. *Data Mining and Knowledge Discovery*, 22(1):291–335, 2011.

[24] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of AAMAS*, pages 689–696, 2009.

[25] Sujin Kim, Raghu Pasupathy, and Shane G Henderson. A guide to sample average approximation. In *Handbook of simulation optimization*, pages 207–243. Springer, 2015.

[26] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *Proceedings of AAAI*, pages 805–810, 2010.

[27] Pavel Laskov and Richard Lippmann. Machine learning in adversarial environments. *Machine Learning*, 81(2):115–119, 2010.

[28] Bo Li and Yevgeniy Vorobeychik. Feature cross-substitution in adversarial classification. In *Proceedings of NIPS*, pages 2087–2095, 2014.

[29] Bo Li and Yevgeniy Vorobeychik. Scalable optimization of randomized operational decisions in adversarial classification settings. In *Proceedings of AISTATS*, 2015.

[30] Jeff Linderoth, Alexander Shapiro, and Stephen Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142(1):215–241, 2006.

[31] Viliam Lisý, Robert Kessl, and Tomáš Pevný. Randomized operating point selection in adversarial classification. In *Proceedings of ECML PKDD*, pages 240–255, 2014.

[32] Patrick Loiseau and Benjamin Roussillon. Scalable optimal classifiers for adversarial settings under uncertainty, 2021.

[33] Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of ACM KDD*, pages 641–647, 2005.

[34] Janusz Marecki, Gerry Tesauro, and Richard Segal. Playing repeated stackelberg games with unknown opponents. In *Proceedings of AAMAS*, pages 821–828, 2012.

[35] 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.

[36] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D. Joseph, Benjamin I. P. Rubinstein, Udam Saini, Charles Sutton, J. D. Tygar, and Kai Xia. Misleading learners: Co-opting your spam filter. In Philip S. Yu and Jeffrey J. P. Tsai, editors, *Machine Learning in Cyber Trust: Security, Privacy, and Reliability*. Springer, 2009.

[37] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. In *Proceedings of IEEE EuroS&P*, April 2018.

[38] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of IEEE S&P*, May 2016.

[39] Juan C. Perdomo and Yaron Singer. Robust attacks against multiple classifiers. *CoRR*, 2019.

[40] Rafael Pinot, Raphael Ettedgui, Geovani Rizk, Yann Chevaleyre, and Jamal Atif. Randomization matters. how to defend against strong adversarial attacks. In *Proceedings of ICML*, 2020.

[41] James Pita, Manish Jain, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Using game theory for los angeles airport security. *AI Magazine*, 30:43–57, 2009.

[42] Aaron Schlenker, Omkar Thakoor, Haifeng Xu, Fei Fang, Milind Tambe, Long Tran-Thanh, Phebe Vayanos, and Yevgeniy Vorobeychik. Deceiving cyber adversaries: A game theoretic approach. In *Proceedings of AAMAS*, pages 892–900, 2018.

[43] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[44] Alexander Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

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

[46] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *Proceedings of USENIX Security*, pages 195–210, 2013.

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

[48] ULB. Credit card fraud detection. `https://www.kaggle.com/mlg-ulb/creditcardfraud/version/3`, 2013.

[49] Bram Verweij, Shabbir Ahmed, Anton J Kleywegt, George Nemhauser, and Alexander Shapiro. The sample average approximation method applied to stochastic routing problems: a computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

[50] Yevgeniy Vorobeychik and Murat Kantarcioglu. *Adversarial Machine Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2018.

[51] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y. Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *Proceedings of USENIX Security*, pages 239–254, 2014.

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

[53] Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Bowei Xi. Adversarial support vector machine learning. In *Proceedings of KDD*, pages 1059–1067, 2012.

[54] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of ICML*, pages 928–936, 2003.