An Adversarial Risk Analysis Framework for Batch Acceptance Problems

Jorge González-Ortega[1], Refik Soyer[2], David Ríos Insua[3], Fabrizio Ruggeri[4]

[1] Facultad de Ciencias Matemáticas, UCM, Madrid, Spain
[2] Department of Decision Sciences, GWU, Washington, DC
[3]School of Management, USST, Shanghai, China &, Instituto de Ciencias Matemáticas, CSIC-UAM-UC3M-UCM, Madrid, Spain
[4] Istituto di Matematica Applicata e Tecnologie Informatiche, CNR, Milano, Italy

Abstract

We provide an adversarial risk analysis framework for batch acceptance problems in which a decision-maker relies exclusively on the size of the batch to accept or reject its admission to a system, while being aware of the presence of an opponent. The adversary acts as a data-fiddler attacker perturbing the observations perceived by the decision-maker through injecting faulty items and/or modifying the existing items to faulty ones. We develop optimal policies against this combined attack strategy and illustrate the methodology with a review spam example.

Keywords: adversarial hypothesis testing; data manipulation; security; review spam.

1 Introduction

In this paper, we deal with batch acceptance problems, where a decision-maker needs to accept or reject an incoming batch of items to a system based on certain observable features. These features may be either attributes (e.g. number of defective items on a sample) or variables (e.g. some summary statistic such as the mean quality). This class of problems encompasses a wide range of applications including acceptance sampling (Schilling and Neubauer, 2009), cargo container screening (Boros et al., 2009) and spam detection (Cormack and Lynam, 2007). We do not cover, though, batch admission problems to queuing systems, as for example in Stidham (1985).

Batch acceptance relates to the more general problem of hypothesis testing (French and Ríos Insua, 2000), which has been extensively studied from a decision theoretic perspective. However, lately, there has been a growing concern about the impact of adversaries in this realm, with recent examples in fields such as adversarial classification (Yu et al., 2018), adversarial machine learning (Jagielski et al., 2018) or adversarial signal processing (Tondi et al., 2019). This consideration of an adversarial component

also applies to the more specific batch acceptance framework, where rational opponents have been incorporated to the aforementioned contexts as in Tapiero (1995) (acceptance sampling), Haphuriwat et al. (2011) (cargo container screening) and Kantarcioglu et al. (2011) (spam detection).

Inclusion of adversaries in hypothesis testing and, in particular, batch acceptance problems, has been mainly undertaken within the scope of game theory, which invariably entails certain common knowledge assumptions. These usually relate to adversaries not only knowing their own payoffs, preferences, beliefs and possible actions, but also those of the other agents. This does not usually hold in the security contexts considered here; see e.g. Hargreaves-Heap and Varoufakis (2004) or Young (2004) for discussions.

In González-Ortega et al. (2019b), we provided an alternative approach to adversarial hypothesis testing based on the Adversarial Risk Analysis (ARA) paradigm (Ríos Insua et al., 2009) studying two types of adversaries: *data-fiddler attackers*, who perturb the data received by the decision-maker; and *structural attackers*, who alter the structure of the corresponding data generation process. In a comparative study of probabilistic risk analysis methods applied to container screening by Merrick and Parnell (2011), ARA was considered specially appropriate for decomposing complex probability distributions, apportioning uncertainty and explicitly showing the adaptation of each agent to their opponent's previous decisions.

Adopting the ARA framework for adversarial hypothesis testing, we present here a novel approach for batch acceptance problems in presence of adversaries. The problem faced is deciding whether to accept a batch of items received over a period of time, some of which could be faulty hence entailing potential security and/or performance problems. To simplify the discussion, the observable feature of the batch will be its size (the number of items in it). We will consider the potential intrusion of a data-fiddler attacker trying to obtain some gain by tampering with the batch prior to our observation. Though simple, this setup is quite general and applicable to many situations such as acceptance sampling (Lindley and Singpurwalla, 1991), where a manufacturer tries to determine the optimal number of replicates in an experiment to convince a consumer who has to decide whether to accept or reject the product. Our final discussion provides examples where other more complex observable features are relevant.

An application in review spam (Heydari et al., 2015) is provided to illustrate the approach. When buying a product, one typically gets feedback in a website from reviews of existing product users. The balance between positive and negative comments profoundly affects purchasing decisions and, thus, encourages review spam which may generate economic or reputational benefits. Typical review spam involves untruthful opinions, advertisements and duplicate reviews; see Jindal and Liu (2008) for a comprehensive classification. According to Xie et al. (2012), sharp increases in the volume of reviews correlate with recently arrived review spam. Therefore, we may develop batch acceptance policies based on monitoring the amount of reviews per product during certain periods of time to determine whether to further inspect for review spam related to specific products, which may be accomplished with techniques such as those in Li et al. (2011) or Lau et al. (2012).

Since we embrace a Bayesian viewpoint throughout the paper, we will make several assumptions about the involved likelihood functions and prior distributions. Our choices will be motivated, as in other Bayesian works, by their mathematical tractability and physical relevance as well as their flexibility in capturing expert opinions. In particular,

the adopted priors will depend on two parameters which can be easily determined from a few experts' assessments.

The outline of the paper is as follows. First, a non-adversarial version of the problem is provided in Section 2. Next, appropriate modifications are introduced in Section 3 to include a data-fiddler opponent, considering the combination of two strategies: faulty item injection and random item modification to faulty. A review spam example illustrates the methodology in Section 4, concluding with a discussion in Section 5. Proofs of all results are included in an Appendix.

2 Non-Adversarial Problem

We start with the non-adversarial version of the batch acceptance problem. A decision-maker (D, she) needs to decide whether to accept or reject a batch of items based on its observable size, without knowing its actual composition, consisting of acceptable and faulty items. She then incurs in a loss which depends on her decision and the batch content. As an example, within the aforementioned context of acceptance sampling, a retailer might face the decision of accepting or rejecting a sample of a manufacturer's product in terms of the provided number of replicates, before even testing them. She would then perceive opportunity costs if she rejects the sample and, otherwise, a loss related to the inspection of the sample and the outcome of its exact composition.

The non-adversarial batch acceptance problem is depicted by the Influence Diagram (ID) (Shachter, 1986) in Figure 1, which reflects the dependencies among the concerned random variables, the decision to be made and its consequences. As usual in IDs, circle nodes relate to random events or uncertainties, square nodes to decisions and hexagonal nodes to values (losses in our case). Arrows into chance or value nodes specify conditional dependence, whereas dashed arrows into decision nodes indicate information available when making the corresponding decision.

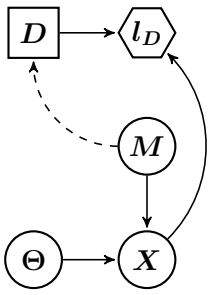

Figure 1: ID for the Non-Adversarial Batch Acceptance Problem.

The non-adversarial problem is described as follows, where capital letters denote chance and decision nodes in the ID and the corresponding values at these nodes are represented using lower case letters:

- The decision-maker needs to determine d, whether to accept ($d = d_0$) or reject ($d = d_1$) a batch of items (node D).
- For this, she observes the size $m \in \{0, 1, 2, \ldots\}$ of the batch (node M). The case $m = 0$ allows for empty batches.

- The items in the batch may be acceptable or not with an acceptability rate $\theta \in [0, 1]$ determining the probability that each item in the batch is acceptable (node Θ), which is unknown to the decision-maker. If Z designates this ($z = 0$, acceptable item; $z = 1$, otherwise), $p(z = 0 \,|\, \theta) = \theta$.
- The batch composition x includes two classes of items: 0, associated with acceptable items; and 1, corresponding to faulty ones (node X). We use x_0 to designate the number of acceptable items and, accordingly, $x_1 = m - x_0$ for the number of faulty items. The batch composition $x = (x_0, x_1)$ is not observed by the decision-maker before making his decision.
- Upon deciding d, and given x, the decision-maker obtains a loss $l_D(d, x)$ (node l_D).

We make two assumptions in this non-adversarial setup:

A1. The acceptability of each item in the batch is independent of that of the others and all items have the same rate. As a consequence, the number x_0 of acceptable items in a batch of size m will follow a binomial distribution $x_0 \,|\, m, \theta \sim \mathcal{B}in(m, \theta)$.

A2. As for the loss l_D, there are many possible choices. Let us consider the case where just allowing one faulty item is as bad as allowing several of them, because of the entailed security and/or performance issues: if D accepts a batch that contains at least one faulty item, she incurs in the worst possible loss, 1; if she blocks such a batch or accepts one with no faulty items, D attains the best loss, 0; finally, if she rejects a batch with all items acceptable, D faces an (expected) opportunity cost $c \in (0, 1)$. Examples where this is realistic abound in cyber security, specially in the case of autonomously propagated attacks (Ye et al., 2006), or terrorism, when considering attacks to series systems (Hausken and Levitin, 2012).

Given the above problem structure, we have:

Proposition 1. *Under Assumptions A1 and A2, the decision-maker's optimal policy in the non-adversarial batch acceptance problem is to accept the batch if and only if* $E_\theta[\theta^m] \geq (1 + c)^{-1}$. *Moreover, if* $p(\theta = 1) = 0$, *we can find a threshold value* m_1, *conditional on* c *and* $p_D(\theta)$, *such that the optimal decision is to reject the batch if* $m > m_1$.

An important case, standard in Bayesian analysis, holds when we have prior beliefs about the item acceptability rate θ modeled through a beta distribution $\mathcal{B}e(\alpha, \beta)$, as this type of distribution forms a conjugate family with the binomial distribution. In this context, if after receiving r items, s have been acceptable (and $r - s$, faulty), we update to the posterior $\theta \,|\, r, s \sim \mathcal{B}e(\alpha + s, \beta + r - s)$ (French and Ríos Insua, 2000). Its moment generating function satisfies

$$E_\theta[\theta^m] = \prod_{k=0}^{m-1} \frac{\alpha + s + k}{\alpha + \beta + r + k} = \frac{\alpha + s + m - 1}{\alpha + \beta + r + m - 1} E_\theta\left[\theta^{m-1}\right]. \tag{1}$$

Hence, according to Proposition 1, the decision-maker's optimal policy would be to accept the batch if and only if

$$\prod_{k=0}^{m-1} \frac{\alpha + s + k}{\alpha + \beta + r + k} \geq \frac{1}{1 + c}.$$

A rejection threshold m_1 on the number of items may be actually obtained recursively using equation (1). A simple scheme would rely on the fact that $E_\theta[\theta^0] = 1$ and start the iterative procedure with $m = 1$. Then, $E_\theta[\theta^m]$ would be computed and, if the relation in Proposition 1 is not satisfied, we would stop and determine $m_1 = m$. Otherwise, we would set $m = m + 1$ and repeat the calculation and verification.

3 Adversarial Problem

We engage now in the adversarial version of the problem. An attacker (A, he) interferes with the batch acceptance process to deceive the defender (D, she) and cause her to make wrong acceptance decisions from which to attain a certain benefit. Specifically, we consider a data-fiddler attacker who alters the incoming batch so that the information received by the defender is perturbed.

We represent the problem with a Bi-Agent Influence Diagram (BAID) (Banks et al., 2015) in Figure 2. Compared with the ID in Section 2, whose original elements are depicted in thicker line style, we reflect the problems of the defender and attacker integrated within a same structure. White nodes refer to issues solely affecting the defender's problem, whereas grey ones refer just to the attacker's problem. Chance nodes are striped implying that they model random events that are relevant for both agents' decision-making. Note though, that each decision-maker may entertain a different probability model over such shared chance nodes which, as anticipated, will not be common knowledge. For example, besides model $p_D(x \mid m, \theta)$ representing the defender's beliefs about batch composition x at node X, given the batch size m and the acceptability rate θ, we need to determine a model $p_A(x \mid m, \theta)$ for the attacker, which might differ from that of the defender.

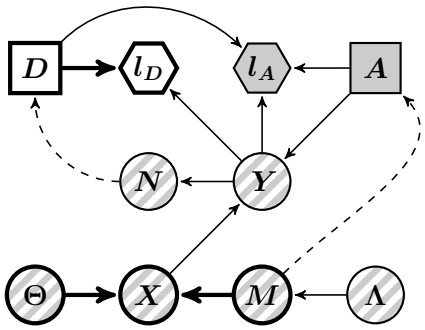

Figure 2: BAID for the Adversarial Batch Acceptance Problem (Thicker Line Style for Non-Adversarial ID).

The adversarial problem is then defined as follows, where we just specify the new elements when comparing with Figure 1:

- The attacker observes the batch size m and may choose to manipulate its content x, contingent on m and θ, through action a (node A), before the defender processes it. The attacker's inaction is modeled as a feasible action itself.
- The attack a will generally result in a new batch composition y derived from the original one x (node Y). Clearly, if the batch remains unperturbed, $y = x$.

- In particular, an altered batch composition y potentially leads to a new batch size n (node N), which would be observed by the defender, instead of the original size m, when making her acceptance decision. Though manipulated, the batch may keep its size unchanged, i.e. $n = m$.

- The original number m of items in the batch follows a generic one-parameter distribution related to an arrival rate λ (node Λ), unknown to the agents. This presumption was omitted in the non-adversarial problem as it had no impact in it, since the defender observes the actual value of m. However, the distribution of the arrival rate λ provides key information to the defender about the original batch size m in this adversarial context.

- Having selected action a, and given the defender's decision d and the final batch composition y, the attacker perceives a loss $l_A(a, d, y)$ (node l_A), in parallel to the loss $l_D(d, y)$ obtained by the defender.

The perturbation effect of the attacker's action a turning x into y, and m into n, poses a major difference with respect to the non-adversarial version in Figure 1. For that reason, the defender should not ignore the possible interference of the attacker, since this may entail a performance degradation of the decision algorithm, as will be discussed later.

To progress in the analysis, we need assumptions about the potential actions undertaken by the attacker such as the following:

A3. The attacker may fiddle with the data combining two types of perturbations: (1) *faulty item injection*, incorporating his own faulty items into the batch; and (2) *random item modification to faulty*, randomly selecting items from the batch (recall he is unaware of the original batch composition) and turning them into faulty.

The original faulty items in the batch will be designated O-faults (outer faults), while faulty items provided by the attacker will be called A-faults (attacker faults). As a consequence of Assumption A3, there will be A-faults of two classes in the final batch: (1) y_1 faulty items injected by the attacker, where $0 \leq y_1$; and (2) y_2 original items modified to faulty without distinguishing the type of items being changed, y_2^0 acceptable and y_2^1 O-fault, where $y_2^0 + y_2^1 = y_2$ and $0 \leq y_2^0 \leq x_0$, $0 \leq y_2^1 \leq m - x_0$. Thus, the final composition of the batch received (yet unobserved) by the defender consists of $x_0 - y_2^0$ acceptable items, $m - x_0 - y_2^1$ O-faults and $y_1 + y_2$ A-faults.

We consider now the optimal defensive policy in the adversarial batch acceptance problem in light of Assumptions A1, A2 and A3.

3.1 The Defender's Problem

Figure 3 displays the ID associated to the defender's perspective of the adversarial batch acceptance problem. Observe the differences with the BAID in Figure 2: the attacker's loss node is omitted, as it is irrelevant to her and, more importantly, his decision node is transformed into a chance node, as the defender is uncertain about the attacker's action. Notice also the impact of the attacker's inclusion on the defender's non-adversarial problem structure from the ID in Figure 1.

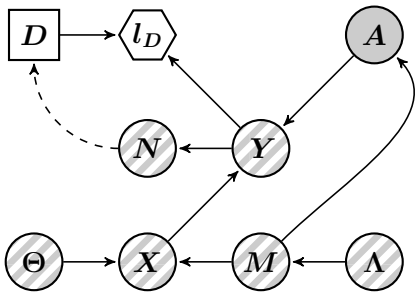

Figure 3: Defender's Problem.

We identify hereafter the defender's optimal policy against a data-fiddler attacker that operates according to Assumption A3. To begin with, suppose the defender knows the distribution $p_D(y_1, y_2 \,|\, m)$ describing her beliefs about the attacker's choice on how many faulty items y_1 to inject and how many items y_2 to randomly modify to faults of his, should the original batch size be m; that is, the distribution modeling her understanding about the attacks she potentially faces. We provide first an auxiliary result:

Lemma 1. *Suppose that the observed batch size is n. Under Assumption A3, the probability $q(n \,|\, \theta, \lambda)$ that all n items in the final batch are acceptable, given the acceptability θ and arrival λ rates, is*

$$q(n \,|\, \theta, \lambda) \coloneqq \frac{p_D(y_1 = 0, y_2 = 0 \,|\, m = n)\, p_D(m = n \,|\, \lambda)}{\sum_{i=0}^{n} p_D(y_1 = n - i \,|\, m = i)\, p_D(m = i \,|\, \lambda)}\, \theta^n.$$

Based on it, the following result parallels Proposition 1 in the adversarial problem:

Proposition 2. *Under Assumptions A1, A2 and A3, the defender's optimal policy in the adversarial batch acceptance problem is to accept the batch if and only if $E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right] \geq (1 + c)^{-1}$. Moreover, if $p(\theta = 1) = 0$, we can find a threshold value n_1, conditional on c, $p_D(\theta)$, $p_D(\lambda)$ and $p_D(y_1 = 0, y_2 = 0 \,|\, n, \lambda)$, such that the optimal decision is to reject the batch if $n > n_1$.*

We can now compare the acceptance rules derived in the adversarial and non-adversarial versions, thus contrasting an adversary-aware defender with an adversary-unaware one. The latter relates to decision-makers who ignore the presence of adversaries, hence believing that the observed batch size n is the original one, which should rather be m.

Corollary 1. *Under assumptions A1, A2 and A3, threshold values m_1 and n_1 for the optimal acceptance policies, respectively defined in Propositions 1 and 2, fulfill $m_1 \geq n_1$*
.

Therefore, we conclude that the adversary-aware defender would always take a more prudent decision than the adversary-unaware one, as she takes into account the information about the adversary at her disposal.

To further advance in the analysis, we would need to make distributional assumptions. For example, in the beta-binomial case concerning the acceptability rate θ (Section 2), we have

$$E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right] = \left(\prod_{k=0}^{n-1} \frac{\alpha + s + k}{\alpha + \beta + r + k}\right) E_{\lambda}\left[\frac{p_D(y_1 = 0, y_2 = 0 \,|\, m = n)\, p_D(m = n \,|\, \lambda)}{\sum_{i=0}^{n} p_D(y_1 = n - i \,|\, m = i)\, p_D(m = i \,|\, \lambda)}\right].$$

As with respect to the arrival rate λ, a typical assumption would be that the number m of original items follows a Poisson distribution with an average of λ items so that $m \,|\, \lambda \sim \mathcal{P}o(\lambda)$. In addition, suppose that the prior over λ is a gamma distribution $\mathcal{G}a(a, b)$. After t batches in which, in total, r items have arrived, the posterior would be $\lambda \,|\, t, r \sim \mathcal{G}a(a + r, b + t)$. Then, it holds

$$E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right] = \left(\prod_{k=0}^{n-1} \frac{\alpha + s + k}{\alpha + \beta + r + k}\right) E_{\lambda}\left[\frac{p_D(y_1 = 0, y_2 = 0 \,|\, m = n)\, \frac{\lambda^n}{n!}}{\sum_{i=0}^{n} p_D(y_1 = n - i \,|\, m = i)\, \frac{\lambda^i}{i!}}\right], \tag{2}$$

which would be computed by simulation.

To completely determine the defender's optimal decision for a given observed batch size n, we still need to assess $p_D(y_1, y_2 \,|\, m)$ which has a strategic component, as it refers to the optimal decision of an attacker. We could assess this distribution via standard structured expert judgement methods as in e.g. Cooke (1991). However, as noted in Ríos Insua et al. (2020), improved forecasts are frequently obtained using the structural decomposition in ARA to assess a probability distribution over the attacker's possible actions reflecting the defender's uncertainty about them. For this, ARA suggests a decomposition approach that requires the defender to consider the attacker's problem from her perspective to obtain the attack distribution $p_D(y_1, y_2 \,|\, m)$ and, thus, the probability $q(n \,|\, \theta, \lambda)$ needed in Proposition 2, as we next consider.

3.2 The Attacker's Problem

As the underlying principle for the assessment of $p_D(y_1, y_2 \,|\, m)$, we consider the attacker to minimize his expected loss in choosing his optimal attack (y_1^*, y_2^*) and our uncertainty about such optimal choice through the random optimal policy $(Y_1^*, Y_2^*)(m)$, which leads to $p_D(y_1, y_2 \,|\, m) = P((Y_1^*, Y_2^*)(m) = (y_1, y_2) \,|\, m)$. To facilitate this, the ID in Figure 4 portrays the defender's perspective of the attacker's problem. As opposed to the ID in Figure 3 and the BAID in Figure 2, the defender's loss node is suppressed, having just implicit interest for the attacker, and her decision node is turned into a chance node, as the attacker is uncertain about the defender's choice concerning the eventual acceptance of the batch. His beliefs about this distribution shall be denoted by $p_A(d_0 \,|\, n)$.

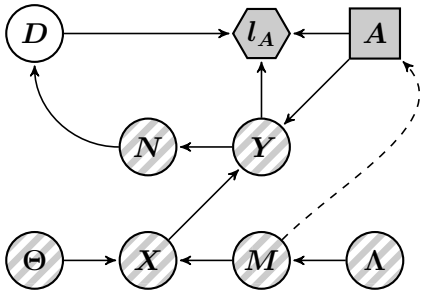

Figure 4: Attacker's Problem.

To be able to solve the attacker's problem, the defender could consider Assumptions A1 and A3 applicable to his problem, presuming that they are pertinent in the attacker's view. This is a natural premise as the defender will relate the attacker's prospect of the problem to hers. Still, an additional assumption needs to be made concerning the

attacker's loss l_A for which, as with the defender's losses, there are many conceivable structures. We adopt the following:

A4. The parameters involved in the attacker's loss are his expected gain g due to each A-fault, his expected gain h due to each O-fault (these could help him conceal his A-faults when the defender inspects an accepted batch), his unitary cost f_1 of injecting A-faults and his incurred cost f_2 of changing one item to faulty.

Given the identified attacker's problem structure, designate by $\gamma(y_1) \coloneqq g\, p_A(d_0 \,|\, n = m + y_1)$ his expected gain per A-fault when the defender perceives the batch size to be $n = m + y_1$. Then:

Proposition 3. *Under Assumptions A1, A3 and A4, the attacker's optimal policy for the adversarial batch acceptance problem is to inject y_1^* A-faults and (randomly) modify y_2^* items to faulty so that the combined attack $(y_1^*, y_2^*)(m)$ minimizes in y_1 and y_2 his expected loss*

$$\psi_A(y_1, y_2 \,|\, m) = y_1\,(f_1 - \gamma(y_1)) + y_2\left(f_2 - \left(1 - \tfrac{h}{g}\,(1 - E_A[\theta])\right)\gamma(y_1)\right) - m\,\tfrac{h}{g}\,(1 - E_A[\theta])\,\gamma(y_1). \tag{3}$$

Moreover: (i) the attacker's optimal amount y_1^* of injected items is finite provided that $p_A(d_0 \,|\, n) \leq \frac{f_1 - \varepsilon}{g}$ for all $n \geq n_0$ for certain threshold value n_0 and $\varepsilon > 0$; and (ii) the attacker's optimal amount y_2^* of (randomly) modified items to faulty is 0 when $f_2 > \left(1 - \frac{h}{g}\,(1 - E_A[\theta])\right)\gamma(y_1^*)$, and m when $f_2 < \left(1 - \frac{h}{g}\,(1 - E_A[\theta])\right)\gamma(y_1^*)$.

Notice, however, that the defender lacks information about the required attacker's loss and probabilities components. Suppose she acknowledges such uncertainty over those ingredients through random parameters and probabilities $\big(G, H, F_1, F_2, P_A(d_0 \,|\, n), P_A(\theta)\big)$. Without loss of generality, assume they are all defined over a common probability space $(\Omega, \mathcal{A}, \mathcal{P})$ with atomic elements $\omega \in \Omega$ (Chung, 2001). For example, $P_A^{\omega}(\theta)$ defines an instance of $P_A(\theta)$ and, based on it, $E_A^{\mathcal{P},\omega}[\theta]$ determines an occurrence of $E_A[\theta]$. Let $E_A^{\mathcal{P}}[\theta]$ denote the attacker's random expected value of θ and $\Gamma(Y_1) \coloneqq G\, P_A(d_0 \,|\, n = m + Y_1)$ his random expected gain per A-fault when the defender receives a batch of size $n = m + Y_1$. We then have:

Proposition 4. *From the defender's perspective, and under Assumptions A1, A3 and A4, the attacker's random optimal policy for the adversarial batch acceptance problem is $(Y_1^*, Y_2^*)(m)$ minimizing in Y_1 and Y_2 his random expected loss*

$$\Psi_A(Y_1, Y_2 \,|\, m) = Y_1\,(F_1 - \Gamma(Y_1)) + Y_2\left(F_2 - \left(1 - \tfrac{H}{G}\left(1 - E_A^{\mathcal{P}}[\theta]\right)\right)\Gamma(Y_1)\right) - m\,\tfrac{H}{G}\left(1 - E_A^{\mathcal{P}}[\theta]\right)\Gamma(Y_1).$$

Moreover, if $P_A\left(F_2 = \left(1 - \frac{H}{G}\left(1 - E_A^{\mathcal{P}}[\theta]\right)\right)\Gamma(Y_1^*)\right) = 0$, the attacker's random optimal amount Y_2^* of (randomly) modified items to faulty will almost surely be 0 or m.

Due to the complexity of directly evaluating $(Y_1^*, Y_2^*)(m)$ through Proposition 4, the defender may resort to a Monte Carlo approximation, see Caflisch (1998) for a general discussion, as specified in Algorithm 1. This involves drawing from the involved components $\big(G, H, F_1, F_2, P_A(d_0 \,|\, n), P_A(\theta)\big)$, computing the corresponding optimal amount of injected and modified A-faults to obtain a sample $\{(Y_{1,k}^*, Y_{2,k}^*)(m)\}_{k=1}^K$ of size K from $(Y_1^*, Y_2^*)(m)$, and then estimating $\hat{p}_D(y_1, y_2 \,|\, m) \approx \#\{(Y_{1,k}^*, Y_{2,k}^*) = (y_1, y_2)\}/K$.

Algorithm 1 Forecasting the Attacker's Choices.

Data: Original batch size m; number of iterations K; upper bound for the amount of injected items $\overline{Y}_1$.

Set $\hat{p}_D(y_1, y_2 \,|\, m) = 0$ for $y_1 = 0, 1, \ldots, \overline{Y}_1$, $y_2 = 0, m$.
For $k = 1$ **to** K **do**
  Sample $g_k \sim G$, $h_k \sim H$, $f_{1,k} \sim F_1$ and $f_{2,k} \sim F_2$.
  Sample distribution $p_k(\theta) \sim P_A(\theta)$ and compute $E_{\theta,k} = \int \theta \, p_k(\theta) \, \mathrm{d}\theta$.
  For $y_1 = 0$ **to** $\overline{Y}_1$ **do**
    Sample $\pi_{0,k}(y_1) \sim P_A(d_0 \,|\, n = m + y_1)$ and compute $\gamma_k(y_1) = g_k \, \pi_{0,k}(y_1)$.
    Compute $\psi_k(y_1, 0) = y_1 \, (f_{1,k} - \gamma_k(y_1)) - m \, \frac{h_k}{g_k} \, (1 - E_{\theta,k}) \, \gamma_k(y_1)$.
    Compute $\psi_k(y_1, m) = y_1 \, (f_{1,k} - \gamma_k(y_1)) + m \, (f_{2,k} - \gamma_k(y_1))$.
  End For
  Find $(y_1^*, y_2^*) = \underset{y_1, y_2}{\arg\min} \; \psi_k(y_1, y_2)$ and set $\hat{p}_D(y_1^*, y_2^* \,|\, m) = \hat{p}_D(y_1^*, y_2^* \,|\, m) + 1$.
End For
Set $\hat{p}_D(y_1, y_2 \,|\, m) = \hat{p}_D(y_1, y_2 \,|\, m)/K$ for $y_1 = 0, 1, \ldots, \overline{Y}_1$, $y_2 = 0, m$.

Convergence of this algorithm to the required probability distribution $p_D(y_1, y_2 \,|\, m)$ is guaranteed by the Strong Law of Large Numbers. See Robert and Casella (2013) who also provide arguments to choose the K required to reach a desired precision in the approximation (based on the Central Limit Theorem).

With regard to modeling the involved attacker's random losses and probabilities, typical assumptions would be:

- Upper and lower bounds on gains and costs may be provided in general based on the available knowledge. Under the assumption of lack of additional information, they can be considered uniformly distributed: $G \sim \mathcal{U}(g^-, g^+)$, $H \sim \mathcal{U}(h^-, h^+)$, $F_1 \sim \mathcal{U}(f_1^-, f_1^+)$ and $F_2 \sim \mathcal{U}(f_2^-, f_2^+)$. When further information is at hand, e.g. their mode or several quantiles, other reasonable models would be triangular or shifted beta distributions.

- $P_A(d_0 \,|\, n)$ may be modeled through a uniform distribution if the defender does not acknowledge a strategic assessment of the attacker about her own decision-making process. However, if she does consider that the attacker deems her as attacker-aware, a distribution compatible with a threshold policy as the one found in Proposition 2 could be used, which might be the prelude of a hierarchy of decision-making problems and require further recursion. See Ríos and Ríos Insua (2012) for a description of the potentially infinite regress in a simpler class of problems.

- As for $P_A(\theta)$, based on a principle of using $p_D(\theta)$ with some uncertainty, we may model it through a Dirichlet process (Ferguson , 1973) with base given by a beta distribution $\mathcal{B}e(\alpha + s, \beta + r - s)$ and concentration parameter ρ, denoted $\mathcal{D}ir\mathcal{P}(\mathcal{B}e(\alpha + s, \beta + r - s), \rho)$.

4 An Example in Review Spam

As an illustration, we present a review spam example built upon the model in Section 3. Simplifications have been made for the sake of a better understanding of the methodology and a reduction of the problem's size and computation time. We support a merchant website manager (defender) who tries to detect review spam about their purchasable products. She regards each daily set of reviews per product as a batch and monitors the number of reviews within it. If this is under a certain threshold, all comments are accepted. However, if such threshold is exceeded, then she further inspects the batch checking additional available information like the origins of each review and/or their actual content as suggested in the references in Section 1. Thus, the proposed approach serves for screening for spam in review batches.

Suppose the website manager is aware that some product is of current interest to a recurrent spammer (attacker) on the website. Common actions for this spammer could be to provide spam reviews from fake user accounts (faulty item injection, see Ramilli and Prandini (2009) for information on spam injection) and/or altering reviews by compromising accounts (item modification to faulty, see Ruan et al. (2015) for details on compromised account behaviour).

We specify now the ingredients involved in the batch acceptance problem. The choice of the form and the parameters of the prior distributions and, similarly, of the utilities should be the result of an elicitation process, e.g. as described in O'Hagan et al. (2006). For illustrative purposes, we select distributions which are convenient from a mathematical point of view, e.g. gamma priors which are conjugate with respect to the Poisson model. Other distributions, such as lognormal or Weibull, could have been chosen making the computation more cumbersome but numerically easily tractable. For the same explanatory purposes, the choice of the parameters is clearly arbitrary, but we briefly describe typical ways of eliciting them in practice.

- The probability θ that a review is acceptable. We consider a beta distribution $\mathcal{B}e(9,1)$ as its prior. Procedures for eliciting the beta distribution parameters have been well developed in the literature (Chaloner and Duncan, 1983). The choice of parameters could be determined by assessments from various experts on the expected acceptability of any single review, along with their uncertainty about it and a range of the most likely values. In statistical terms, the $\mathcal{B}e(9,1)$ prior corresponds to an expected probability of a review's acceptability of 0.9 with variance 0.008, while there is an approximate 90% probability of having an acceptability rate in the interval $[0.717, 0.994]$.

- The current rate λ of original posted reviews on the product. We associate a gamma distribution $\mathcal{G}a(5,1)$ as its prior. We could resort again to expert judgement and expect an original batch size of 5 daily reviews with an equal variance, implying that there is an approximate 90% probability of having a value in the interval $[1.970, 9.150]$. The number m of actual reviews per day will follow a Poisson distribution with an average of λ reviews so that $m \,|\, \lambda \sim \mathcal{P}o(\lambda)$.

- The (expected) costs associated with rejecting a batch with all acceptable reviews and further inspecting it will be assumed to be $c = 0.9$ utility units.

As for the attacker's problem, suppose the following assessments are made:

- His gains and costs will be uniformly distributed as $G \sim \mathcal{U}(0.8, 1)$, $H \sim \mathcal{U}(0, 0.25)$, $F_1 \sim \mathcal{U}(0.25, 0.5)$ and $F_2 \sim \mathcal{U}(0.3, 0.6)$. Two implicit assumptions are: (i) the expected gain due to his own spam reviews (A-faults) is greater than that due to already existing spam reviews (O-faults) as the attacker may better design them to fulfill his objectives ($E[G] > E[H]$); and, (ii) on average, injecting spam reviews involves less effort for the attacker than modifying comments by compromising accounts as he has more control over the process ($E[F_1] < E[F_2]$).

- $P_A(d_0 \,|\, n)$ will be modeled through a uniform distribution dependent on the final batch size n. To avoid recursions, consider that the attacker relates it to the defender's non-adversarial context in Section 2. In terms of the batch original expected acceptability, he could presume that the defender admits batches of size n with probability $E_\theta[\theta^n]$. Additionally, he could weigh that probability by 0.5, admitting that the defender might suspect him to be manipulating every other batch. Then, we estimate

$$E[P_A(d_0 \,|\, n)] = \frac{E_\theta[\theta^n]}{2} = \frac{1}{2}\prod_{k=0}^{n-1}\frac{9+k}{10+k} = \frac{9}{18+2n},$$

  making use of the defender's prior over θ and expression (1). To allow for some uncertainty, and assuming that $P_A(d_0 \,|\, n) > P_A(d_0 \,|\, n+1)$ for every $n \in \mathbb{N}$, we adopt

$$P_A(d_0 \,|\, n) \sim \mathcal{U}\left(\frac{9}{19+2n}, \frac{9\,\frac{10+n}{9+n}}{19+2n}\right).$$

  For the case $n = 0$, we consistently assume $P_A(d_0 \,|\, n = 0) = 1$.

- $P_A(\theta)$ will be a Dirichlet process with a $\mathcal{B}e(9, 1)$ base and concentration parameter $\rho = 100$, i.e. $\mathcal{D}ir\mathcal{P}(\mathcal{B}e(9, 1), 100)$. A Bayesian approach on how to assess the concentration parameter of a Dirichlet process, given its base distribution, relying both on available information and/or expert judgement may be found in Dorazio (2009).

Recall that $y_2^* \in \{0, m\}$. Therefore, the attack probabilities for each original batch size m may be estimated using Monte Carlo simulation adapting Algorithm 1.

Algorithm 2 Forecasting the Spammer's Choices.

Data: Original batch size m; number of iterations K; upper bound for the amount of injected spam reviews $\overline{Y}_1$.

Set $\hat{p}_D(y_1, y_2 \,|\, m) = 0$ for $y_1 = 0, 1, \ldots, \overline{Y}_1$, $y_2 = 0, m$.

For $k = 1$ **to** K **do**

  Sample $g_k \sim \mathcal{U}(\frac{4}{5}, 1)$, $h_k \sim \mathcal{U}(0, \frac{1}{4})$, $f_{1,k} \sim \mathcal{U}(\frac{1}{4}, \frac{1}{2})$ and $f_{2,k} \sim \mathcal{U}(\frac{3}{10}, \frac{3}{5})$.

  Sample $p_k(\theta) \sim \mathcal{D}ir\mathcal{P}(\mathcal{B}e(9, 1), 100)$ and compute $E_{\theta,k} = \int \theta \, p_k(\theta) \, \mathrm{d}\theta$.

  For $y_1 = 0$ **to** $\overline{Y}_1$ **do**

    If $m = 0$ **and** $y_1 = 0$ **then**

      Set $\gamma_k(y_1) = g_k$.

    Else

      Sample $\pi_{0,k}(y_1) \sim \mathcal{U}\left(\dfrac{9}{19 + 2m + 2y_1}, \dfrac{9\,\frac{10+m+y_1}{9+m+y_1}}{19 + 2m + 2y_1}\right)$ and compute $\gamma_k(y_1) = g_k \, \pi_{0,k}(y_1)$.

    End If

    Compute $\psi_k(y_1, 0) = y_1 \, (f_{1,k} - \gamma_k(y_1)) - m \, \frac{h_k}{g_k} \, (1 - E_{\theta,k}) \, \gamma_k(y_1)$.

    Compute $\psi_k(y_1, m) = y_1 \, (f_{1,k} - \gamma_k(y_1)) + m \, (f_{2,k} - \gamma_k(y_1))$.

  End For

  Find $(y_1^*, y_2^*) = \underset{y_1, y_2}{\arg\min} \; \psi_k(y_1, y_2)$ and set $\hat{p}_D(y_1^*, y_2^* \,|\, m) = \hat{p}_D(y_1^*, y_2^* \,|\, m) + 1$.

End For

Set $\hat{p}_D(y_1, y_2 \,|\, m) = \hat{p}_D(y_1, y_2 \,|\, m)/K$ for $y_1 = 0, 1, \ldots, \overline{Y}_1$, $y_2 = 0, m$.

Tables 1 and 2 reflect an application of the scheme with $K = 10^3$ iterations (sufficient for illustrative purposes) and an upper bound for the amount of injected spam reviews of $\overline{Y}_1 = 5$, leading to the estimates of $\hat{p}_D(y_1, y_2 \,|\, m)$ for an original batch size of $m = 0, 1, \ldots, 10$. According to both tables, the relevant final batch size possibilities may be constrained to $n = 0, 1, \ldots, 10$.

Table 1: Defender's Estimation of $\hat{p}_D(y_1, y_2 = 0 \,|\, m)$.

	Original Batch Size - m										
Attack - y_1	0	1	2	3	4	5	6	7	8	9	10
0	0.394	0.365	0.543	0.710	0.841	0.923	0.969	0.992	1.000	1.000	1.000
1	0.294	0.204	0.167	0.134	0.105	0.062	0.029	0.008	0.000	0.000	0.000
2	0.214	0.134	0.092	0.055	0.029	0.006	0.002	0.000	0.000	0.000	0.000
3	0.084	0.051	0.029	0.008	0.000	0.001	0.000	0.000	0.000	0.000	0.000
4	0.014	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 2: Defender's Estimation of $\hat{p}_D(y_1, y_2 = m \,|\, m)$.

	Original Batch Size - m										
Attack - y_1	0	1	2	3	4	5	6	7	8	9	10
0	0.394	0.191	0.149	0.088	0.025	0.008	0.000	0.000	0.000	0.000	0.000
1	0.294	0.036	0.015	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.214	0.011	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.084	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.014	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

We conclude that:

- The spammer will unlikely alter all original reviews by compromising accounts ($y_2 = m$), but rather just favour the production of his own spam reviews from fake user accounts without compromising any accounts ($y_2 = 0$).
- Attacks are only likely for small amounts of original reviews ($m = 0, 1$ in the present example), avoiding significant perturbations of the batch size (given the small probabilities for $y_1 > 2$).

Making use of Tables 1 and 2 to compute $q(n \mid \theta, \lambda)$ in Lemma 1, the website manager may estimate the expected probability that all n reviews she is monitoring are acceptable. Table 3 provides such probabilities, as well as the optimal choice based on decision rule in Proposition 2 with $1/(1 + c) = 0.526$ (being $c = 0.9$).

Table 3: Defender's Optimal Decision Given the Final Amount of Reviews.

	Final Batch Size - n										
	0	1	2	3	4	5	6	7	8	9	10
Accept, d_0	Yes	No	Yes	Yes	Yes	No	No	No	No	No	No
$E_{\theta,\lambda}\left[q(n \mid \theta, \lambda)\right]$	1.000	0.524	0.532	0.535	0.529	0.509	0.511	0.526	0.512	0.500	0.474

The following remarks stem from Tables 1, 2 and 3:

- When an empty batch is received ($n = 0$), the model obviously accepts the batch since there cannot be spam reviews.
- For smaller original batch sizes, the spammer is encouraged to both inject spam reviews and/or modify comments by compromising accounts as it is more likely that all original reviews are acceptable. This might cause the website manager to further inspect (reject) batches with a small size ($n = 1$ in our example).
- For bigger original batch sizes, the spammer is discouraged to intervene and thus avoid costs as it is more likely that some original reviews are already spam. This might lead the website manager to accept batches with a medium size ($n = 2, 3, 4$ in this example).
- In line with Proposition 2, there is a threshold size ($n_1 = 5$ in the present example) such that any monitored batch bigger than that will always be rejected (hence investigated) by the website manager, as she will expect the original batch to already include spam reviews.

To exemplify the Bayesian updating of the defender's optimal acceptance policy, assume that 38 reviews have been posted about the concerned product within a week and the website manager has inspected all of them determining that just 4 were spam. According to expression (2), and identifying $t = 7$ (received batches), $r = 38$ (total amount of items in the batches) and $s = 34$ (acceptable items), Table 4 reflects the updated optimal decision rule, which differs from the initial one.

Table 4: Defender's Updated Decision Given the Final Amount of Reviews.

	Final Batch Size - n										
	0	1	2	3	4	5	6	7	8	9	10
Accept, d_0	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	No
$E_{\theta,\lambda}\left[q(n \mid \theta, \lambda)\right]$	1.000	0.534	0.547	0.548	0.536	0.502	0.480	0.463	0.433	0.402	0.366

We end up by providing a comparison with what would be the behaviour of an adversary-unaware decision-maker, as discussed in Corollary 1. The acceptance rule for an adversary-unaware defender is given by Proposition 1 based on $E_\theta[\theta^n]$. In our example, this results in the optimal policy depicted in Table 5, leading the website manager to accept all batches with $n \leq 8$, so that the threshold value alluded in such proposition is $m_1 = 8$, indeed verifying $m_1 \geq n_1$. This is caused by having high values for both the expected acceptability rate (0.9) and the costs associated with inspecting a batch (reject) when all reviews are acceptable ($c = 0.9$), so that the adversary-unaware defender has low incentives for inspecting small to medium size batches. Hence, we may safely conclude that by only considering the potential inclusion of O-faults in the batch and not acknowledging the addition of A-faults, the website manager's decision policy is clearly less prudent from that used by the adversary-aware defender.

Table 5: Adversary-Unaware Defender's Decision Given the Final Amount of Reviews.

	Final Batch Size - n										
	0	1	2	3	4	5	6	7	8	9	10
Accept, d_0	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
$E_\theta[\theta^n]$	1.000	0.900	0.818	0.750	0.692	0.643	0.600	0.562	0.529	0.500	0.474

5 Discussion

We have provided an ARA framework to deal with adversarial batch acceptance problems. In this way, symmetric losses and strong common knowledge assumptions typical of non-cooperative game theory are avoided. We have assumed that we were supporting a defender who needs to accept or reject an incoming batch of items observing its size, while a purposeful attacker may perturb the batch composition and, potentially, its size, prior to her observation. In doing this, the defender has to forecast the attacker's action and find her own optimal alternative.

We have assumed that the defender only perceives the batch size, although other observable features may be recognized and incorporated to the model separately or using more complex metrics, e.g. as in cargo container screening (Gaukler et al., 2012; Dreiding and McLay, 2013; Merrick and Albert, 2018) or spam detection (Sculley and Wachman, 2007; Zhang et al., 2014; Luckner et al., 2014). When the defender has no information about the observable feature(s) other than her previous experience, we could think of a multi-stage version of the model in Section 3 or a sequential hypothesis testing framework (Tartakovsky et al., 2014) as in compromised machine detection (Duan et al., 2012). Other loss functions for the defender could be explored as well, e.g. depending on the number of faulty items in the batch, as in Schlenker et al. (2016). Regarding the attacker, different loss structures could be explored as well.

Finally, our batch acceptance model has been illustrated with a review spam example considering a simplified version of a merchant website manager trying to detect review spam on purchasable products. When applying the model to a real case, a multi-period problem allowing for information updating, as in Zhuang et al. (2010) or Hausken and Zhuang (2011), or deeper recursive thinking strategies, as in McLay et al. (2012), Ríos and Ríos Insua (2012) or Shapiro et al. (2014), could be considered. In addition, new strategies for the attacker, such as the injection of (apparently) non-spam reviews to confound the defender (Duan et al., 2012), could be used as an evasion technique.

Moreover, the defender could use alternative procedures to detect spam reviews based on batches consisting of weekly reviews per user towards identifying fraudulent rating behaviour as in Lim et al. (2010) or Hooi et al. (2016).

Batch acceptance problems with multiple attackers are also of significance (Hausken and Bier, 2011), where an ARA perspective would support the defender versus all attackers. In such cases, we would need to determine the relationship between the attacks which could be completely independent, influence somehow each other or be partially or totally coordinated. Multiple defenders could also be considered (Jiang et al., 2013) with different cooperation levels and their own observations of the batch features.

Proofs of Results

Proof of Proposition 1: Under Assumptions A1 and A2, Table 6 reflects the losses suffered by the decision-maker for both of her choices under the two key scenarios: (i) having a batch with all items acceptable; and (ii) having some (at least one) faulty items in the batch. The table also displays the probability of such scenarios for a batch with m items, as well as the expected losses for each decision, given θ.

Table 6: Decision-maker's Loss Structure - Batch of $\boldsymbol{m}$ Items.

D's Decision	All Acceptable	Some Faulty	Exp. Loss
Accept, d_0	0	1	$1-\theta^m$
Reject, d_1	c	0	$c\,\theta^m$
Probability	θ^m	$1-\theta^m$	

As displayed in Table 6, the expected losses of both decisions d_0 (accept) and d_1 (reject) with respect to the unknown item acceptability rate θ are, respectively,

$$\psi_D(d_0) = E_\theta\left[1-\theta^m\right] = 1 - E_\theta\left[\theta^m\right], \qquad \psi_D(d_1) = E_\theta\left[c\,\theta^m\right] = c\,E_\theta\left[\theta^m\right].$$

The decision-maker's optimal decision is then to accept the batch (d_0) if and only if

$$\psi_D(d_1) \geq \psi_D(d_0) \iff c\,E_\theta\left[\theta^m\right] \geq 1 - E_\theta\left[\theta^m\right] \iff E_\theta\left[\theta^m\right] \geq \frac{1}{1+c}. \tag{4}$$

Now, if $p(\theta = 1) = 0$, so that $\theta \in [0,1)$ almost surely, the expected value $E_\theta\left[\theta^m\right]$ is a decreasing function in m converging to 0 as $m \to \infty$. Therefore, there is a threshold value m_1 such that the optimal decision is to reject the batch (d_1) if $m > m_1$. □ □

Proof of Lemma 1: Let us express the probability that all items in a final batch of size n are acceptable as $q(n\,|\,\theta,\lambda) = p_D(x_0 = n, y_1 = 0, y_2 = 0\,|\,n,\theta,\lambda)$. According to the Law of Total Probability, the probability of having a final batch with size $n = m + y_1$ items, given λ, is

$$p_D(n\,|\,\lambda) = \sum_{i=0}^{n} p_D(y_1 = n-i\,|\,m=i)\,p_D(m=i\,|\,\lambda),$$

reflecting all feasible combinations of m initial batch sizes and y_1 injected faulty items. Then, we deduce that the probability of having no A-faults ($y_1 = y_2 = 0$) in a final

batch of n items, given λ, corresponds to

$$p_D(y_1 = 0, y_2 = 0 \,|\, n, \lambda) = \frac{p_D(y_1 = 0, y_2 = 0 \,|\, m = n)\, p_D(m = n \,|\, \lambda)}{p_D(n \,|\, \lambda)}$$

$$= \frac{p_D(y_1 = 0, y_2 = 0 \,|\, m = n)\, p_D(m = n \,|\, \lambda)}{\sum_{i=0}^{n} p_D(y_1 = n - i \,|\, m = i)\, p_D(m = i \,|\, \lambda)}.$$

Finally, the probability that all n items in the final batch are acceptable, given θ and λ, is

$$q(n \,|\, \theta, \lambda) = p_D(y_1 = 0, y_2 = 0 \,|\, n, \lambda)\, p_D(x_0 = n \,|\, m = n, \theta)$$

$$= \frac{p_D(y_1 = 0, y_2 = 0 \,|\, m = n)\, p_D(m = n \,|\, \lambda)}{\sum_{i=0}^{n} p_D(y_1 = n - i \,|\, m = i)\, p_D(m = i \,|\, \lambda)}\, \theta^n,$$

since the only circumstance for an acceptable final batch is having n initial acceptable items ($x_0 = m = n$) and no faulty items included through injection or modification ($y_1 = y_2 = 0$). □ □

Proof of Proposition 2: According to Assumption A3 and Lemma 1, Table 7 presents the loss structure for the defender, given our assumptions, for both of her choices under the two scenarios of interest (all items acceptable, some items faulty). Besides, the probability of such scenarios for a final batch with n items is also reflected, as well as the expected losses for each decision given θ and λ.

Table 7: Defender's Loss Structure - Final Batch of $\boldsymbol{n}$ Items.

D's Decision	All Acceptable	Some Faulty	Exp. Loss
Accept, d_0	0	1	$1 - q(n \,\|\, \theta, \lambda)$
Reject, d_1	c	0	$c\, q(n \,\|\, \theta, \lambda)$
Probability	$q(n \,\|\, \theta, \lambda)$	$1 - q(n \,\|\, \theta, \lambda)$	

Thus, the expected losses of both decisions d_0 (accept) and d_1 (reject) with respect to the unknown item acceptability θ and arrival λ rates are

$$\psi_D(d_0) = 1 - E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right], \qquad \psi_D(d_1) = c\, E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right].$$

The defender's optimal rule is then to accept the batch (d_0) if and only if

$$E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right] \geq \frac{1}{1 + c}. \tag{5}$$

Now, since $p_D(y_1 = 0, y_2 = 0 \,|\, n, \lambda) \in [0, 1]$, the expected value $E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right] = E_{\theta,\lambda}\left[p_D(y_1 = 0, y_2 = 0 \,|\, n, \lambda)\, \theta^n\right]$ is bounded from below by 0 and bounded from above by $E_\theta\left[\theta^n\right]$ which is a decreasing function in n converging to 0 as $n \to \infty$. Thus, $E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right]$ also converges to 0 as $n \to \infty$ and there is a threshold value n_1 such that the optimal choice is to reject the batch (d_1) if $n > n_1$. □ □

Proof of Corollary 1: For an adversary-unaware defender, the acceptance rule would be expression (4) based on $E_\theta\left[\theta^n\right]$ (Proposition 1). An adversary-aware defender would use rule (5) resorting to $E_{\theta,\lambda}\left[q(n \,|\, \theta, \lambda)\right]$ instead (Proposition 2). The result holds since $q(n \,|\, \theta, \lambda) = p_D(y_1 = 0, y_2 = 0 \,|\, n, \lambda)\, \theta^n$ with $p_D(y_1 = 0, y_2 = 0 \,|\, n, \lambda) \leq 1$ being a probability. □ □

Proof of Proposition 3: Under Assumption A4, Table 8 depicts the attacker's loss structure from the defender's perspective. Losses depend on the batch composition and the decisions made by both agents with $x_0 \in \{0, 1, \ldots, m\}$, $y_1 \in \{0, 1, 2, \ldots\}$ and $y_2 = y_2^0 + y_2^1 \in \{0, 1, \ldots, m\}$.

Table 8: Attacker's Loss Structure per Item.

D's Decision	Acceptable	O-Fault	Inj. A-Fault	Mod. A-Fault
Accept, d_0	0	$-h$	$f_1 - g$	$f_2 - g$
Reject, d_1	0	0	f_1	f_2
Amount	$x_0 - y_2^0$	$m - x_0 - y_2^1$	y_1	y_2

According to the loss structure in Table 8, given that the attacker chooses to inject y_1 and modify y_2 items to faulty, his expected losses associated with both defender's decisions are

$$l_A(d_0, y_1, y_2) = -h\,(m - x_0 - E\left[y_2^1\right]) + (f_1 - g)\,y_1 + (f_2 - g)\,y_2, \qquad l_A(d_1, y_1, y_2) = f_1\,y_1 + f_2\,y_2.$$

As he randomly picks the y_2 items among the m original items in the batch (Assumption A3), then $E\left[y_2^1\right] = y_2\,\frac{m - x_0}{m}$, so that

$$l_A(d_0, y_1, y_2) = -h\,(m - x_0)\,(1 - \tfrac{y_2}{m}) + (f_1 - g)\,y_1 + (f_2 - g)\,y_2.$$

Knowing that the original batch size is m, the attacker should select $(y_1^*, y_2^*)(m)$ to minimize his expected loss

$$\begin{aligned}
\psi_A(y_1, y_2 \,|\, m) &= p_A(d_0 \,|\, n = m + y_1) \int \left(\sum_{x_0=0}^{m} l_A(d_0, y_1, y_2)\, p_A(x_0 \,|\, m, \theta) \right) p_A(\theta)\, \mathrm{d}\theta \\
&\quad + (1 - p_A(d_0 \,|\, n = m + y_1))\, l_A(d_1, y_1, y_2) \\
&= y_1\,(f_1 - \gamma(y_1)) + y_2\,(f_2 - \gamma(y_1)) \\
&\quad - \tfrac{h}{g}\left(1 - \tfrac{y_2}{m}\right)\gamma(y_1) \int \left(\sum_{x_0=0}^{m} \binom{m}{x_0} \theta^{x_0}\,(1 - \theta)^{m - x_0}\,(m - x_0) \right) p_A(\theta)\, \mathrm{d}\theta \\
&= y_1\,(f_1 - \gamma(y_1)) + y_2 \left(f_2 - \left(1 - \tfrac{h}{g}\,(1 - E_A[\theta])\right)\gamma(y_1) \right) - m\,\tfrac{h}{g}\,(1 - E_A[\theta])\,\gamma(y_1).
\end{aligned}$$

For the additional conclusions (i) and (ii) observe that:

(i) If $p_A(d_0 \,|\, n) \leq \frac{f_1 - \varepsilon}{g}$ for all $n \geq n_0$, so that the attacker's expected loss per injected A-fault $f_1 - p_A(d_0 \,|\, n)\,g$ is uniformly bounded from below by ε for large enough batch sizes n, then $0 \leq \gamma(y_1) \leq f_1 - \varepsilon$ for all $y_1 \geq n_0 - m$. As a result, since we can rewrite the attacker's expected loss as

$$\psi_A(y_1, y_2 \,|\, m) = y_1\,(f_1 - \gamma(y_1)) + y_2\,(f_2 \pm k_1\,\gamma(y_1)) - k_2\,\gamma(y_1)$$

with $k_1, k_2 \geq 0$ (note that $1 - \frac{h}{g}\,(1 - E_A[\theta])$ might be lower than 0 if $h > g$), it holds that

$$\psi_A(y_1, y_2 \,|\, m) \geq y_1\,\varepsilon + y_2\,(f_2 - k_1\,(f_1 - \varepsilon)) - k_2\,(f_1 - \varepsilon)$$

for all $y_1 \geq n_0 - m$. Being $0 \leq y_2 \leq m$, we conclude that the optimal amount y_1^* of injected items must be bounded.

(ii) The attacker's expected loss (3) is linear in $y_2 \in \{0, 1, \ldots, m\}$ with slope $f_2 - \left(1 - \frac{h}{g}(1 - E_A[\theta])\right)\gamma(y_1)$, reflecting whether it is worth for the attacker to modify items based on its sign. Then, whichever the actual value of y_1^*, the optimal value of y_2 is $y_2^* = 0$ when $f_2 > \left(1 - \frac{h}{g}(1 - E_A[\theta])\right)\gamma(y_1^*)$ (positive slope), and $y_2^* = m$ when $f_2 < \left(1 - \frac{h}{g}(1 - E_A[\theta])\right)\gamma(y_1^*)$ (negative slope). A null slope makes optimal all $y_2 \in \{0, 1, \ldots, m\}$. □

□

Proof of Proposition 4: For each atomic element $\omega \in \Omega$, choose the corresponding parameters and probabilities $\left(G^\omega, H^\omega, F_1^\omega, F_2^\omega, P_A^\omega(d_0 \,|\, n), P_A^\omega(\theta)\right)$. Proposition 3 is then applicable and we may find the attacker's optimal policy $(y_1^*, y_2^*)^\omega(m)$ minimizing in y_1 and y_2 his expected loss

$$\Psi_A^\omega(y_1, y_2 \,|\, m) = y_1\left(F_1^\omega - \Gamma^\omega(y_1)\right) + y_2\left(F_2^\omega - \left(1 - \tfrac{H^\omega}{G^\omega}\left(1 - E_A^{\mathcal{P},\omega}[\theta]\right)\right)\Gamma^\omega(y_1)\right) - m\,\tfrac{H^\omega}{G^\omega}\left(1 - E_A^{\mathcal{P},\omega}[\theta]\right)\Gamma^\omega(y_1).$$

The set of optimal solutions $(y_1^*, y_2^*)^\omega(m)$ for each atomic element $\omega \in \Omega$, together with the probability space $(\Omega, \mathcal{A}, \mathcal{P})$, thus defines the attacker's random optimal policy $(Y_1^*, Y_2^*)(m)$. Now, if $P_A\left(F_2 = \left(1 - \frac{H}{G}\left(1 - E_A^{\mathcal{P}}[\theta]\right)\right)\Gamma(Y_1^*)\right) = 0$, conclusion (ii) in Proposition 3 guarantees that $y_2^{*,\omega} \in \{0, m\}$ almost surely for each atomic element $\omega \in \Omega$. Hence, $Y_2^* \in \{0, m\}$ almost surely. □ □

Acknowledgements

The work of DRI is supported by the AXA-ICMAT Chair on Adversarial Risk Analysis and the Spanish Ministry of Science program MTM2017-86875-C3-1-R AEI/FEDER, UE. FR also acknowledges the contribution of the Community of Madrid through its Chair of Excellence programme. JGO's research has been financed by the Spanish Ministry of Economy and Competitiveness under FPI SO grant agreement BES-2015-072892. This work has also been partially supported by the Spanish Ministry of Economy and Competitiveness through the "Severo Ochoa" Program for Centers of Excellence in R&D (SEV-2015-0554) and project MTM2015-72907-EXP, as well as by the US National Science Foundation through grant DMS-163851 and the BBVA Foundation project "Adversarial Machine Learning: Methods, Computations and Applications to Malware, Fake News and Autonomous Vehicles (AMALFI)".

References

Banks DL, Ríos J, Ríos Insua D (2015) *Adversarial Risk Analysis* (CRC Press, Boca Raton, FL).

Boros E, Fedzhora L, Kantor PB, Saeger K, Stroud P (2009) A large-scale linear programming model for finding optimal container inspection strategies. *Naval Res. Logist.* 56(5):404–420.

Caflisch RE (1998) Monte Carlo and quasi-Monte Carlo methods. *Acta Numer.* 7:1–49.

Chaloner KM, Duncan GT (1983) Assessment of a beta prior distribution: PM elicitation. *J. Royal Stat. Soc. D Stat.* 32(1-2):174–180.

Chung KL (2001) *A Course in Probability Theory* (Academic Press, San Diego, CA).

Cooke RM (1991) *Experts in Uncertainty: Opinion and Subjective Probability in Science* (Oxford University Press, New York, NY).

Cormack GV, Lynam TR (2007) Online supervised spam filter evaluation. *ACM Trans. Inf. Syst.* 25(3):11.

Dorazio RM (2009) On selecting a prior for the precision parameter of Dirichlet process mixture models. *J. Stat. Plan. Inference* 139(9):3384–3390.

Dreiding RA, McLay LA (2013) An integrated model for screening cargo containers. *Eur. J. Oper. Res.* 230(1):181–189.

Duan Z, Chen P, Sanchez F, Dong Y, Stephenson M, Baker JM (2012) Detecting spam zombies by monitoring outgoing messages. *IEEE Trans. Dependable Secure Comput.* 9(2):198–210.

Ferguson TS (1973) A Bayesian analysis of some nonparametric problems. *Ann. Stat.* 1(2):209–230.

French S, Ríos Insua D (2000) *Kendall's Library of Statistics 9: Statistical Decision Theory* (Wiley, New York, NY).

Gaukler GM, Li C, Ding Y, Chirayath SS (2012) Detecting nuclear materials smuggling: Performance evaluation of container inspection policies. *Risk Anal.* 32(3):531–554.

González-Ortega J, Ríos Insua D, Ruggeri F, Soyer R (2019) Adversarial hypothesis testing in presence of adversaries. *Am. Stat.*, ePub ahead of print June 14, https://doi.org/10.1080/00031305.2019.1630001.

Haphuriwat N, Bier VM, Willis HH (2011) Deterring the smuggling of nuclear weapons in container freight through detection and retaliation. *Decis. Anal.* 8(2):88–102.

Hargreaves-Heap SP, Varoufakis Y (2004) *Game Theory: A Critical Introduction* (Routledge, New York, NY).

Hausken K, Bier VM (2011) Defending against multiple different attackers. *Eur. J. Oper. Res.* 211(2):370-384.

Hausken K, Levitin G (1986) Review of systems defense and attack models. *Int. J. Performability Eng.* 8(4):355–366.

Hausken K, Zhuang J (2011) Governments' and terrorists' defense and attack in a T-period game. *Decis. Anal.* 8(1):46–70.

Heydari A, ali Tavakoli M, Salim N, Heydari Z (2015) Detection of review spam: A survey. *Expert Syst. Appl.* 42(7):3634–3642.

Hooi B, Shah N, Beutel A, Günnemann S, Akoglu L, Kumar M, Makhija D, Faloutsos C (2016) BIRDNEST: Bayesian inference for ratings-fraud detection. Venkatasubramanian SC, Meira W, eds. *Proc. 16th SIAM Int. Conf. Data Min.* (Society for Industrial and Applied Mathematics, Philadelphia, PA), 495–503.

Jagielski M, Oprea A, Biggio B, Liu C, Nita-Rotaru C, Li B (2018) Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. Parno B, Kruegel C, eds. *Proc. 39th IEEE Symp. Secur. Priv.* (Institute of Electrical and Electronics Engineers Computer Society, Los Alamitos, CA), 19–35.

Jiang AX, Procaccia AD, Qian Y, Shah N, Tambe M (2013) Defender (mis)coordination in security games. Rossi F, ed. *Proc. 23rd Int. Jt. Conf. Artif. Intell.* (Association for the Advancement of Artificial Intelligence Press, Palo Alto, CA), 220–226.

Jindal N, Liu B (2008) Opinion spam and analysis. Broder A, Chakrabarti S, eds. *Proc. 2008 Int. Conf. Web Search and Data Min.* (Association for Computing Machinery, New York, NY), 219–230.

Kantarcioglu M, Xi B, Clifton C (2011) Classifier evaluation and attribute selection against active adversaries. *Data Min. Knowl. Discov.* 22(1-2):291–335.

Lau RYK, Liao SY, Kwok RCW, Xu K, Xia Y, Li Y (2012) Text mining and probabilistic language modeling for online review spam detection. *ACM Trans. Manag. Inf. Syst.* 2(4):25/1-30.

Li FH, Huang M, Yang Y, Zhu X (2011) Learning to identify review spam. Walsh T, ed. *Proc. 22nd Int. Jt. Conf. Artif. Intell.* (Association for the Advancement of Artificial Intelligence Press, Palo Alto, CA), 2488–2493.

Lim EP, Nguyen VA, Jindal N, Liu B, Lauw HW (2010) Detecting product review spammers using rating behaviors. Koudas N, Jones G, Wu X, Collins-Thompson K, An A, eds. *Proc. 19th ACM Int. Conf. Inf. and Knowl. Manag.* (Association for Computing Machinery, New York, NY), 939–948.

Lindley DV, Singpurwalla ND (1991) On the evidence needed to reach agreed action between adversaries, with application to acceptance sampling. *J. Am. Stat. Assoc.* 86(416):933–937.

Luckner M, Gad M, Sobkowiak P (2014) Stable web spam detection using features based on lexical items. *Comput. Secur.* 46:79–93.

McLay LA, Rothschild C, Guikema S (2012) Robust adversarial risk analysis: A level-k approach. *Decis. Anal.* 9(1):41–54.

Merrick JRW, Albert LA (2018) Expert judgment based nuclear threat assessment for vessels arriving in the US. Dias LC, Morton A, Quigley J, eds. *Elicitation: The Science and Art of Structuring Judgement* (Springer, Cham, Switzerland), 495–509.

Merrick JRW, Parnell GS (2011) A comparative analysis of PRA and intelligent adversary methods for counterterrorism risk management. *Risk Anal.* 31(9):1488–1510.

O'Hagan A, Buck CE, Daneshkhah A, Eiser JR, Garthwaite PH, Jenkinson DJ, Oakley JE, Rakow T (2006) *Uncertain Judgements: Eliciting Experts' Probabilities* (Wiley, Chichester, UK).

Ramilli M, Prandini M (2009) Comment spam injection made easy. Buford JF, ed. *Proc. 6th IEEE Consum. Commun. and Netw. Conf.* (Institute of Electrical and Electronics Engineers Computer Society, Red Hook, NY), 1–5.

Ríos J, Ríos Insua D (2012) Adversarial risk analysis for counterterrorism modeling. *Risk Anal.* 32(5):894–915.

Ríos Insua D, Banks DL, Ríos J, González-Ortega J (2020) Adversarial risk analysis as a structured expert judgement decomposition tool. French S, Nane T, Hanea A, Bedford T, eds. *Expert Judgement in Risk and Decision Analysis* (Springer, Cham, Switzerland), forthcoming.

Ríos Insua D, Ríos J, Banks DL (2009) Adversarial risk analysis. *J. Am. Stat. Assoc.* 104(486):841–854.

Robert CP, Casella G (2013) *Monte Carlo Statistical Methods* (Springer, New York, NY).

Ruan X, Wu Z, Wang H, Jajodia S (2015) Profiling online social behaviors for compromised account detection. *IEEE Trans. Inf. Forensics and Secur.* 11(1):176–187.

Schilling EG, Neubauer DV (2009) (CRC Press, Boca Raton, FL).

Sculley D, Wachman GM (2007) Relaxed online SVMs for spam filtering. Clarke CLA, Fuhr N, Kando N, eds. *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. and Dev. Inf. Retr.* (Association for Computing Machinery, New York, NY), 415–422.

Shachter RD (1986) Evaluating influence diagrams. *Oper. Res.* 34(6):871–882.

Shapiro D, Shi X, Zillante A (2014) Level-k reasoning in a generalized beauty contest. *Games Econ. Behav.* 86:308–329.

Schlenker A, Brown M, Sinha A, Tambe M, Mehta R (2016) Get me to my GATE on time: Efficiently solving general-sum Bayesian threat screening games. Kaminka GA, Fox M, Bouquet P, Hüllermeier E, Dignum V, eds. *Proc. 22nd Eur. Conf. Artif. Intell.* (IOS Press, Amsterdam, Netherlands), 1476–1484.

Stidham S (1985) Optimal control of admission to a queueing system. *IEEE Trans. Autom. Control* 30(8):705–713.

Tapiero CS (1995) Acceptance sampling in a producer-supplier conflicting environment: Risk neutral case. *Appl. Stoch. Model. Data Anal.* 11(1):3–12.

Tartakovsky AG, Nikiforov IV, Basseville M (2014) *Sequential Analysis: Hypothesis Testing and Changepoint Detection* (CRC Press, Boca Raton, FL).

Tondi B, Merhav N, Barni M (2019) Detection games under fully active adversaries. *Entropy* 21(1):23.

Xie S, Wang G, Lin S, Yu PS (2012) Review spam detection via temporal pattern discovery. Agarwal D, Pei J, eds. *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. and Data Min.* (Association for Computing Machinery, New York, NY), 823–831.

Ye N, Newman C, Farley T (2018) A system-fault-risk framework for cyber attack classification. *Inf. Knowl. Syst. Manag.* 5(2):135–151.

Young HP (2004) *Strategic Learning and its Limits* (Oxford University Press, Oxford, NY).

Yu S, Vorobeychik Y, Alfeld S (2018) Adversarial classification on social networks. Dastani M, Sukthankar G, eds. *Proc. 17th Int. Conf. Auton. Agents and MultiAgent Syst.* (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 211–219.

Zhang Y, Wang S, Phillips P, Ji G (2014) Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowl.-Based Syst.* 64:22–31.

Zhuang J, Bier VM, Alagoz O (2010) Modeling secrecy and deception in a multiple-period attacker-defender signaling game. *Eur. J. Oper. Res.* 203(2):409–418.

Notation Summary

To facilitate reading, we provide a summary of the main notation.

ARA Adversarial Risk Analysis.

ID Influence Diagram.

BAID Bi-Agent Influence Diagram.

O-faults Outer faults (original faulty items).

A-faults Attacker faults (faulty items introduced by attacker).

$\boldsymbol{d}$ **/** $\boldsymbol{D}$ Defender's choice upon acceptance / related decision node.

$\boldsymbol{m}$ **/** $\boldsymbol{M}$ Original batch size / related chance node.

$\boldsymbol{\theta}$ **/** $\boldsymbol{\Theta}$ Item acceptability rate / related chance node.

$\boldsymbol{x}$ / $\boldsymbol{X}$ Original batch composition / related chance node.

$\boldsymbol{x_0}$ / $\boldsymbol{x_1}$ Number of original acceptable / faulty items.

$\boldsymbol{l_D}$ Defender's loss function and related loss node.

$\boldsymbol{c}$ Defender's (expected) opportunity cost upon rejecting a batch with all items acceptable.

$\boldsymbol{E_\xi}[\cdot]$ Expected value with respect to random variable ξ.

$\boldsymbol{\psi_D}$ Defender's expected loss function.

$\boldsymbol{a}$ / $\boldsymbol{A}$ Attacker's action / related decision node.

$\boldsymbol{y}$ / $\boldsymbol{Y}$ Final batch composition / related chance node.

$\boldsymbol{n}$ / $\boldsymbol{N}$ Final batch size / related chance node.

$\boldsymbol{\lambda}$ / $\boldsymbol{\Lambda}$ Item arrival rate / related chance node.

$\boldsymbol{x}$ / $\boldsymbol{X}$ Original batch composition / related chance node.

$\boldsymbol{l_A}$ Attacker's loss function and related loss node.

$\boldsymbol{y_1}$ Number of faulty items injected by attacker.

$\boldsymbol{y_2}$ Number of original items modified to faulty by attacker.

$\boldsymbol{y_2^0}$ / $\boldsymbol{y_2^1}$ Number of original acceptable / faulty items modified to faulty by attacker.

$\boldsymbol{q(n \mid \theta, \lambda)}$ Probability that all items in a final batch of size n are acceptable.

$\boldsymbol{p_D(\cdot)}$ Probabilities from defender's perspective.

$\boldsymbol{p_A(\cdot)}$ / $\boldsymbol{P_A(\cdot)}$ Probabilities from attacker's perspective / defender's random version.

$\boldsymbol{h}$ / $\boldsymbol{H}$ Attacker's (expected) gain per O-fault / defender's random version.

$\boldsymbol{g}$ / $\boldsymbol{G}$ Attacker's (expected) gain per A-fault / defender's random version.

$\boldsymbol{f_1}$ / $\boldsymbol{F_1}$ Attacker's (expected) cost per injected A-fault / defender's random version.

$\boldsymbol{f_2}$ / $\boldsymbol{F_2}$ Attacker's (expected) cost per item modified to A-fault / defender's random version.

$\boldsymbol{\gamma(y_1)}$ / $\boldsymbol{\Gamma(y_1)}$ Attacker's expected gain per A-fault / defender's random version.

$\boldsymbol{\psi_A}$ / $\boldsymbol{\Psi_A}$ Attacker's expected loss function / defender's random version.

$\boldsymbol{y_1^*}$ / $\boldsymbol{y_2^*}$ Optimal amount of injected / modified items by attacker.

$\boldsymbol{(\Omega, \mathcal{A}, \mathcal{P})}$ Common probability space modeling defender's uncertainty about attacker.

$\boldsymbol{\omega}$ Atomic element of probability space.

$\boldsymbol{J^{\omega}}$ Instance of random parameter J based on atomic element ω.

$\boldsymbol{P_A^{\omega}(\cdot)}$ / $\boldsymbol{E_A^{\mathcal{P},\omega}[\theta]}$ Instance of distribution / expected value of $P_A(\cdot)$ on probability space.

$\boldsymbol{E_A^{\mathcal{P}}[\xi]}$ Random expected value of randomized variable ξ on probability space.

$\boldsymbol{Y_1^*}$ / $\boldsymbol{Y_2^*}$ Distribution over optimal amount of injected / modified items by attacker.

$\boldsymbol{K}$ Number of iterations in Monte Carlo simulation.

$\boldsymbol{\overline{Y}_1}$ Upper bound for amount of injected items by attacker.