Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Adversarial Machine Learning: Bayesian Perspectives

David Rios Insua, Roi Naveiro, Víctor Gallego & Jason Poulos

To cite this article: David Rios Insua, Roi Naveiro, Víctor Gallego & Jason Poulos (2023): Adversarial Machine Learning: Bayesian Perspectives, Journal of the American Statistical Association, DOI: 10.1080/01621459.2023.2183129

To link to this article: https://doi.org/10.1080/01621459.2023.2183129

Published online: 31 Mar 2023.

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2023, VOL. 00, NO. 0, 1–12
https://doi.org/10.1080/01621459.2023.2183129

Adversarial Machine Learning: Bayesian Perspectives

David Rios Insua*[a], Roi Naveiro*[a,b], Víctor Gallego*[a,c], and Jason Poulos[d]

[a]ICMAT-CSIC, Madrid, Spain; [b]CUNEF University, Madrid, Spain; [c]Komorebi AI Technologies, Madrid, Spain; [d]Harvard Medical School, Boston, MA

ABSTRACT

Adversarial Machine Learning (AML) is emerging as a major field aimed at protecting Machine Learning (ML) systems against security threats: in certain scenarios there may be adversaries that actively manipulate input data to fool learning systems. This creates a new class of security vulnerabilities that ML systems may face, and a new desirable property called adversarial robustness essential to trust operations based on ML outputs. Most work in AML is built upon a game-theoretic modeling of the conflict between a learning system and an adversary, ready to manipulate input data. This assumes that each agent knows their opponent's interests and uncertainty judgments, facilitating inferences based on Nash equilibria. However, such common knowledge assumption is not realistic in the security scenarios typical of AML. After reviewing such game-theoretic approaches, we discuss the benefits that Bayesian perspectives provide when defending ML-based systems. We demonstrate how the Bayesian approach allows us to explicitly model our uncertainty about the opponent's beliefs and interests, relaxing unrealistic assumptions, and providing more robust inferences. We illustrate this approach in supervised learning settings, and identify relevant future research problems. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received May 2020
Accepted February 2023

KEYWORDS
Adversarial risk analysis; Bayesian methods; Cybersecurity; Machine learning; Security

1. Introduction

Over the last decade, an increasing number of processes have been automated through Machine Learning (ML) algorithms, making it more crucial that these algorithms become robust and reliable if we are to trust operations based on their output. State-of-the-art ML methods perform extraordinarily well on standard data, but have been shown to be vulnerable to adversarial examples (Goodfellow, Shlens, and Szegedy 2015), data instances targeted at fooling them. The presence of adversaries has been highlighted in areas such as spam detection (Zeager et al. 2017), computer vision (Goodfellow, Shlens, and Szegedy 2015) and automated driving systems (ADS, Caballero, Ríos Insua, and Banks 2021). In those contexts, algorithms should acknowledge the presence of possible adversaries to protect from their eventual data manipulations. Comiter (2019) provides a review from a policy perspective showing how many AI systems, including content filters, and military and law enforcement systems, are vulnerable to attacks. As a motivating example, consider fraud detection: as ML algorithms are incorporated to such a task, fraudsters learn how to evade them. For instance, they could find out that making a huge transaction increases the probability of being detected, and instead would issue smaller transactions.

As a fundamental assumption, ML systems rely on using iid data for both training and operations (Zhang et al. 2021). However, the security aspects of ML, part of the emerging field of Adversarial Machine Learning (AML), challenge such hypothesis, given the presence of adaptive adversaries ready to intervene to modify the data and obtain a benefit. Stemming from the pioneering work in adversarial classification (Dalvi et al. 2004), the prevailing paradigm in AML has modeled the confrontation between learning-based systems and adversaries through game theory (Menache and Ozdaglar 2011). This entails common knowledge (CK) assumptions (Hargreaves-Heap and Varoufakis 2004), which are questionable in the security domain as adversaries try to hide and conceal information. Thus, there is a need for developing a better founded paradigm: as Fan, Ma, and Zhong (2021) point out, a framework that guarantees robustness of ML against adversarial manipulations in a principled manner is required.

After providing an overview of key concepts and methods in AML emphasizing the underlying game theoretical assumptions, we suggest an alternative formal Bayesian decision theoretical framework based on Adversarial Risk Analysis (ARA, Rios Insua, Rios, and Banks 2009) and illustrate it in supervised learning settings. We end by suggesting a research agenda.

2. Motivating Examples

Two examples serve us to motivate key issues in AML. They showcase how the performance of ML systems may considerably degrade under subtle data manipulations, suggesting the need to take into account the presence of adversaries.

Case 1. Attacking spam detection algorithms. Consider spam detection, an example of content filters which are at the

CONTACT Roi Naveiro roi.naveiro@cunef.edu CUNEF University, Madrid, Spain.
*These authors contributed equally to this work.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

Table 1. Accuracy (with precision) of four algorithms on clean data (untainted); attacked data and unprotected; protected through ARA during operations; protected through ARA during training.

Algorithm	Untainted	Unprotected	ARA op.	ARA tr.
Naive Bayes	0.882 ± 0.004	0.754 ± 0.027	0.939 ± 0.006	—
Logistic Regression	0.932 ± 0.004	0.673 ± 0.005	0.898 ± 0.008	0.946 ± 0.003
Neural Network	0.904 ± 0.029	0.607 ± 0.009	0.882 ± 0.025	0.960 ± 0.002
Random Forest	0.912 ± 0.005	0.731 ± 0.008	0.807 ± 0.007	—

(a) Original image. (b) Attacked image.

Figure 1. Original input and attacked version.

backbone of many security systems. We study the performance degradation of different algorithms under the Good-Words-Insertion attacks described in Naveiro et al. (2019a): the adversary attacks spam E-mails by inserting at most two *good words*[1] into them. Table 1 presents the accuracy of four standard algorithms — *naive Bayes, logistic regression, neural network (NN)* and *random forest* — when facing clean and attacked data,[2] using the Spambase Data Set from the UCI ML repository (Dua and Graff 2017). Accuracy means and standard deviations are estimated via repeated hold-out validation over 10 repetitions (Kim 2009). Observe, columns 2 and 3, the important loss in accuracy of the four algorithms: major performance degradation may affect them when ignoring the possible presence of adversaries. Columns 4 and 5 are discussed in Section 5.

Case 2. Attacking vision algorithms. Computer vision algorithms are at the core of many AI applications such as perception systems in ADS (Caballero, Ríos Insua, and Banks 2021). The simplest and most notorious attacks targeting such algorithms consist of modifications of images so that the alteration becomes irrelevant to the human eye, yet drives a model trained on millions of images to misclassify the attacked ones. This attack entails potentially relevant security consequences. As an example, with a relatively simple convolutional NN (CNN, Goodfellow et al. 2016), we achieve 99% test set accuracy predicting the handwritten digits in the MNIST dataset (LeCun, Cortes, and Burges 1998). However, accuracy reduces to 59% if we attack those data with the *fast gradient sign method* (FGSM, Szegedy et al. 2014). Figure 1 provides examples of an original MNIST image and an attacked one. Our CNN correctly classifies the original image (left) as a 4; however, it misclassifies the attacked one (right) as a 9. FGSM and related attacks described below are easily built through low cost computational methods. In certain settings, they require the attacker to have precise knowledge about the architecture of the corresponding predictive model. This is debatable in most security settings and is a driving force in this article. △

[1] Words that are common in legitimate e-mail but rare in spam.

[2] The logistic regression is applied with L1 regularization, which is equivalent to performing maximum a posteriori estimation in a logistic regression model with a Laplace prior (Park and Casella 2008). The NN has two hidden layers.

3. Adversarial Machine Learning: A Review

We review the main results and concepts in AML. We focus on key ideas in the brief history of the field, motivating a reflection that will lead to our alternative Bayesian framework in Section 4. Further perspectives may be seen in Vorobeychik and Kantarcioglu (2018), Joseph et al. (2019), Biggio and Roli (2018), and Dasgupta and Collins (2019). We describe first the usual workflow in AML within which most previous research can be embedded. In general, we refer to the learning system as *defender* (D, she), and to the adversary manipulating data as *attacker* (A, he).

3.1. An Adversarial Machine Learning workflow

Guaranteeing protection of a learning system against attacks involves undertaking various activities related to security evaluation, threat modeling, and attack simulation. Let us frame these activities within a workflow for AML with three steps, as in Biggio and Roli (2018): (a) gathering intelligence to study the likely attacks that a system may face; (b) forecasting likely attacks; and (c), protecting learning systems from such attacks.

1. Gathering intelligence. This activity is critical to ensure ML security in adversarial environments. Obviously, any algorithm could be fooled if adversarial data modifications are not somehow restricted: in an extreme case, if an instance in a binary classification problem is modified so that it is indistinguishable from an instance of the other class, clearly, the algorithm would misclassify such instance. However, the adversary is probably not interested in making such data modifications as the new instance might lose its malicious purposes. Thus, an in-depth study of likely attackers is key in AML. In general, we should gather information about three attacker features.

First, we assess their *goals*, which may range from appropriating funds to causing harm to people or organizations (Couce-Vieira, Insua, and Kosgodagan 2020). Prior to deploying a ML system, it is crucial to guarantee robustness against attackers with the most common goals. For instance, in fraud detection, the attacker usually obfuscates fraudulent transactions to make the system classify them as legitimate in search of an economic benefit: a fraud detection system should be robust against such attacks, trying to minimize economic losses. In general, attacker goals are classified along two dimensions. For the first one, *violation type*, a usual distinction is between *integrity* (aimed at moving the prediction about particular instances toward the attacker's target, for example, to have malicious samples

misclassified as legitimate); *availability* (aimed at increasing the predictive error to make the system unusable) and *privacy* (exploratory attacks to gather information about the ML system) *violations*. The second dimension refers to *attack specificity*, where the usual distinction is between *targeted attacks*, addressing a few specific defenders, and *indiscriminate attacks*, affecting many defenders in a random manner (Rios Insua et al. 2021).

Second, we assess the *knowledge* that the adversary could have about the ML system. At one end of the spectrum, we find *white box* or *perfect knowledge* attacks: the adversary knows every aspect of the system. This is almost never the case in security scenarios, except perhaps for insider attacks (Joshi, Aliaga, and Insua 2021). Yet, they could be useful in sequential settings where the ML system moves first, training an algorithm to fit its parameters; and the adversary, who moves afterwards, has time to observe the behavior of the system and learn about it.[3] At the other end, *black box* or *zero knowledge* attacks assume that the adversary has capabilities to query the system but does not have any information about the data, feature space, or specific algorithms used. This is the most reasonable assumption when attacking and defending decisions are made simultaneously. In between, attacks are called *gray box* or *limited knowledge*, the most common type of attacks in security settings, especially when attacking and defending decisions are made sequentially but there is private information that agents are not willing to share.

Finally, the third feature refers to the adversary's *capabilities* to influence on data and other features. With *poisoning attacks*, he may obfuscate training data to later induce errors during operations. Alternatively, *evasion attacks* have no influence on training data, but perform modifications during operations, for instance when trying to evade a detection system. These data crafting activities are typical in AML and designated to come from a so-called *data-fiddler*. There could be as well attackers capable of changing the underlying structure of the problem affecting process parameters, called *structural attackers*. Moreover, some adversaries could be making decisions in parallel to those of the defender with the agents' losses depending on both decisions, which we term *parallel attackers*.[4]

2. Forecasting likely attacks. In our path to enhance protection, once having gathered intelligence about the potential attacks to a learning system, we should produce models for how the adversary may behave when facing new data. A central argument for us is that such models must take into account our uncertainty about adversarial aspects. As mentioned, most previous research along these lines has been usually based on game theory assuming full knowledge about the adversary. Thus, given some data, the adversary would behave deterministically: the standard approach forecasts attacks solving constrained optimization problems with different assumptions about the adversary's knowledge, goals and capabilities. The corresponding objective function assesses attack effectiveness, taking into account the assumptions about the attacker features. The constraints frame assumptions such as the adversary wanting to avoid detection or having available a maximum attacking budget. However, full knowledge assumptions are generally unrealistic in the AML realm as adversaries try to conceal information: adversary modeling must take into account the lack of information and corresponding uncertainty that we have about the adversary. Beyond the *aleatoric* and *epistemic* uncertainties typical in risk analysis, in AML, analysts need to consider as well *concept uncertainty* (Banks et al. 2020). Thus, given some data, we associate an *attacking model* with a probability distribution over attacks which encodes our uncertainty about how the adversary will act when seeing a particular instance.[5]

3. Protecting ML algorithms. Once relevant adversarial models have been produced, the last step consists of protecting learning systems against the modeled attacks. Broadly speaking, two types of defenses have been proposed. *Reactive defenses* aim to mitigate, even eliminate, the effects of an eventual attack. They include timely detection of attacks (e.g., Naveiro et al. 2019b); frequent retraining of learning algorithms; or verification of algorithmic decisions by experts. The second type, *proactive defenses*, aim to prevent attack execution. They can entail *security-by-design* approaches such as explicitly accounting for adversarial manipulations (Naveiro et al. 2019a) or producing provably secure algorithms against perturbations (Gowal et al. 2018); or *security-by-obscurity* techniques such as randomization of algorithm responses, or gradient obfuscation to make attacks less likely to succeed (Athalye, Carlini, and Wagner 2018). A more interesting classification of defenses, later emphasized, refers to whether the protection is carried out at *training* or at *operation* time. Defenses of the former class, train learning systems robustly, anticipating future adversarial attacks. Defenses of the latter class, when receiving a potentially attacked instance, undertake inference about possible originating instances to make the corresponding decision.

3.2. Core Concepts in Adversarial Machine Learning

Most work in AML has dealt with supervised learning facing adversarial threats, with a focus on either proposing new attacks to learning systems to showcase their vulnerabilities (thus, related to Steps 1–2 of the above workflow) or proposing defenses to protect algorithms from common attacks (Step 3).

Attacks to learning systems. The most common goal of proposed attacks is to modify instance covariates to induce the learning system into making wrong decisions upon observing or analyzing such contaminated covariates. One of the most influential concepts triggering the current interest in AML are *adversarial examples*. They are introduced within NN models as perturbed data instances aimed at fooling NNs, obtained through solving certain optimization problems (Szegedy et al. 2014). These models are highly sensitive to such examples; recall case 2 in Section 2.

[3] However, although the adversary may have some knowledge, assuming that this knowledge is perfect is not realistic and has been criticized (Dalvi et al. 2004).

[4] Some attackers could combine the three capabilities in certain scenarios. For example, in cybersecurity an attacker might add spam modifying its proportion (structural); alter some spam messages (data-fiddler); and, in addition, undertake his own business decisions (parallel). See Ríos Insua et al. (2018).

[5] Previous attacking models can be easily recovered in this framework assuming degenerate distributions as later illustrated.

Adversarial examples have traditionally targeted computer vision systems, using techniques such as FGSM or *projected gradient descent* (PGD, Madry et al. 2018). These techniques find a constrained perturbation of an image that maximizes the loss function used to train the computer vision system. This optimization is usually approximated using gradient ascent routines. In addition, attacks like these have been extended to target other systems such as natural language processing (Zhang et al. 2020b), due to their increasing relevance. Attacking strategies targeting tabular data are usually application specific. Most approaches model the confrontation between the attacker and the learning system as a game (Brückner, Kanzow, and Scheffer 2012). Assuming that each agent knows their opponent's interests and uncertainty judgments, the adversary will perform the attacks dictated by the Nash equilibrium strategy of that game or emerging as best responses.

Much less AML work is available in relation with unsupervised learning. Kos, Fischer, and Song (2018) describe adversarial attacks to generative models, where slight perturbations to the model input may yield a reconstructed output that is very different from the original input. Biggio et al. (2013) study clustering under adversarial disturbances. The authors describe how to create attacks that significantly alter cluster assignments, as well as obfuscation attacks that slightly perturb inputs to be clustered in a predefined assignment, showing that single-link hierarchical clustering is sensitive to such attacks. Lastly, adversarial attacks targeting time series forecasting systems have started to attract interest. Alfeld, Zhu, and Barford (2016) describes an attacker manipulating the inputs to drive the latent space of a linear autoregressive (AR) model toward a region of interest. Similarly, Papernot et al. (2016) propose adversarial perturbations over recurrent NNs, and Naveiro (2021) studies adversarial attacks against Bayesian dynamic models.

Defenses against attacks. As mentioned in Section 3.1, we distinguish two types of AML defenses: those that promote protection strategies during training, and those that protect during operations. All these defenses assume that a clean training set is available and the attacks happen once the ML system is deployed, the most common case in realistic scenarios.

Protection during operations. The pioneering defense of this type, proposed in Dalvi et al. (2004), was devoted to protect classification systems. Given the importance of classification in many cutting-edge ML applications, this article opens up a research area known as *adversarial classification* (AC). The authors view AC as a game between a classifier (defender), and an adversary (attacker). During operations, upon observing a new vector of covariates, D aims at finding an optimal classification strategy against A's optimal attacks. Computing Nash equilibria in such general games quickly becomes very complex. Thus, the authors propose a forward myopic version: D first assumes that data is untainted, computing her optimal classification decision; then, A deploys his optimal attack against it. Subsequently, D implements her best response against such attack, and so on. They assume CK as all parameters of both players are known to each other. Although standard in game theory, this assumption is unrealistic in security settings typical in AML, an issue acknowledged in Dalvi et al. (2004) and largely unsolved.

Subsequent AC approaches, reviewed in Biggio, Fumera, and Roli (2014), have focused on analyzing attacks over algorithms and upgrading their robustness against them, always making strong assumptions about the adversary. For instance, Lowd and Meek (2005) consider that the attacker can send membership queries to the classifier to issue optimal attacks. Other approaches have focused on improving Dalvi et al.'s model but, as far as we know, none have disposed of the unrealistic CK assumptions.

Protection during training.. An important family of AML defenses try to robustify learning systems by modifying the way training is performed. A relevant source of AML defenses of this type are *Adversarial Prediction Problems* (APPs), whose focus is on building adversarially robust predictive models. It is assumed that during operations an adversary that exercises some control over the data generation process will be present: the data generation distributions at operations and training will be different, jeopardizing standard prediction techniques. To address this, APPs model interactions between the predictor and a fictitious adversary during training as a two-agent game with a system aimed at learning a parametric predictive model and an adversary trying to transform the distribution governing data. The predictor will therefore minimize the expected cost under the operations data distribution. In turn, the fictitious adversary will modify data optimizing his expected cost under this distribution. As such distributions are not known, agents optimize their regularized empirical costs, based on training data: the predictor chooses her model minimizing her expected cost with respect to an attacked version of the training data chosen so as to minimize the adversary's cost. The final optimization problems are case dependent.

In *Stackelberg prediction games*, Brückner and Scheffer (2011) assume full information of the attacker about the predictive model used by the defender who, in addition, has full information about the adversary's costs and action space. D acts first choosing her parameters; then, A, who observes this decision, chooses the optimal data transformation. Finding NE in these games leads to a bi-level optimization problem, minimizing the defender's cost function subject to the adversary, after observing the defender's choice, minimizing his cost function. As nested optimization problems are intrinsically hard, the authors restrict to simple classes where analytical solutions are available.[6]

In *Nash prediction games* both agents act simultaneously. Brückner, Kanzow, and Scheffer (2012) provide conditions for existence and uniqueness of NE in certain subclasses of these games. Notice that APPs propose training using instances that are attacked by "fictitious attackers," and hope that this will serve as a proxy for dealing with real attackers. However, it is assumed that the fictitious attackers' costs and probabilities are CK, which is not realistic in security scenarios. Deviations from the assumed attackers' models potentially lead to severe performance degradation as we later illustrate.

Another important family of defenses that affect the training stage are those that aim at robustifying models against adversarial examples. *Adversarial training* (AT) (Madry et al.

[6]More recently, Naveiro and Insua (2019) provide efficient gradient methods to approximate solutions in more general problems.

2018) is the most important one. It aims at choosing a parametric model (usually a NN) that minimizes the empirical risk evaluated under worst-case data perturbations. Thus, it can be viewed as a zero-sum version of an APP: in AT, the fictitious attacker is assumed to select the data manipulation that maximizes the defender's costs within some constrained region. AT approximates the inner optimization through the PGD algorithm, ensuring that the perturbed input falls within a tolerable boundary, usually specified through some restriction on a norm distance. Attack complexity depends on the chosen norm. However, recent pointers urge modelers to depart from using norm based approaches (Carlini et al. 2019) and develop more realistic attack models as in Brown et al.'s (2017) adversarial patches.

Liu et al. (2018) adapted the idea of AT to Bayesian NNs, using the notion that incorporating randomness in the NN weights enhances their robustness. They propose training Bayesian NNs using mean-field variational inference. However, as in AT, instead of maximizing the evidence lower bound (ELBO) under the original training instances, they propose maximizing the ELBO under a worst-case attacker that chooses the best data manipulation inside a ball in a normed space. As with AT, this heuristic implicitly assumes full knowledge in the construction of the fictitious attacker, not taking into account the existing uncertainty. This may produce performance degradation when dealing with actual, unknown attackers. Finally, another relevant but more heuristic family of defenses is called *adversarial logit pairing* (Kannan, Kurakin, and Goodfellow 2018) in which logits of pairs of attacked and clean instances are encouraged to be the same, thus, yielding the same prediction for both.

All defenses presented assume full knowledge in the attacking models, leading to deterministic attacks. Taking into account existing uncertainties about adversaries would be crucial to produce sensible defenses. This is the goal of the Bayesian framework for AML presented in Section 3.1.

Adversarial Reinforcenment Learning. While there is considerable AML research in supervised and unsupervised learning, much less work is available in relation to reinforcement learning (RL). In it, adversarial aspects refer to the presence of agents whose decisions affect the reward perceived by our supported agent. The prevailing solution approach in standard RL is Q-learning (Sutton and Barto 1998); its adaptation to large problems, deep Q-learning, has faced an incredible growth recently (Silver et al. 2017). It relies on models, such as convolutional NNs, to process input information. Consequently, the adversarial examples described above apply when fooling RL systems (Lin et al. 2017).

Single-agent RL methods fail in presence of other agents that interfere with their learning process, as they do not take into account the nonstationarity due to the other agents' actions: Q-learning may lead to sub-optimal results (Buşoniu, Babuška, and De Schutter 2010). Thus, a deployed RL system must be able to reason about and forecast the adversaries' behavior. Several methods to enhance Q-learning in multiagent systems have been proposed, mostly focusing on adapting ideas from game theory into RL, mainly focusing on modeling the multiagent system through Markov games. Three well-known solutions (Tuyls and Weiss 2012) are minimax-Q learning; Nash-Q learning; and friend-or-foe-Q learning, but these come with unrealistic CK assumptions or can only be applied in restrictive scenarios.

Further comments. Practically all ML methods have been touched upon from an adversarial perspective. Of major importance in this field is the `cleverhans` (Papernot et al. 2018) library, aimed at accelerating research in developing new attack threats and more robust defenses specifically for deep neural models.

AML is a difficult area which evolves rapidly and leads to an arms race in which the community alternates cycles of proposing attacks and implementing defenses that deal with them. Thus, it is important to develop sound techniques. Note that, stemming from Dalvi et al. (2004), most of AML research has been framed, sometimes implicitly, within a standard game theory approach characterized by NE and refinements. However, these entail CK assumptions which are hard to maintain in the security contexts typical of AML. We next propose a Bayesian decision theoretical methodology to solve AML problems, using an ARA perspective (Rios Insua, Rios, and Banks 2009) to model the confrontation between attackers and defenders mitigating questionable CK assumptions.

4. A Bayesian Workflow for AML

We now revisit the workflow in Section 3.1 proposing a Bayesian decision theoretical alternative to AML. It is based on ARA, which operationalizes the Bayesian approach to games (Kadane and Larkey 1982) and facilitates a procedure to forecast adversarial attacks. ARA provides prescriptive support to a decision maker (DM), the ML system in our case, facing one or more attackers whose actions affect her decision making process. The DM is assumed to be a rational, expected utility maximizing agent. Her utility and beliefs (epistemic uncertainty) depends on her decision, the adversaries' rationality and decisions (concept uncertainty), and possibly some other random variables (aleatoric uncertainty). Since CK is not assumed, random variables model adversaries' decisions that must be integrated out to compute expected utilities. ARA provides a coherent procedure to obtain probabilistic forecasts of adversaries' actions. The main idea is to model the adversaries' decision making process, putting priors on unknown quantities to reflect the lack of knowledge. This way, the optimal adversarial decision becomes probabilistic. Simulations from such random optimal decisions are used to compute the DM's expected utility.

Our focus will be on protecting a supervised learning system (D) which receives instances described by covariates $x \in \mathbb{R}^d$, with each instance having an associated output y. Uncertainty about the instances' output given its covariates is modeled through a distribution $p(y|x)$. This distribution can arise from a generative model, where distributions $p(x)$ and $p(x|y)$ are modeled explicitly and $p(y|x)$ is obtained via Bayes formula; or from a discriminative model, in which $p(y|x)$ is modeled directly (Bishop 2006). It may be derived through maximum likelihood or in a Bayesian way using training data which, by assumption, is free of attacks. Whichever estimation method is adopted, upon observing a new instance with covariates x, the Defender must decide the corresponding output. As D is rational, she decides

based on maximum predictive utility through

$$\arg\max_{y_D} \int u(y_D, y)p(y|x)\mathrm{d}y,$$

where $u(y_D, y)$ is the utility that she perceives when an instance whose actual output is y is assigned output y_D.[7] In adversarial settings, agent A applies an attack a to the features x leading to the transformation $x' = a(x)$, the observation actually received by D. We focus on exploratory attacks, which affect just over operational data, leaving training data untainted. Let us revisit the three stages of the workflow.

1. Gathering intelligence. This stage entails modeling the attacker's problem. Assessing his goals requires determining the actions that he may undertake and the utility that he perceives when performing a specific action, given a defender's strategy: the output is the set of attacker's decisions and a functional form for his utility, generally dependent on his and the defender's decisions. Assessing the attacker knowledge entails looking for information that he may have when performing the attack, and his degree of knowledge about it, as we do not assume CK. This requires not only a modeling activity, but also a *security assessment* of the ML system determining which of its elements (training data, feature space, architecture, loss function, parameters, etc.) are accessible to the attacker. Finally, identifying his capabilities requires determining which part of the defender problem the attacker has influence on.

Consider an adversary aiming at fooling a supervised learning system by modifying the value of the covariates. The adversary receives objects with covariates x and output y and manipulates x, transforming them into x'. His goal is to induce the defender to make nonoptimal decisions for the output corresponding to the observed covariates. Following a normative decision theoretical perspective, we model the adversary as a rational agent choosing data manipulations to maximize expected utility. Let $u_A(y_D, y)$ be the adversary's utility when the defender assigns output y_D to an instance whose actual output is y. This utility can also depend on the specific data manipulation, as distinct manipulations can entail different costs; however, to simplify notation we do not include explicitly this dependence. The adversary thus chooses data manipulations through

$$x'(x, y) = \arg\max_z \int u_A(y_D, y)p_A(y_D|z = a(x))\mathrm{d}y_D, \qquad (1)$$

where $p_A(y_D|z = a(x))$ models the adversary's belief about the defender's decision upon observing the manipulated instance $z = a(x)$.

2. Forecasting likely attacks. Based on Step 1, we produce models for how the adversary would modify data, encoding not only the information gathered, but also our uncertainty about the adversary's elements. The output of this stage is an *attacking model*, a probability distribution over adversarial manipulations that encodes all relevant uncertainties. A formal Bayesian way to do this, as suggested by ARA, is to place priors on every unknown element of the adversary's decision making problem. The uncertainty implied by these priors is propagated to the optimal adversarial data modification that becomes random. Its associated probability distribution conforms to the attacking model used to protect the learning algorithm. In general, evaluating analytically such model will be unfeasible. However, in most cases it is conceptually and computationally simple to sample from it. This just entails sampling from our priors and, for each sample, solving the adversary's decision making problem, which provides a sample from the random optimal data manipulation. In our adversarial supervised learning context, the adversary will produce data manipulations solving (1). Under standard CK assumptions, our *attacking model* would be $p(x'|x) = \delta\left(x' - \arg\max_z \int u_A(y_D, y)p_A(y_D|z)\mathrm{d}y_D\right)$. However, as argued, CK rarely holds in security domains there being multiple sources of uncertainty. First of all, unlike the adversary, we do not know the actual output y for a given instance x. Thus, our *attacking model* must account for this uncertainty through $p(x'|x) = \int p(x'|x, y)p(y|x)\mathrm{d}y$. Sampling from $p(y|x)$ is standard. Sampling from $p(x'|x, y)$ is more complex, as we usually have uncertainty about the adversary's utility $u_A(y_D, y)$ and his probability estimates $p_A(y_D|z)$. We propose modeling such uncertainty with, respectively, random utilities U_A and random probabilities $P_A^{y_D}$ defined, without loss of generality, over an appropriate common probability space $(\Omega, \mathcal{A}, \mathcal{P})$ with atomic elements $\omega \in \Omega$. This induces a distribution over the Attacker's optimal attack defined through

$$X'_\omega(x, y) = \arg\max_z \int U_A^\omega(y_D, y)P_A^\omega(y_D|z)\mathrm{d}y_D,$$

leading to $p(x'|x, y) = \mathcal{P}(X'_\omega(x, y) = x')$. In such a way, our model for $p(x'|x, y)$ properly accounts for the existing uncertainty about the adversary. By construction, if we sample utilities and probabilities from their corresponding priors and solve (1), this solution would be distributed according to $p(x'|x, y)$. Overall, to produce samples from $p(x'|x)$, we first sample from y from the posterior predictive distribution $p(y|x)$, and then x' from $p(x'|x, y)$.

The specifications of random utilities and random probabilities are application-specific. Guidelines for adversarial classification are given in Gallego et al. (2020). Notice that, to model our uncertainty about the adversary, we study his decision making problem from our point of view. Obviously, when analyzing the Attacker's problem, we must take into account his uncertainty about our elements; for example, his uncertainty about the defender's decision upon observing the manipulated instance. This could lead to a infinite hierarchy of decision making problems as presented in Rios and Insua (2012), albeit in a simpler context. One would typically model several steps in the hierarchy and stop at a level in which no more information is available. At that stage, noninformative priors over the involved probabilities and utilities can be used.

To sum up, we have now a general, formal, decision-theoretic alternative to produce attacking models, that is, samples from $p(x'|x)$, keeping CK assumptions at a minimum. Note, however, that previous attacks proposed in the literature can be adopted within the proposed workflow. For instance, consider the FGSM attack (case 2, Section 2): it assumes that the defender uses a parameterized model with parameters θ, trained minimizing a loss function $L(\theta, x, y)$; the attacker has full knowledge about such loss, or at least its gradient and has resources to

[7] If the output is discrete rather than continuous (as in classification problems) the integral is replaced by a sum.

perturb the covariates by adding a small vector ϵ. Under this CK setting, the proposed (deterministic) attack is $x' = x + \epsilon \cdot \text{sign}\left[\nabla_x L(\theta, x, y)\right]$, leading to an attacking model $p(x'|x, y, \theta)$, degenerated at such x'.

Our proposed attacking model would be classified as gray box since, even if perfect knowledge is not assumed, certain assumptions about the adversary are being made such as him being an expected utility maximizer. One of the advantages of the proposed approach is its ability to create more complex attacking models; for example, through mixtures of attackers with different solution concepts (Rios Insua, Banks, and Rios 2016). Finally, for purely black-box settings in which the attacker is only assumed to have query access to the target learning system, an interesting approach to produce attacking models has been recently introduced (Lee et al. 2022). Here, the authors propose to generate attacks using Bayesian optimization, modeling the (unknown) attacker's objective function with a Gaussian process that is sequentially updated after the queries' results are received.

3. Protecting ML algorithms. Once with a reasonable probabilistic *attacking model*, we protect our learning system against such attacks, either at operations or training.

Protection during operations. The Defender observes a potentially attacked vector of covariates x'. Based on it, she has to assign an output y_D. An adversary-unaware D will make this decision by maximizing the posterior predictive utility, $\arg\max_{y_D} \int u(y_D, y)p(y|x')\mathrm{d}y$. In terms of the spam detection example (Section 2), x' would represent the words used in an E-mail, potentially manipulated by a spammer, and y_D corresponds to labeling such E-mail as spam or legitimate. As illustrated, solving this classification using an adversary-unaware D could lead to serious performance degradation.

Upon observing x', D is uncertain about the actual originating vector of covariates x (the words of the originating E-mail). She may model this uncertainty through a distribution $p(x|x')$ and decide based on the posterior predictive utility, where x has been marginalized out

$$\arg\max_{y_D} \int u(y_D, y) \left[\int p(y|x)p(x|x')\mathrm{d}x \right] \mathrm{d}y, \tag{2}$$

where conditional independence of y and x' given x is assumed. Thus, when adversaries are present, rather than deciding based on the posterior predictive distribution of a new instance, we do it based on what we designate the *robust adversarial posterior predictive distribution* (RAPPD) $\int p(y|x)p(x|x')\mathrm{d}x$. This is generally not available in closed form and has to be evaluated numerically using Monte Carlo methods. The key step for this is the ability to sample from $p(x|x')$, that is, the distribution of possible originating covariates given the observed ones x'. Here is where the attacker models from Step 2 come into play. Having constructed a model for $p(x'|x)$ and being able to sample from it, all we need is to generate samples from the inverse distribution $p(x|x')$. Techniques for this based on Approximate Bayesian Computation (ABC) are discussed in Gallego et al. (2020). In terms of the spam detection case, the attack model $p(x'|x)$ would reflect D's uncertainty about the manipulated words x' that the adversary selects, given the E-mail with words x.

Protection during training. As mentioned, other defenses modify how training is performed in order to take into account the possible presence of an adversary during operations. Their goal is to train using artificial data that somehow mimic actual, potentially attacked, operational data through several heuristics. Most of them model how the attacker would modify the instances in the training set. Having trained the classifier in this manner, $p(y|x')$ could be directly evaluated at the operation stage as this probability has been inferred taking into account the presence of an attacker. As discussed, these methods assume models for how the attacker would modify training instances that do not take into account the existing uncertainty. For instance, AT, as a proxy to robustify classifiers against attacks, considers an attacker that produces the worst data modification for the classifier, assuming explicitly that the classifier has knowledge about the attacker's objectives, and implicitly that he has knowledge about the classifier's utility. However, in realistic settings, we would not have precise information about how the attacker modifies a given instance, as we do not know, in general, his intentions and probability assessments. Thus, assuming a deterministic attack for robustification purposes may result inappropriate. We believe that it is crucial to account for such uncertainty explicitly.

Ye and Zhu (2018) take a step in this direction. They provide a Bayesian counterpart of AT, designated Bayesian Adversarial Learning, assuming that the Defender has observed clean training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ which are samples from an unknown distribution. Based on this, an adversary unaware defender using a model parameterized by θ will simply compute the posterior $p(\theta|\mathcal{D})$ and employ this to calculate the predictive distribution used during operations. However, the presence of an adversary at operations changes the data-generation mechanism, Thus, using the original $\mathcal{D}$ for inference could lead to large performance degradation. Instead, the authors suggest computing a *robust adversarial posterior distribution* (RAPD) over the parameters $\int p(\theta|\widetilde{\mathcal{D}})p(\widetilde{\mathcal{D}}|\mathcal{D})\mathrm{d}\widetilde{\mathcal{D}}$, where $\widetilde{\mathcal{D}}$ refers to the manipulated training data. Gibbs sampling provides samples from it iterating through

$$\widetilde{\mathcal{D}}^{(t)}|\theta^{(t-1)}, \mathcal{D} \sim p(\widetilde{\mathcal{D}}|\theta^{(t-1)}, \mathcal{D}), \tag{3}$$

$$\theta^{(t)}|\widetilde{\mathcal{D}}^{(t)} \sim p(\theta|\widetilde{\mathcal{D}}^{(t)}). \tag{4}$$

After a burn-in period, samples $\{\theta^{(T)}, \widetilde{\mathcal{D}}^{(T)}\}$ follow the joint posterior $p(\theta, \widetilde{\mathcal{D}}|\mathcal{D})$ and, consequently, sample $\theta^{(T)}$ follows the RAPD. The distribution $p(\widetilde{\mathcal{D}}|\theta, \mathcal{D})$ quantifies our uncertainty about the data generation process; that is, about how the adversary will modify data $\mathcal{D}$.

As with attacks, if we assume a high degree of CK, earlier defense mechanisms can be framed within our workflow. In AT, the defender uses a parametric model with parameters θ. An adversary unaware D makes inference about θ minimizing a loss function $\sum_{i=1}^N L(\theta, x_i, y_i)$, with N training points. When taking into account the adversary, AT proposes minimizing $\sum_{i=1}^N \max_{\|\gamma\| \leq \epsilon} L(\theta, x_i + \gamma, y_i)$; that is, minimize the loss evaluated under worst-case perturbations in some constrained region.

Most common losses can be written as negative log posterior distributions. Thus, for the rest of the discussion, assume that

the loss function can be written as

$$\sum_{i=1}^{N} L(\theta, x_i, y_i) = -\sum_{i=1}^{N} \log p(x_i, y_i|\theta) - \log p(\theta). \quad (5)$$

If we assume that, for some fixed value of θ, $p(\widetilde{\mathcal{D}}|\mathcal{D})$ has the form

$$p(\widetilde{\mathcal{D}}|\mathcal{D}) = \prod_{i=1}^{N} p(\widetilde{x}_i, y_i|x_i, y_i)$$
$$= \prod_{i=1}^{N} \delta\left(\widetilde{x}_i - \left[x_i + \arg\max_{\|\gamma\|\leq\epsilon} L(\theta, x_i + \gamma, y_i)\right]\right),$$

and we recover AT as a maximum a posteriori (MAP) estimate of θ under the robust adversarial posterior distribution. Indeed, notice that the robust posterior can be written

$$\int p(\theta|\widetilde{\mathcal{D}}) p(\widetilde{\mathcal{D}}|\mathcal{D}) \mathrm{d}\widetilde{\mathcal{D}} = p(\theta|\{(x_i^*, y_i)\}_{i=1}^N),$$

where $x_i^* = x_i + \arg\max_{\|\gamma\|\leq\epsilon} L(\theta, x_i+\gamma, y_i)$. The MAP estimate of θ is

$$\theta^{MAP} = \arg\max_{\theta} \left[\log p(\theta|\{(x_i^*, y_i)\}_{i=1}^N)\right]$$
$$= \arg\min_{\theta} \left[-\log \sum_{i=1}^{N} p(x_i^*, y_i|\theta) - \log p(\theta)\right],$$

which, according to (5) is nothing else but the loss evaluated under the worst-case transformation x_i^*. Thus, AT is a special case of our workflow, in which we assume CK in the sense that the attacking model $p(\widetilde{\mathcal{D}}|\mathcal{D})$ is deterministic; that is, a degenerate distribution.

There are several ways of sampling from the conditionals (3) and (4). Rios Insua, Naveiro, and Gallego (2020) propose a scalable way of doing it leveraging efficient SG-MCMC sampling algorithms, and, in particular, stochastic gradient Langevin dynamics (SGLD, Welling and Teh 2011). First, to account for the uncertainty that the defender has over the attacking model, the authors propose defining $p(\widetilde{D}|D, \theta) \propto \exp\{L(\theta, x_i, y_i)\}$. Under SGLD, sampling iterations adopt the form

$$x_{i,t+1} = x_{i,t} - \epsilon \nabla_x (\log p(y|x_{i,t}, \theta) + \log p(\theta)) + \xi_t, \quad (6)$$

with $\xi_t \sim \mathcal{N}(0, 2\epsilon)$, and $t = 1, \ldots, T$ with $x_{i,1} = x_i$. Note that this is a sampler from the distribution $p(\widetilde{D}|D, \theta)$ and we approximate this distribution with the set of attacked samples $\{x_i^*\}_{i=1}^N$, setting $x_i^* = x_{i,T}$. Further uncertainties can also be accounted for; let us denote with λ the vector of hyperparameters of the optimizer, such as the step sizes ϵ_t or the number T of iterations. Then, we have $p(\widetilde{D}|D, \theta) = \int p(\widetilde{D}|D, \theta, \lambda) p(\lambda) d\lambda$. To generate a perturbed data sample from the previous distribution, we need to sample from $p(\lambda)$. For instance, we could sample the step sizes ϵ_t from a beta distribution over $[1e-5, 1e-3]$, which are typical values in computer vision tasks using deep NNs; the number of iterations T could be sampled from a Poisson distribution. Moreover, we could consider mixtures of different attackers, for instance by sampling a Bernoulli random variable and then choosing the gradient corresponding to either FGSM or another attack, such as Carlini and Wagner's (2017).

The attacker might have also uncertainty over the model the defender adopts, let it be the concrete model architecture or its parameters' values. Accounting for this would entail that the previous sampling scheme is done over an uncertain defender model $p(y|x, \theta)$ and start a hierarchy of level-k thinking (Rios and Insua 2012), which can be computationally intractable. Instead, we propose mixing both steps and sample from the posterior distribution of the defended model rather than just arriving at θ^{MAP}. To do so efficiently, if the model is optimized using gradient descent routines (as usual with deep NN models), we can again leverage SG-MCMC techniques to sample from the robust posterior $p(\theta|\{(x_i^*, y_i)\}_{i=1}^N)$, by repeating the following procedure for some number of training iterations:

1. Sample perturbed samples $x_1, \ldots, x_K \sim p(\widetilde{D}|D, \theta)$ using the sampler from (6), or from the natural distribution, for a mini-batch of size K.
2. Update $\theta_{t+1} = \theta_t - \epsilon \nabla \sum_{i=1}^{K} L(\theta_t, x_i, y_i) + \mathcal{N}(0, 2\epsilon I)$

In the end, we collect S samples $\{\theta_i\}_{i=1}^S$ from the robust posterior distribution. Then, given an instance x, we compute the predicted output y_D approximating the posterior predictive utility using MC.

5. Case Studies

We illustrate the proposed defense mechanisms through the motivating examples from Section 2.[8]

Case 1. Spam detection. Consider the set-up from Section 2.

Protection during operation. For the first batch of experiments, we use the same algorithms in Section 2. Recall the severe performance degradation resulting from Good-Words-Insertion attacks (Table 1, cols. 2 and 3). Once the models are trained, we perform attacks over the instances in the test set, solving problem (1) for each test spam E-mail, assuming certain values for the attacker's utilities and probability judgements. D is uncertain about the attacker's elements and models these uncertainties with random utilities and probabilities: we use beta distributions centered at the attacker's actual utility and probability values with variances chosen to guarantee that the distribution is concave in its support (they must be bounded from above by $\min\{[\mu^2(1-\mu)]/(1+\mu), [\mu(1-\mu)^2]/(2-\mu)\}$, where μ is the corresponding mean). The variance size informs about the degree of knowledge the defender is assumed to have about the attacker; reflecting a moderate lack of knowledge, we set the variance to be 10% of this upper bound. Of course, we are assuming certain degree of knowledge about the adversary, as the expected values of the random utilities and probabilities coincide with the actual values used by the attacker. We later study how deviations from the assumed attacker behavior affect performance.

Having a model for the attacker, for each instance x' of the test set, the defender computes the robust adversarial posterior predictive distribution $\int p(y|x) p(x|x') \mathrm{d}x$, and assigns x' to the class maximizing the posterior predictive utility (2). To compute the RAPPD, samples from $p(x'|x)$ are obtained leveraging the ability to sample from the attacker model and using ABC as in Gallego et al. (2020).

[8]Code to reproduce these experiments is available at *https://github.com/roinaveiro/aml_bayes*.

Table 2. Average accuracy plus minus one standard deviation of four algorithms on attacked data without defense (col 2); with CK defense (col 3); and with ARA defense (col 4).

Classifier	Acc. Taint.	Acc. CK Taint.	Acc. ARA Taint.
Naive Bayes	0.793 ± 0.005	0.867 ± 0.004	0.883 ± 0.005
Logistic Reg.	0.687 ± 0.008	0.803 ± 0.007	0.864 ± 0.005
Neural Network	0.774 ± 0.007	0.767 ± 0.007	0.792 ± 0.006
Random Forest	0.682 ± 0.005	0.819 ± 0.007	0.821 ± 0.007

Column 4 in Table 1 compares the average accuracy of the robustified during operation classifiers on tainted data. As can be seen, our approach allows us to reduce the performance degradation of the four original classifiers, showcasing the benefits of explicitly modeling the attacker's behavior in adversarial environments. Interestingly, in the naïve Bayes case, our approach even outperforms the algorithm behavior under untainted data (column 2). This effect has been observed also in Naveiro et al. (2019a) and Goodfellow, Shlens, and Szegedy (2015) for other algorithms and application areas. This is likely due to the fact that the presence of an adversary has a regularizing effect, being able to improve the original accuracy of the base algorithm, and making it more robust.

The previous experiment used beta distributions centered around the values actually employed by the attacker to quantify the uncertainty about the attacker's utility and probability. It is natural to explore how deviations from the assumed values affect performance. Our second batch of experiments tests the approach against an attacker whose utilities and probabilities are different from those assumed by the defender. In particular, for each attack, A deviates uniformly around the assumed probability and utility. The size of the deviation is constrained to be less than 50% the assumed value: if we center our beta distribution for, for example, the attacker's probability at value μ, the attacker will deviate from the assumed behavior in the range $(0.5 \cdot \mu, 1.5 \cdot \mu)$. Thus, in this experiment, our beta distributions will be centered around *wrong values*. We set the variance of the beta priors to be relatively high, at 50% of the upper bound, and compare our approach with the CK one, in which the elements of the attacker are assumed to be known, and thus are point masses (on wrong values).

Table 2 shows average accuracy plus minus one standard deviation (estimated through repeated hold out validation) of the four algorithms on attacked data without defense (col. 2), the standard CK defense (col. 3) and, finally, our ARA defense (col. 4). Note first the overall performance drop with respect to the results in Table 1 col. 4: when the attacker deviates from his assumed behavior, the performance of both ARA-based and CK defenses is lower. However, we can also observe that the ARA-based defense outperforms the CK defense for all classifiers: when the attacker deviates from the assumed behavior, accounting for the uncertainty over his elements is beneficial. This experiment showcases the increase in robustness due to modeling uncertainty in scenarios in which CK is not realistic.

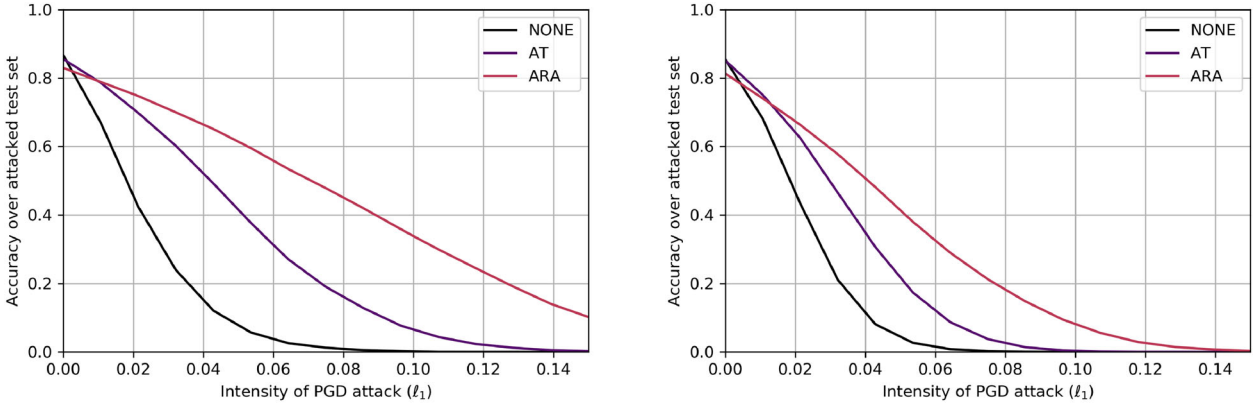
Protection during training. We next assess ARA based robustification during training. This requires the underlying model to be differentiable in the parameters, thus, leaving just two candidates among the original models: logistic regression and NN. Both models can be trained using SGD plus noise methods to obtain uncertainty estimates from the posterior. Next, we attack the clean test set using the procedure in Section 2 and evaluate the performance of our robustification proposal. Since we are dealing with discrete attacks, we cannot use the uncertainty over attacks as in (6). Instead, we model it using the distribution $p(x'|x)$ and take samples from it as discussed in step 3 from our Section 4 workflow. We evaluate the Bayesian predictive distribution using $S = 5$ posterior samples obtained after $T = 2000$ SGLD iterations, and present the results in Table 1, col. 5. Observe again that the proposed robustification process protects differentiable classifiers, recovering from the degraded performance under attacked data. Note that the robustified algorithms achieve even higher accuracies than those attained by the original classifier over clean data, due to the regularizing effect mentioned above.

Case 2. Vision. When the input data is high-dimensional (such as with images), our ARA robustification at operation easily becomes computational intractable. We thus robustify model at training. For illustration purposes, we perform additional experiments using two benchmarks: Fashion-MNIST, a clothing classification problem (Xiao, Rasul, and Vollgraf 2017), and Kuzushiji-MNIST, a traditional Japanese handwritten character recognition problem (Clanuwat et al. 2018). For both datasets we trained standard deep NNs over their respective training sets (consisting of 60.000 images each) using SGD (i.e., no defense), adversarial training (AT defense), and our robustification procedure from Section 4 (ARA defense). We then attacked the respective test sets using five iterations of PGD, with varying attack intensities (the step-size ϵ in (6)), and evaluated the accuracies of both models under these attacked test sets. Figure 2 displays these results. Note that our scalable approach from Section 4 offers fairly superior robustification defenses compared to the AT defense mechanism, showing that incorporating the uncertainties provided by the ARA methodology has additional benefits when adversarially training a ML model in diverse datasets.

6. Conclusion

We have provided a review of key approaches, models, and concepts in AML. This area is of major importance in security and cybersecurity to protect systems that increasingly rely on ML algorithms (Comiter 2019; Ciancaglini et al. 2020). The pioneering work by Dalvi et al. (2004) framed most of this research within the game theory realm, with entailed CK conditions which hardly hold in AML security contexts.

We have proposed a Bayesian alternative to AML. Its main difference with respect to previous approaches is that unrealistic CK conditions are not entailed. As a consequence of the increased realism, the resulting models are more robust, especially with respect to deviations in the assumed attacker behavior, given the better reflection of the involved uncertainties as empirically illustrated. Our Bayesian framework enjoys greater flexibility than previous game-theoretic approaches, some of which can be framed as limit or degenerate cases of our proposal. However, the increased robustness and flexibility comes at a higher computational and modeling costs. Investigating how to



(a) Fashion-MNIST dataset. (b) Kuzushiji-MNIST dataset.

Figure 2. Robustness of a deep network against the PGD attack under three defense mechanisms (NONE, AT, ARA). (a) depicts the security evaluation curves for the attacked Fashion-M. dataset. (b) depicts the respective curves for the attacked Kuzushiji-M. dataset.

alleviate these costs is an open research question. Related to this, we sketch several promising avenues for future work.

A promising research line consists of developing efficient algorithms for approximate Bayesian inference with robustness guarantees. For example, regarding opponent modeling in sequential decision making, an agent has uncertainty over her opponent type initially; as more information is gathered, she might reduce her uncertainty via Bayesian updating. Similarly, work in robust Bayesian analysis (Rios Insua and Ruggeri 2000), in particular referring to likelihood robustness, is relevant. Not taking into account an attacked data generation process is an example of model misspecification; robustness of Bayesian inference to such issue has been revisited recently in Miller and Dunson (2019).

There are also several enhancements aimed at improving operational aspects of the framework. For example, we discussed only problems with two agents. It would be relevant to deal with multiple agents, including cases in which agents on attack or defense attempt to cooperate. There is also potential in new algorithmic approaches. Exploring gradient-based techniques for bi-level optimization problems arising in AML is a fruitful line (Naveiro and Insua 2019). More efficient MCMC samplers can be adopted in the proposed workflow (Gallego and Insua 2018). Recall that our framework essentially goes through simulating from the attacker problem to forecast attacks and then optimizing for the defender to find her optimal decision. This may be computationally demanding and we could explore single stage approaches, such as augmented probability simulation (Ekin et al. 2022).

As mentioned in Section 3.2, adversarial versions of various ML problems have been studied. However, further research is required in unsupervised learning, including clustering methods, dynamic linear models (Naveiro 2021), natural language processing models (Wang et al. 2019), and in RL, including policy gradient (Lin et al. 2017) and extensions to semi-Markov Decision Processes (Du, Futoma, and Doshi-Velez 2020).

Applications, such as those presented in Comiter (2019) and Ciancaglini et al. (2020), are abound. We mention four of direct interest to us: (i) the development of defenses against fake news; (ii) the development of robust ADS algorithms (Caballero, Ríos Insua, and Banks 2021); (iii) the use of AML for improving counterfactual inference in observational studies (Johansson, Shalit, and Sontag 2016); and (iv) leveraging ideas from causal inference to improve adversarial robustness (Schölkopf et al. 2021). Several recent works, for instance, aim to improve the robustness of deep NNs for image classification by leveraging a causally informed model of unseen perturbations or adversarial examples (Zhang, Zhang, and Li 2020a; Zhang et al. 2021).

Supplementary Materials

Python code to reproduce the results presented in this article is available at *https://github.com/roinaveiro/aml_bayes*. Reproducibility workflow is included in the repository's readme file.

Funding

The authors acknowledge the support of the National Science Foundation under Grant DMS-1638521 to the Statistical and Applied Mathematical Science Institute (SAMSI), NC, USA. RN acknowledges the support of CUNEF University. DRI is supported by the AXA-ICMAT Chair and the Spanish Ministry of Science program PID2021-124662OB-I00. VG acknowledges support from grant FPU16-05034. This work is supported by the Severo Ochoa Excellence Programme CEX-2019-000904-S, the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101021797 (Starlight) and 815003 (Trustonomy), the US NSF grant DMS-1638521 and a grants from the FBBVA (Amalfi), EOARD (FA8655-21-1-7042) and AFOSR (FA-9550-21- 1-0239).

ORCID

Roi Naveiro https://orcid.org/0000-0001-9032-2465

References

Alfeld, S., Zhu, X., and Barford, P. (2016), "Data Poisoning Attacks Against Autoregressive Models," in *Proceedings of the 30th AAAI Conference Artificial Intelligence*, pp. 1452–1458. [4]

Athalye, A., Carlini, N., and Wagner, D. (2018), "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples," in *International Conference on Machine Learning*, pp. 274–283. [3]

Banks, D., Gallego, V., Naveiro, R., and Ríos Insua, D. (2020), "Adversarial Risk Analysis: An Overview," *Wiley Interdisciplinary Reviews: Computational Statistics*, 14, e1530. [3]

Biggio, B., Fumera, G., and Roli, F. (2014), "Security Evaluation of Pattern Classifiers Under Attack," *IEEE Transactions on Knowledge and Data Engineering*, 26, 984–996. [4]

Biggio, B., Pillai, I., Rota Bulò, S., Ariu, D., Pelillo, M., and Roli, F. (2013), "Is Data Clustering in Adversarial Settings Secure?" in *Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security*, pp. 87–98, ACM. [4]

Biggio, B., and Roli, F. (2018), "Wild Patterns: Ten Years after the Rise of Adversarial Machine Learning," *Pattern Recognition*, 84, 317–331. [2]

Bishop, C. (2006), *Pattern Recognition and Machine Learning*, New York: Springer. [5]

Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2017), "Adversarial Patch," arXiv:1712.09665. [5]

Brückner, M., Kanzow, C., and Scheffer, T. (2012), "Static Prediction Games for Adversarial Learning Problems," *Journal of Machine Learning Research*, 13, 2617–2654. [4]

Brückner, M., and Scheffer, T. (2011), "Stackelberg Games for Adversarial Prediction Problems," in *Proceedings of the 17th ACM SIGKDD International Conference*, pp. 547–555. [4]

Buşoniu, L., Babuška, R., and De Schutter, B. (2010), "Multi-Agent Reinforcement Learning: An Overview," in *Innovations in Multi-Agent Systems and Applications - 1*, eds. D. Srinivasan and L. C. Jain, pp. 183–221, Berlin, Heidelberg: Springer-Verlag. [5]

Caballero, W. N., Ríos Insua, D., and Banks, D. (2021), "Decision Support Issues in Automated Driving Systems," *International Transactions in Operational Research*, 30, 1216–1244. [1,2,10]

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., and Madry, A. (2019), "On Evaluating Adversarial Robustness," arXiv:1902.06705. [5]

Carlini, N., and Wagner, D. (2017), "Towards Evaluating the Robustness of Neural Networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. [8]

Ciancaglini, C. G., Sancho, D., McCarthy, O., Eira, M., Amann, P., Klayn, A., McArdle, R., Beridze, I., and Amann, P. (2020), "Malicious uses and Abuses of Artificial Intelligence," *Trend Micro Research*, Available at *https://documents.trendmicro.com/assets/white_papers/wp-malicious-uses-and-abuses-of-artificial-intelligence.pdf*. [9,10]

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. (2018), "Deep Learning for Classical Japanese Literature," arXiv:1812.01718. [9]

Comiter, M. (2019), "Attacking Artificial Intelligence," Available at *https://www.belfercenter.org/sites/default/files/2019-08/AttackingAI/AttackingAI.pdf*. [1,9,10]

Couce-Vieira, A., Insua, D. R., and Kosgodagan, A. (2020), "Assessing and Forecasting Cybersecurity Impacts," *Decision Analysis*, 17, 356–374. [2]

Dalvi, N., Domingos, P., Mausam, Sumit, S., and Verma, D. (2004), "Adversarial Classification," in *Proceedings of the 10th ACM SIGKDD International Conference*, KDD '04, pp. 99–108. [1,3,4,5,9]

Dasgupta, P., and Collins, J. B. (2019), "A Survey of Game Theoretic Approaches for Adversarial Machine Learning in Cybersecurity Tasks," *AI Magazine*, 40, 31–43. [2]

Du, J., Futoma, J., and Doshi-Velez, F. (2020), "Model-based Reinforcement Learning for Semi-Markov Decision Processes with Neural ODEs," arXiv:2006.16210. [10]

Dua, D., and Graff, C. (2017), "UCI Machine Learning Repository," Available at *http://archive.ics.uci.edu/ml*. [2]

Ekin, T., Naveiro, R., Insua, D. R., and Torres-Barrán, A. (2022), "Augmented Probability Simulation Methods for Sequential Games," *European Journal of Operational Research*, 306, 418–430. [10]

Fan, J., Ma, C., and Zhong, Y. (2021), "A Selective Overview of Deep Learning," *Statistical Science*, 36, 264–290. [1]

Gallego, V., and Insua, D. R. (2018), "Stochastic Gradient MCMC with Repulsive Forces," in *Bayesian Deep Learning Workshop, Neural Information and Processing Systems*, Available at *http://bayesiandeeplearning.org/2018/papers/154.pdf*. [10]

Gallego, V., Naveiro, R., Redondo, A., Insua, D. R., and Ruggeri, F. (2020), "Protecting Classifiers From Attacks. A Bayesian Approach," arXiv:2004.08705. [6,7,8]

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016), *Deep Learning*, Cambridge, MA: MIT Press. [2]

Goodfellow, I., Shlens, J., and Szegedy, C. (2015), "Explaining and Harnessing Adversarial Examples," in *International Conference on Learning Representations*, Available at *https://arxiv.org/abs/1412.6572*. [1,9]

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. (2018), "On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models," arXiv:1810.12715. [3]

Hargreaves-Heap, S., and Varoufakis, Y. (2004), *Game Theory: A Critical Introduction*, New York: Routledge. [1]

Johansson, F., Shalit, U., and Sontag, D. (2016), "Learning Representations for Counterfactual Inference," in *International Conference on Machine Learning*, pp. 3020–3029, Available at *http://proceedings.mlr.press/v48/johansson16.pdf*. [10]

Joseph, A., Nelson, B., Rubinstein, B., and Tygar, J. (2019), *Adversarial Machine Learning*, Cambridge, UK: Cambridge University Press. [2]

Joshi, C., Aliaga, J. R., and Insua, D. R. (2021), "Insider Threat Modeling: An Adversarial Risk Analysis Approach," *IEEE Transactions on Information Forensics and Security*, 16, 1131–1142. [3]

Kadane, J. B., and Larkey, P. D. (1982), "Subjective Probability and the Theory of Games," *Management Science*, 28, 113–120. [5]

Kannan, H., Kurakin, A., and Goodfellow, I. (2018), "Adversarial Logit Pairing," arXiv:1803.06373. [5]

Kim, J.-H. (2009), "Estimating Classification Error Rate: Repeated Cross-Validation, Repeated Hold-Out and Bootstrap," *Computational Statistics and Data Analysis*, 53, 3735–3745. [2]

Kos, J., Fischer, I., and Song, D. (2018), "Adversarial Examples for Generative Models," in *2018 IEEE Security and Privacy Workshops (SPW)*, IEEE, pp. 36–42. [4]

LeCun, Y., Cortes, C., and Burges, C. (1998), "The MNIST Database of Handwritten Digits," Available at *http://yann.lecun.com/exdb/mnist/*. [2]

Lee, D., Moon, S., Lee, J., and Song, H. O. (2022), "Query-Efficient and Scalable Black-Box Adversarial Attacks on Discrete Sequential Data via Bayesian Optimization," in *International Conference on Machine Learning*, pp. 12478–12497, PMLR. [7]

Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., and Sun, M. (2017), "Tactics of Adversarial Attack on Deep Reinforcement Learning Agents," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, p. 3756–3762, AAAI Press. [5,10]

Liu, X., Li, Y., Wu, C., and Hsieh, C.-J. (2018), "ADV-BNN: Improved Adversarial Defense through Robust Bayesian Neural Network," arXiv:1810.01279. [5]

Lowd, D., and Meek, C. (2005), "Adversarial learning," in *Proceedings of the 11th ACM International Conference on Knowledge Discovery in Data Mining*, KDD'05, pp. 641–647. [4]

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018), "Towards Deep Learning Models Resistant to Adversarial Attacks," in *International Conference on Learning Representations* [4,5]

Menache, I., and Ozdaglar, A. (2011), "Network Games: Theory, Models, and Dynamics," *Synthesis Lectures on Communication Networks*, 4, 1–159. [1]

Miller, J. W., and Dunson, D. B. (2019), "Robust Bayesian Inference via Coarsening," *Journal of the American Statistical Association*, 114, 1113–1125. [10]

Naveiro, R. (2021), "Adversarial Attacks against Bayesian Forecasting Dynamic Models," in *22nd European Young Statisticians Meeting*, p. 66, Available at *https://arxiv.org/pdf/2110.10783.pdf*. [4,10]

Naveiro, R., and Insua, D. R. (2019), "Gradient Methods for Solving Stackelberg Games," in *International Conference on Algorithmic DecisionTheory*, pp. 126–140, Springer. [4,10]

Naveiro, R., Redondo, A., Ríos Insua, D., and Ruggeri, F. (2019a), "Adversarial Classification: An Adversarial Risk Analysis Approach," *International Journal of Approximate Reasoning*, 113, 133–148. [2,3,9]

Naveiro, R., Rodríguez, S., and Ríos Insua, D. (2019b), "Large-Scale Automated Forecasting for Network Safety and Security Monitoring," *Applied Stochastic Models in Business and Industry*, 35, 431–447. [3]

Papernot, N., Faghri, F., Carlini, N., Goodfellow, I., Feinman, R., Kurakin, A., Xie, C., Sharma, Y., Brown, T., Roy, A., Matyasko, A., Behzadan, V., Hambardzumyan, K., Zhang, Z., Juang, Y.-L., Li, Z., Sheatsley, R., Garg, A., Uesato, J., Gierke, W., Dong, Y., Berthelot, D., Hendricks, P., Rauber, J., and Long, R. (2018), "Technical Report on the CleverHans v2.1.0 Adversarial Examples Library," arXiv:1610.00768. [5]

Papernot, N., McDaniel, P., Swami, A., and Harang, R. (2016), "Crafting Adversarial Input Sequences for Recurrent Neural Networks," in *2016 IEEE Military Communications Conference*, IEEE, pp. 49–54. [4]

Park, T., and Casella, G. (2008), "The Bayesian Lasso," *Journal of the American Statistical Association*, 103, 681–686. [2]

Rios, J., and Insua, D. R. (2012), "Adversarial Risk Analysis for Counterterrorism Modeling," *Risk Analysis: An International Journal*, 32, 894–915. [6,8]

Rios Insua, D., Banks, D., and Rios, J. (2016), "Modeling Opponents in Adversarial Risk Analysis," *Risk Analysis*, 36, 742–755. [7]

Rios Insua, D., Couce-Vieira, A., Rubio, J. A., Pieters, W., Labunets, K., and G. Rasines, D. (2021), "An Adversarial Risk Analysis Framework for Cybersecurity," *Risk Analysis*, 14, 16–36. [3]

Rios Insua, D., Naveiro, R., and Gallego, V. (2020), "Perspectives on Adversarial Classification," *Mathematics*, 8. [8]

Rios Insua, D., Rios, J., and Banks, D. (2009), "Adversarial Risk Analysis," *Journal of the American Statistical Association*, 104, 841–854. [1,5]

Rios Insua, D., and Ruggeri, F. (2000), *Robust Bayesian Analysis*, Lecture Notes in Statistics (Vol. 152), New York: Springer. [10]

Ríos Insua, D., González-Ortega, J., Banks, D., and Ríos, J. (2018), "Concept Uncertainty in Adversarial Statistical Decision Theory," in *The Mathematics of the Uncertain*, eds. E. Gil, E. Gil, J. Gil, and M. Á. Gil, pp. 527–542, Cham: Springer. [3]

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021), "Toward Causal Representation Learning," *Proceedings of the IEEE*, 109, 612–634. [10]

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017), "Mastering the Game of Go Without Human Knowledge," *Nature*, 550, 354–359. [5]

Sutton, R. S., and Barto, A. G. (1998), *Introduction to Reinforcement Learning* (Vol. 2), Cambridge, MA: MIT Press. [5]

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014), "Intriguing Properties of Neural Networks," in *International Conference on Learning Representations*. [2,3]

Tuyls, K., and Weiss, G. (2012), "Multiagent Learning: Basics, Challenges, and Prospects," *AI Magazine*, 33, 41–41. [5]

Vorobeychik, Y., and Kantarcioglu, M. (2018), "Adversarial Machine Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12, 1–169. [2]

Wang, C., Bunel, R., Dvijotham, K., Huang, P.-S., Grefenstette, E., and Kohli, P. (2019), "Knowing When to Stop: Evaluation and Verification of Conformity to Output-Size Specifications," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 12260–12269. [10]

Welling, M., and Teh, Y. W. (2011), "Bayesian Learning via Stochastic Gradient Langevin Dynamics," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 681–688. [8]

Xiao, H., Rasul, K., and Vollgraf, R. (2017), "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms," arXiv:1708.07747. [9]

Ye, N., and Zhu, Z. (2018), "Bayesian Adversarial Learning," in *Proceedings of the 32nd International Conference on Advances in Neural Information Processing Systems*, pp. 6892–6901, Red Hook, NY: Curran Associates Inc. [7]

Zeager, M. F., Sridhar, A., Fogal, N., Adams, S., Brown, D. E., and Beling, P. A. (2017), "Adversarial Learning in Credit Card Fraud Detection," in *Systems and Information Engineering Design Symposium (SIEDS), 2017*, IEEE, pp. 112–116. [1]

Zhang, C., Zhang, K., and Li, Y. (2020a), "A Causal View on Robustness of Neural Networks," in *Advances in Neural Information Processing Systems* (Vol. 33), pp. 289–301. [10]

Zhang, W. E., Sheng, Q. Z., Alhazmi, A., and Li, C. (2020b), "Adversarial Attacks on Deep-Learning Models in Natural Language Processing: A Survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11, 1–41. [4]

Zhang, Y., Gong, M., Liu, T., Niu, G., Tian, X., Han, B., Schölkopf, B., and Zhang, K. (2021), "Adversarial Robustness through the Lens of Causality," arXiv:2106.06196. [1,10]