Robust Spammer Detection by Nash Reinforcement Learning

Yingtong Dou
Univ. of Illinois at Chicago
ydou5@uic.edu

Guixiang Ma*
Intel Labs
guixiang.ma@intel.com

Philip S. Yu
Univ. of Illinois at Chicago
psyu@uic.edu

Sihong Xie
Lehigh University
xiesihong1@gmail.com

ABSTRACT

Online reviews provide product evaluations for customers to make decisions. Unfortunately, the evaluations can be manipulated using fake reviews ("spams") by professional spammers, who have learned increasingly insidious and powerful spamming strategies by adapting to the deployed detectors. Spamming strategies are hard to capture, as they can be varying quickly along time, different across spammers and target products, and more critically, remained unknown in most cases. Furthermore, most existing detectors focus on detection accuracy, which is not well-aligned with the goal of maintaining the trustworthiness of product evaluations. To address the challenges, we formulate a minimax game where the spammers and spam detectors compete with each other on their practical goals that are not solely based on detection accuracy. Nash equilibria of the game lead to stable detectors that are agnostic to any mixed detection strategies. However, the game has no closed-form solution and is not differentiable to admit the typical gradient-based algorithms. We turn the game into two dependent Markov Decision Processes (MDPs) to allow efficient stochastic optimization based on multi-armed bandit and policy gradient. We experiment on three large review datasets using various state-of-the-art spamming and detection strategies and show that the optimization algorithm can reliably find an equilibrial detector that can robustly and effectively prevent spammers with any mixed spamming strategies from attaining their practical goal. Our code is available at https://github.com/YingtongDou/Nash-Detect.

CCS CONCEPTS

• **Information systems** → **Spam detection**; • **Security and privacy**; • **Theory of computation** → **Adversarial learning**;

KEYWORDS

Spam Detection; Reinforcement Learning; Adversarial Learning

1 INTRODUCTION

Online reviews and ratings contributed by real customers help shape reputations of the businesses and guide customer decision-makings, playing an integrative role in e-commerce and websites such as Amazon [8], Yelp [21], and Google Play [26]. However, monetary incentives therein have also attracted a large number of spammers to hold sway over less informed customers: it is estimated that about 40% of the reviews on Amazon are fake (called "review spams") [33]. To cope with the spams and restore the trustworthiness of online reviews, many detection methods based on texts [13, 24, 39], reviewer behaviors [16, 23, 40], and graphs [10, 14, 20, 27] have been proposed. See Table 1 for some state-of-the-art.

Table 1: Comparison of the proposed Nash-Detect and prior work. "Attack Agnostic" indicates a method does not assume a fixed attack. "Practical Goal" means at a business metric is considered.
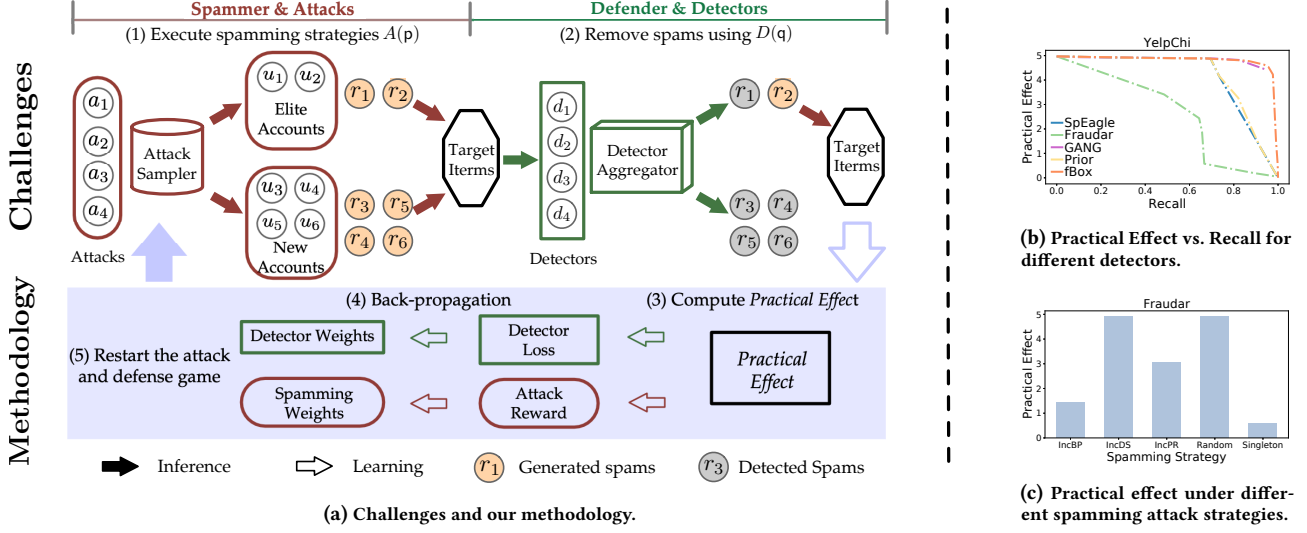
	Business		Fraud Detection					Proposed
	[17]	[22]	[45]	[10]	[42]	[12]	[3]	Nash-Detect
Practical Goal	✓	✓	✓		✓	✓		✓
Attack Agnostic				✓			✓	✓

We note two drawbacks of existing detectors. **1)** Most detectors assume spams are generated by spammers[1] with the same mindset and can detect the spams by relying on the assumed spamming strategy. In the real world, there are multiple groups of spammers who have different goals, targets, resources, and strategies. One spammer may want to promote a new business while another aims to demote a popular brand's competitors [22]. The wide spectrum of detection signals published so far serves as strong evidence that multiple spamming strategies co-exist and no single detector can stop the spams. **2)** Professional spammers are more persistent and committed, and can research the latest detection techniques from published papers [24], third-party detection websites with detailed detection manuals [29], and labeled spams released by the review website [27]. The spammer can integrate these resources and learn to infiltrate the deployed detectors.

The above drawbacks are partially addressed by a diverse set of spamming modes considered in the prior work [3, 5, 10, 12, 37, 41]. Nonetheless, the prior work aimed at high detection recall, AUC, or top-k accuracy. Without considering practical spamming goals, the accuracy-oriented detectors may not directly counteract the unknown spamming goal. For example, as shown in Figure 1a, when the human inspectors can only screen the most suspicious reviews identified by the detector, one can achieve high top-k accuracy by reporting k easy-to-catch but insignificant spams, while letting go the more significant ones that are actually manipulating the review system [46]. In other words, the accuracy-oriented detectors and

---

[1] A "spammer" refers to a physical person or entity in the real world that spams, rather than a "spamming account" that is set up by a spammer on the online review system.
*This work was done when the author was at University of Illinois at Chicago.

(a) Challenges and our methodology.

(b) Practical Effect vs. Recall for different detectors.

(c) Practical effect under different spamming attack strategies.

Figure 1: (a): A vulnerable spam detection pipeline, with steps numbered as (1), (2), etc. Accuracy-based detectors can be misled to detect numerous insignificant spams from new accounts, leaving behind the more manipulative elite spams. We define a zero-sum game to find a robust defender against unknown and evolving spamming strategies $A(p)$. (b): The Practical Effect vs. Recall of individual detectors (shown in legend) against a mixed spamming strategy. The curve is obtained by sweeping the detection thresholds. For most detectors, the attack could attain high practical effects even with high detection recall scores. (c): For a fixed spam detector (Fraudar), a spammer can choose the best out of five attack strategies to maximize the practical effect.

the limited human screening resources create a unique vulnerability in the spam detection pipeline.

We reconsider spam detection by posing the following questions. **1)** What is the ultimate intention of the spammers? What is considered as a successful spamming campaign? From a marketing perspective, a spammer aims to maximize reputation manipulations and get paid by dishonest businesses. We distinguish the manipulative effect by elite accounts from that by regular accounts, and adopt marketing research results [17, 21] to formulate a practical quantitative spamming goal. To show the vulnerability mentioned above, we create spamming attacks against various state-of-the-art spam detectors. The attacks effectively attain the practical spamming goals while saturating the detectors to rather high detection recall scores (Figure 1b). **2)** What if there are multiple evolving spamming strategies? Our experiment results in Figure 1c show that a single fixed detector will always fail to detect some spams when a spammer has knowledge about the detector to carefully craft evasive spams. This implies a black-box attack: if a fixed detector is adopted, a spammer can probe the detection system by continuously changing the spam generation strategy until success. Further, if multiple spamming strategies co-exist, either because a spammer diversifies its strategy portfolio, or because multiple spammers are adopting different strategies simultaneously, likely, some strategies will successfully infiltrate the fixed detector.

To answer the above questions, we propose Nash-Detect, an attack-agnostic spam detector that is a Nash equilibrium of a minimax game and can tame adaptive and evolving spamming. **1)** We define practical goals for both spammers and detectors from the business perspective. We calibrate the detector via a cost-sensitive loss function focusing on practical spamming effect measured by revenue manipulations rather than detection accuracy. By minimizing this loss function, the detector can be trained to go after a small number of more manipulative spams, while letting go spams that

are less significant. **2)** To reveal the vulnerability, we design strong *base* spamming strategies against specific state-of-the-art detectors. The base strategies will be combined into *mixed* strategies to explore even stronger attacks. Compared to neural network-based attacks [31, 47], our spamming strategies rely on unlabeled data and have no parameters to train. A spamming strategy based on parameter learning on a labeled set will be less general and can overfit the training data. **3)** We formulate a minimax game, where the spammer will vary its mixed spamming strategies to maximize the practical spamming effect, while the detector will reconfigure its detection strategy to minimize the updated practical spamming effect. The two strategies will hopefully evolve to an equilibrium consisting of a robust detection strategy. To ensure computation tractability, we propose a reinforcement learning approach to find an equilibrium of the minimax game in multiple episodes of fictitious plays of the game. Each episode has two phases. In the inference phase, the spammer samples a base spamming strategy according to the current mixed strategy, and the detector runs its current detection strategy. This step will evaluate the practical spamming effect under the current two strategies. In the learning phase, we update the two strategies by backpropagating the practical effect to the strategy parameters, so that both players can move in opposite directions to maximize and minimize the effect. After multiple rounds of fictitious play, the converged detection strategy will be robust to any spamming strategy, including the worst one that the spammer can synthesize using the base spamming strategies.

Experiments show that Nash-Detect can find the best configurations of detectors which always have better defending performance than the worst cases the spammer can synthesize. Nash-Detect also exhibits great stability under various scenarios.

2 PRELIMINARIES AND CHALLENGES

The data of a review system can be represented by a bipartite graph $\mathcal{G}$ consisting of a set of accounts $\mathcal{U} = \{u_1, \ldots, u_m\}$, a set of products $\mathcal{V} = \{v_1, \ldots, v_n\}$ where m is the number of accounts and n is the number of products, and a set of edges $\mathcal{R} = \{r_{ij} : i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\}$. Account u_i posts a review to product v_j if and only if there is $r_{ij} \in \mathcal{R}$ and we use $\mathcal{R}$ and r_{ij} to denote the set of reviews and a specific review. Node v_i (u_j, edge r_{ij}, resp.) can have various attributes, represented by the vector $\mathbf{x}_i$ ($\mathbf{x}_j$, $\mathbf{x}_{ij}$, resp.), describing account and product profiles, and review metadata (e.g., posting time, ratings, and the related texts or image contents). The review system evaluates a node $v \in \mathcal{V}$ with a reputation score $s(v)$, according to the product ratings, rankings, and credibility of the review account . Users may rely on $s(v)$ to decide whether to commit trust, attention, and money to v. For example, elite accounts have higher credibility and more likely to be trusted by users [8, 22].

A physical spammer is a person who registers and controls a set of accounts (the "Sybils") $\mathcal{U}_S = \mathcal{U}_E \cup \mathcal{U}_N \subset \mathcal{U}$ [45], where $\mathcal{U}_E$ is a set of elite accounts carrying more weights in their ratings, and $\mathcal{U}_N$ is a set of new accounts that the spammer register at any time. When committing a spamming campaign, the spammer creates a set of spams, represented as a set of new edges $\mathcal{R}_S$ emitting from some accounts in $\mathcal{U}_S$ towards the target nodes $\mathcal{V}_T \subset \mathcal{V}$ and some non-targets $\mathcal{V} \setminus \mathcal{V}_T$ (for camouflaging) within a time window. For simplicity, we assume the spams are posted towards the target nodes only. The edges $\mathcal{R}_S$ and the attributes of the edges are decided by a spamming strategy with the goal of increasing or decreasing $s(v)$ for $v \in \mathcal{V}_T$. A spammer can adopt multiple spamming strategies simultaneously. We assume there are K *base attack strategies* a $= [a_1, \ldots, a_K]$ (specified in Section 3.4). Given the current review system status and a spamming goal, a spamming strategy decides when to use what accounts to post reviews with what ratings to which targets. When multiple strategies are used at the same time, the mixed attack strategy is denoted by $A(\mathbf{p}) = \mathbb{E}_{k \sim \mathbf{p}}[a_k] = \sum_{k=1}^{K}(p_k a_k)$, where $\mathbf{p} = [p_1, \ldots, p_K]$ is the mixture parameter with $\sum p_k = 1$.[2]

No single detector can tackle all spamming strategies, and a myriad of detectors based on different spamming assumptions have been proposed in [1, 10, 24, 27, 40]. See the related work for more details. Ensembling a diverse set of detectors has been proven useful [27, 28]. We assume there is an array of L base detectors d $= [d_1, \ldots, d_L]$ and each detector outputs a number in $[0, 1]$, with a larger value denoting more suspiciousness. The detectors are given various importance q $= [q_1, \ldots, q_L]$, so that the detector in effect is $D(\mathbf{q}) = \sum_{l=1}^{L}(q_l d_l) = \mathbf{q}^\top \mathbf{d}$.

Challenges and the threat model. While the existing detectors are designed to defend against one or multiple spamming strategies, the underlying *true* spamming strategy has never been fully revealed. To make the research problem tractable, we are not trying to discover the *true* spamming strategy, but instead addressing the strongest spamming strategy that the spammers can engineer given due knowledge and resources [36]. This goal is more practically meaningful: professional spammers can access the details of spam detectors through published papers [24], reverse engineering from

---

[2]Rigorously, $A(\mathbf{p})$ is not a weighted sum of numbers, since a_k are not numbers but a spam-generating function. Instead, think of $A(\mathbf{p})$ as a weighted sum of functions.

labeled spams [27], and detection explanations [29], and update their spamming strategies to evade a fixed detector. Indeed, previous work [5, 10, 30, 42] have considered spams in camouflage for evading a *fixed* detector. Even if the fixed detector remains unknown to the spammer, there are only a handful of salient detection signals, leading to a relatively small space of detectors (all q over the set of known salient detectors). A spammer can continue to probe the current detector until it finds something vulnerable and then crafts a strong spamming strategy.

The second challenge is that, spam detection is treated as a binary classification problem and evaluated using accuracy [10], recall [23], precision, AUC, or nDCG [27, 38]. These metrics are not well aligned with the practical spamming goal, namely, to perturb product reputation and revenue in a relatively short period. We observed that the amount of perturbation is only weakly correlated with the detection recall score. See Figures 1b and 2. In reality, human spam screening and removals are limited to the top-k suspicious spams, and the revenue manipulation is not entirely due to the *number* of missed spams (false negatives). We identify the following vulnerability: massive obvious spams can easily be detected, saturate the human screening capacity, and pose a deceivingly high detection accuracy, while a small number of spams posted by elite accounts are missed by the top-k screening and remain to manipulate the target reputation and revenue significantly [21]. We term such detectors as "accuracy-focusing detectors", as they optimize accuracy and other related metrics (F1-score, recall rate, etc.)

3 METHODOLOGY

3.1 Turning Reviews into Business Revenues

A high (or low) rating will contribute to the growth (or decrease) in the sales of a product [8, 22], and a one-star increase in average rating contributes to a 5 to 9 percent increase in revenues [17, 21]. More specifically, the unit increase in revenue can be attributed to two different types of reviews: those posted by regular accounts ("regular reviews") and those by elite accounts ("elite reviews"). Different from the regular accounts, an elite account is recognized by the review system if the account has frequently posted high-quality reviews. The elite reviews are more influential in product revenue [21], as they count more in product evaluation in the review system and are more frequently presented to the customers than the regular reviews.

Formally, let $\mathcal{R}$ ($\mathcal{R}_E$, resp.) be the set of all (elite, resp.) reviews before a spamming attack, and let $\mathcal{R}(v)$ ($\mathcal{R}_E(v)$, resp.) be the (elite, resp.) reviews posted towards product v. We adopt the *revenue estimation function* $f(v; \mathcal{R})$ from [21] to measure the influence of the reviews on the revenue of product v:

$$f(v; \mathcal{R}) = \beta_0 \times \text{RI}(v; \mathcal{R}) + \beta_1 \times \text{ERI}(v; \mathcal{R}_E(v)) + \alpha$$
$$= \beta_0 \times (g(\mathcal{R}(v)) - g(\mathcal{R})) + \beta_1 \times g(\mathcal{R}_E(v)) + \alpha, \quad (1)$$

where the function $g(\cdot)$ computes the average rating from the given set of reviews. $\text{RI}(v; \mathcal{R}) \triangleq g(\mathcal{R}(v)) - g(\mathcal{R})$ is the **R**eview **I**nfluence derived from how much the average rating of v is better or worse than the overall average rating. $\text{ERI}(v; \mathcal{R}_E(v)) \triangleq g(\mathcal{R}_E(v))$ is the **E**lite **R**eview **I**nfluence due to the average rating contributed solely by the elite reviews for v. β_0 and β_1 are coefficients of the two

influences, and α is the baseline revenue of all items[3]. Although these coefficients are estimated using Yelp's data and may not be applicable to other review systems, the underpinning of the paper can be applied to other systems such as Amazon so long as this estimation can be done.

3.2 Practical Spamming and Detection Goals

While the spammers aim to manipulate (promote or demote) the targets' revenues, the spam detector aims to tame such manipulations. In the following, assuming a spammer wants to promote the given targets' revenue, we define the goals of the spammer and the defender (demotions are discussed at the end of Section 3). We denote a spamming strategy by $A(\mathrm{p})$ with parameters p, and the detector with detection strategy q by $D(\mathrm{q})$. The letter v is designated to a target product.

A practical spamming goal. Let $\mathcal{R}(\mathrm{p})$ be the spams posted using the mixed spamming strategy $A(\mathrm{p})$, and $\mathcal{R}(\mathrm{p}, \mathrm{q})$ be the false-negative spams that remain after $\mathcal{R}(\mathrm{p})$ is purged by the detector $D(\mathrm{q})$ [4]. Based on the revenue f defined in Eq. (1), a metric to evaluate the Practical Effect (PE) of spamming using $A(\mathrm{p})$ on v against the detection of $D(\mathrm{q})$ is:

$$
\begin{aligned}
\mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q}) &= f(v; \mathcal{R}(\mathrm{p}, \mathrm{q})) - f(v; \mathcal{R}) \\
&= \beta_0 \times \Delta\,\mathrm{RI}(v) + \beta_1 \times \Delta\,\mathrm{ERI}(v),
\end{aligned}
\tag{2}
$$

which is the difference in the revenues of v after and before the spamming and the detection.

$$
\Delta\,\mathrm{RI}(v) = \mathrm{RI}(v; \mathcal{R}(\mathrm{p}, \mathrm{q})) - \mathrm{RI}(v; \mathcal{R}) = g(\mathcal{R}(v; \mathrm{p}, \mathrm{q})) - g(\mathcal{R}(v))
$$
$$
\Delta\,\mathrm{ERI}(v) = \mathrm{ERI}(v; \mathcal{R}_E(\mathrm{p}, \mathrm{q})) - \mathrm{ERI}(v; \mathcal{R}_E) = g(\mathcal{R}_E(v; \mathrm{p}, \mathrm{q})) - g(\mathcal{R}_E(v))
$$

are the change in influences due to the missed spams.

$\mathrm{PE}(v)$ can be negative when the spams and the detector bring v's rating down. This can happen when the human inspectors delete some organic reviews with high ratings from v. We assume that the inspectors will screen the top k (k is typically small) detected reviews carefully so that there is no genuine review deleted. A drop in revenue can also be caused by a spamming strategy that posts negative reviews for the camouflage purpose. We consider such more complex strategies in future work.

> **Spamming goal:** $\quad \max_{\mathrm{p}} \quad \max\{0, \mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q})\}.$ $\quad$ (3)

A practical detection goal. The practical detection goal should be minimizing $\max\{0, \mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q})\}$ and make sure the resulting detection strategy q will not entirely focus on detection accuracy but will suppress the spamming promotion.

In the following, we define a back-propagation algorithm for detection strategy learning. According to Eq. 2, $\max\{0, \mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q})\}$ summarized the effect of the false-negative spams $\mathcal{R}(\mathrm{p}, \mathrm{q})$. To guide the detection strategy q, we first back-propagate (or attribute) $\max\{0, \mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q})\}$ to individual false-negative spams, and the attributed costs are further back-propagated to the current detection strategy q that lead to the false negatives. From the spammer's

---

perspective, elite spamming reviews are more influential. From the detector's perspective, a missed elite spamming review leads to a larger amount of revenue manipulation that a missed regular spamming review. Based on cost-sensitive learning [7], we turn the spamming effect $\max\{0, \mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q})\}$ into detection costs according to different detection outcomes: the detection costs of true positives and true negatives are 0 ($C_{\mathrm{TP}} = C_{\mathrm{TN}} = 0$); the false positives will be handled by human screening and will cause zero effect on the product revenue ($C_{\mathrm{FP}} = 0$); a false negative will not contribute to the promotion of product v if $\mathrm{PE}(v) \le 0$ unless otherwise it will contribute to the following cost through $\Delta\,\mathrm{RI}(v)$ and $\Delta\,\mathrm{ERI}(v)$:

$$
C_{\mathrm{FN}}(v, r) = \frac{\beta_0 \Delta\,\mathrm{RI}(v)}{Z_1} + \mathbb{1}_{r \in \mathcal{R}_E(\mathrm{p}, \mathrm{q})} \left[\frac{\beta_1 \Delta\,\mathrm{ERI}(v)}{Z_2} \right],
\tag{4}
$$

where Z_1 and Z_2 are respectively the amount of non-elite and elite spams posted towards v. $\mathbb{1}$ is the indicator function. Based on the analysis, the detection goal is defined as:

> **Detection goal:** $\min_{\mathrm{q}} \quad \mathcal{L}(\mathrm{q}) = \frac{1}{|\mathcal{R}(\mathrm{p}, \mathrm{q})|} \sum_{r \text{ is FN}} C_{\mathrm{FN}}(v, r),$ $\quad$ (5)

where v is the target product that r was posted to. $C_{\mathrm{FN}}(v, r)$ implicitly depends on the strategy q, through the ranking of reviews by the detector $D(\mathrm{q})$ and the screening of the top k reviews. To facilitate the optimization of q, we define the following *cost-sensitive* surrogate detection loss function:

$$
\mathcal{L}_{\mathrm{q}} = \frac{1}{|\mathcal{R}(\mathrm{p}, \mathrm{q})|} \sum_{r \text{ is FN}} -C_{\mathrm{FN}}(v, r) \log P(y = 1 | r; \mathrm{q}),
\tag{6}
$$

where $y \in \{0, 1\}$ is the label of r ($y = 1$ if and only if r is spam). $P(y = 1 | r; \mathrm{q})$ is the probability of r being a spam predicted by $D(\mathrm{q})$:

$$
P(y = 1 | r; \mathrm{q}) = \sigma(\mathrm{q}^\top \mathrm{d}(r)), \quad \mathrm{d}(r) = [d_1(r), \dots, d_L(r)],
\tag{7}
$$

where σ represents the sigmoid function, the surrogate loss says to reduce the cost $C_{\mathrm{FN}}(r, v)$, the detector should output a large $P(y = 1 | r; \mathrm{q})$ so that the spam r can be pushed into the top k suspicious reviews.

3.3 Minimax Game and Optimization

The spammer and the detector's goals are now well-aligned: the spammer aims at promoting the revenues of the targets $v \in \mathcal{V}_T$ and the detector wants to suppress such promotion. They will play the following zero-sum game over the practical spamming effect $\max\{0, \mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q})\}$:

$$
\min_{\mathrm{q}} \max_{\mathrm{p}} \sum_{v \in \mathcal{V}_T} \max\{0, \mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q})\}.
\tag{8}
$$

Solving the above game will lead to a detector that can withstand any mixing spamming strategies weights $[a_1, \dots, a_K]$. In particular, we aim at a robust detector, parameterized by q^*, that will minimize practical spamming effects caused by any spamming strategies $A(\mathrm{p})$. One limitation of q^* is that, during test time, a spammer may use a pure strategy not considered in the pure strategies $[a_1, \dots, a_K]$ that $D(\mathrm{q}^*)$ was trained on. It is thus essential to include representative spamming strategies during the training of q. Exhausting all spamming strategies is out of the scope of this work, and in

---

[3] $\beta_0 = 0.035$, $\beta_1 = 0.036$, and $\alpha = 1$ based on the empirical findings in [21]. $\beta_1 g(\mathcal{R}_E(v))$ is much larger than $\beta_0(g(\mathcal{R}(v)) - g(\mathcal{R}))$ since β_0 multiplies a difference.
[4] Possibly with human screening of the detected reviews to further reduce false positives

Section 3.4, we specify evasive attacks against mainstream detectors. In the sequel, "spammer" can refer to the fictitious spammers during training or the real spammer during test time.

The objective function is not differentiable, as the calculation of PE (shown in Eq. (2)) are based on nonlinear operators such as adding reviews to $\mathcal{R}$ using the spamming strategy $A(p)$, ranking the reviews in $\mathcal{R}(p)$ based on $P(y = 1|r; q)$ (which depends on the detectors $[d_1, \ldots, d_L]$), the removal of spams from the top k suspicious ones, and the consideration of elite accounts. We note that such non-differentiability is a necessary evil in robust spam detection: diverse spamming and detection strategies help explore the limit of both players, while many state-of-the-art spam attacks and detectors are non-differentiable, non-continuous [42], stochastic [10]. Therefore, gradient-based optimization methods are not applicable, and we propose a multi-agent non-cooperative reinforcement learning approach [11, 19, 34] and use stochastic optimization based on Monte-Carlo to address the challenges. Regarding the spammer and the detector as two competing agents who play the attack and defense game in Eq. (8). The experiences of detector and spammer will be obtained from a T-step roll-out of two *dependent* Markov Decision Processes (MDPs) [32]: for each episode (indexed by $t = 1, \ldots, H$) of the H episodes of game playing, the spammer will attack the review system using the current mixed strategy $A(p^{(t)}) = \sum_{k=1}^{K}(p_k^{(t)} a_k)$ and the detector will respond with the mixed detection strategy $D(q^{(t)}) = \sum_{l=1}^{L}(q_l^{(t)} d_l)$. Both mixing parameters p and q will be updated as follows.

An MDP for the spammer. We adopt the multi-armed bandit formulation [4] so that the spammer has no *explicit* state representation of the review system but acts based on the practical spamming effects. To be precise, the spammer maintains the distribution p over the *base* spamming strategies $a_1, \ldots, a_K$ as a policy. In episode t, for each target v, the spammer samples an action a_k with $k \sim \text{Multinominal}(p^{(t)})$ to execute a spamming attack (via posting a fake review to v using an account selected by a_k). The attack on v by a_k is denoted by $a_k \rightsquigarrow v$. At the end of the episode, the detector uses the mixed strategy $D(q^{(t)})$ to remove some posted spams, and the final $PE(v; \mathcal{R}, p^{(t)}, q^{(t)})$ is calculated as in Eq. (2). The reward for the spamming strategy a_k is the portion of PE in this episode due to the spams posted by a_k but missed by $D(q^{(t)})$. The specific credit assignment at episode t is:

$$G^{(t)}(a_k) = \sum_{a_k \rightsquigarrow v} \sigma\left(\frac{PE(v; \mathcal{R}, p^{(t)}, q^{(t)}) - AVG(PE)}{Z}\right), \quad (9)$$

where $Z = \max_{v \in \mathcal{V}_T} PE(v; \mathcal{R}, p^{(t)}, q^{(t)}) - \min_{v \in \mathcal{V}_T} PE(v; \mathcal{R}, p^{(t)}, q^{(t)})$. The maximum, minimum, and average are calculated over all targets $\mathcal{V}_T$, including those not attacked by the strategy a_k. The subtraction of the average from $PE(v; \mathcal{R}, p^{(t)}, q^{(t)})$ can help differentiate effective attacks from less effective ones [32]. The rewards are accumulated across multiple episodes of the game, and $p^{(t)}$ is as:

$$p_k^{(t+1)} \propto \left[p_k^{(0)} + \eta \sum_{\tau=1}^{t} \frac{G^{(\tau)}(a_k)}{|\{v : a_k \rightsquigarrow v\}|}\right]. \quad (10)$$

An MDP for the detector. In episode t, the detector senses the state of the review system as the vector $d(r_i) = [d_1(r_i), \ldots, d_L(r_i)]$

for each review $r_i \in \mathcal{R}(p^{(t)})$ after the spamming attack $A(p^{(t)})$. To evaluate the current detection strategy, the labels of the added spams are not disclosed to the detector when the detector $D(q^{(t)})$ takes its actions. To simplify training, the base detectors d_l are fixed. The strategy $q^{(t)}$ generates the probability $P(y = 1|r; q)$ on each review $r \in \mathcal{R}(p^{(t)})$ according to Eq. (7). The top-k suspicious reviews based on $P(y = 1|r, q^{(t)})$ are removed, leading to the set $\mathcal{R}(p^{(t)}, q^{(t)})$, which potentially contains false negatives that contribute to $PE(v; \mathcal{R}, p^{(t)}, q^{(t)})$. The spamming effect attributed to individual false negatives is C_{FN} defined in Eq. (4). The mixing parameter $q^{(t)}$ will be updated to $q^{(t+1)}$ by minimizing the following cost-sensitive loss function:

$$\mathcal{L}(q) = \frac{1}{|\mathcal{R}(p^{(t)}, q^{(t)})|} \sum_{r \in \mathcal{R}(p^{(t)}, q^{(t)})} -C_{FN}(v, r) \log P(y = 1|r; q),$$
$$(11)$$

where v is the target that r was posted to. After the detection, the current episode is done, and both agents move to the next episode and play the same game with the updated parameters $(p^{(t+1)}, q^{(t+1)})$.

Optimization algorithm. We propose Nash-Detect in Algorithm 1 for finding a Nash equilibrium (p^*, q^*), and q^* is the resulting robust detection strategy. Figure 1a presents a toy-example of Nash-Detect. At a Nash equilibrium, both the spammer and detector will not want to change their strategies (p^*, q^*) since it will not lead to further benefit. There can be multiple Nash equilibria and the algorithm just finds one. It is challenging to prove the uniqueness of Nash equilibria except in some very restrictive problem settings. Experiments show that Nash-Detect always finds a robust detector regardless of what p^* is. Nash-Detect is trained in H episodes. During each episode $t = 1, \ldots, H$, there are the forward and backward steps:

- Inference (forward): for each target $v \in \mathcal{V}_T$, the fictitious spammer samples one pure strategy from $a_1, \ldots, a_K$ according to $p^{(t)}$ and posts spams to v using an account determined by the selected strategy. The sampling of a_k and the selection of accounts by a_k are conducted independently among the targets. That is, posting spams to one target will not affect the selection of a_k and how a_k runs its attack on the subsequent targets. It is left as future work to consider updating the review data before spamming the next target so there can be dependencies between the spams towards two targets. When all targets receive their spams, the detector $D(q^*)$ removes the top-k suspicious spams and the practical effect $PE(v; \mathcal{R}, p, q)$ is computed as the final output of the inference (forward pass).
- Learning (backward): the current rewards for the spamming strategies are calculated using Eq. (9) and $p^{(t)}$ is updated to $p^{(t+1)}$. Simultaneously, the detection strategy $q^{(t)}$ is also updated by minimizing the loss function Eq. (6) to obtain $q^{(t+1)}$.

Discussion. Nash-Detect assumes the spammer aims at promoting, rather than demoting the targets. Nash-Detect can handle demotion by changing the practical spamming effect from $\max\{0, PE(v; \mathcal{R}, p, q)\}$ to $\min\{0, PE(v; \mathcal{R}, p, q)\}$ where $PE(v; \mathcal{R}, p, q))$ shall be made a small negative number to bring down the targets' revenues.

Algorithm 1 Nash-Detect: Training a Robust Spam Detector

1: **Input**: all reviews $\mathcal{R}$, target items $\mathcal{V}_T$, pure attack strategies $[a_1, \ldots, a_K]$, pure spam detectors $[d_1, \ldots, d_L]$, initial spamming strategy $\mathrm{p}^{(0)} = [p_1, \ldots, p_K]$ and initial detection strategy $\mathrm{q}^{(0)} = [q_1, \ldots, q_L]$ to uniform distributions.
2: **Output**: a Nash equilibrium $(\mathrm{p}^*, \mathrm{q}^*)$.
3: **repeat**   ▷ Go through the H episodes indexed by t
4:    Inference:
5:      $\mathcal{R}(\mathrm{p}^{(t)}) = \mathcal{R}$.
6:      **for all** $v \in \mathcal{V}_T$ **do**   ▷ Post fake reviews
7:         Sample a_k using ϵ-greedy for v according to $\mathrm{p}^{(t)}$.
8:         Post spams to v using a_k.
9:      Remove spams in the top k reviews detected by $D(\mathrm{q}^{(t)})$.
10:     Compute $\mathrm{PE}(v, \mathcal{R}, \mathrm{p}^{(t)}, \mathrm{q}^{(t)})$ using Eq. (2) on $\mathcal{R}(\mathrm{p}^{(t)}, \mathrm{q}^{(t)})$.
11:    Learning:
12:     Compute $C_{\mathrm{FN}}(v, r)$ using Eq. (4) and $G(a_k)$ using Eq. (9).
13:     Update $\mathrm{p}^{(t)}$ to $\mathrm{p}^{(t+1)}$ using the gains $G(a_k)$.
14:     Update $\mathrm{q}^{(t)}$ to $\mathrm{q}^{(t+1)}$ by minimizing Eq. (11).
15: **until** $\mathcal{L}(\mathrm{q})$ converges

When applying the resulting detector, admittedly, one must know whether a spammer will promote or demote a product. A simple solution is to learn about the probability that a product will commit self-promotion and apply Nash-Detect trained based on $\max\{0, \mathrm{PE}(v; \mathcal{R}, \mathrm{p}, \mathrm{q}))\}$. For example, pieces of evidence showed that when a product recently received negative reviews, it is more likely to commit spamming for self-promotion. On the other hand, demotions are more likely among competitors who offer similar products (e.g., Samsung vs. Apple [35], or Starbucks vs. Dunkin' Donuts [22]). We leave as future work regarding apply two robust detectors to handle demotions and promotions simultaneously.

3.4 Base Detection and Spamming Strategies

Base detection strategies. As mentioned in Section 2, there have been a variety of graph-based and behavior-based detectors. We select the following five base detectors:

- GANG [38]: a social network Sybil detection algorithm via linearized belief propagation.
- SpEagle [27]: an advanced belief propagation algorithm verified on Yelp review spam datasets.
- fBox [30]: an SVD-based algorithm that spots small-scale suspicious links in social networks.
- Fraudar [10]: a fraudster detection algorithm that detects dense blocks in graphs.
- Prior: an algorithm that ranks spams based on multiple suspicious behavior features listed in [23, 24, 27].

These detectors are unsupervised [10, 27, 30], have no or very few parameters [10, 23], are scalable [38], diverse, and representative of explicit assumptions about the spamming strategies, making the resulting detector more robust and interpretable. Supplement A.1.4 shows their implementation details. There is no hurdle for Nash-Detect to include deep learning methods [13, 20] as base detectors. However, the underlying spamming strategies are *learned* from

many *labeled* data, leading to a less interpretable detector (in the sense of explaining the interplay between any two detection and spamming strategies, and why the resulting detector is robust).

Base spamming strategies. Different from the secure machine learning literature [2], a spamming strategy cannot manipulate some important detection features directly but has to do so via posting new reviews. On some review systems, a spammer can retract posted reviews and can be used in attacks against clustering and classification on graphs [31, 47]. We assume that only additions of reviews are allowed, similar to the attacks on graphs proposed in [5]. The reasons are that adding reviews is the most direct way to perturb target ratings, while deleting reviews can make a spamming account look suspicious [29] and reduce the effects of previous attacks. There are attacks based on text generation [14, 42] and control of spamming tempo [9], but we do not consider such controls in this work for simplicity. All these prior attacks do not distinguish elite and regular accounts and are not trained to maximize the practical spamming effect. We consider the situation where elite and regular accounts contribute differently to the practical spamming effect [40, 45]. We propose the following *base* spamming strategies that differ in their target detectors and whether elite accounts are employed, and let Nash-Detect learn the importance of each base spamming strategy. The implementation details of following strategies are shown in Supplement A.1.5.

- IncBP: it uses elite accounts and tries to evade detectors that use both behavior [23, 24, 40] and graph information [27, 38]. It employs linearized belief propagation (LinBP) [38] on a Markov Random Field (MRF) to estimate account suspiciousness. The strategy estimates the suspiciousness of controlled accounts after posting fake reviews to one target and posts reviews to the next target using accounts with the minimum suspiciousness.
- IncDS: it uses elite accounts and aims to evade detectors that regard dense blocks suspicious [10]. IncDS estimates the suspiciousness of each account using the density of the subgraph composed of its neighbors after a spam is posted. IncDS selects the account with minimum density to post the next spam.
- IncPR: it uses elite accounts and aims to evade the detector Prior. The spamming process is iterative with account suspiciousness estimation, similar to IncBP and IncDS.
- Random [10, 30]: it randomly picks an elite account to post a spam.
- Singleton: it uses new accounts, each of which posts only one review. Such spams can avoid creating dense blocks and thus can evade Fraudar.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We conduct experiments on three Yelp review datasets (YelpChi, YelpNYC, and YelpZip) used in [24, 27]. The datasets contain reviews labeled as spams and legitimate, but we simply assume the set of reviews are $\mathcal{R}$ where the spammer will start their spamming attacks. Dataset details are in the Supplement.

Elite account selection. The datasets only contain partial information of each account and we could not crawl the accounts' elite status from Yelp. Since Yelp takes the number of reviews posted by the accounts as a crucial factor in determining account elite status,

and there are estimated $6\% - 9\%$ elite members among all accounts of Yelp [6, 15, 44], we regard accounts with more than *ten* reviews as elite accounts in the experiments, representing 1.4%, 4.30% and 4% of the total accounts on the three datasets, respectively.

Spamming attack and detector setting. We select fixed sets of elite accounts and target products to train/test all spamming and defense strategies (details are in Supplement A.1.3). After calculating the review suspicious scores using Eq. (7), all reviews are ranked by their suspicious scores in descending order, and we remove the top $k\%$ suspicious reviews as spams. We set $k = 1$ in all experiments.

Evaluation criteria. According to Section 3.3, we expect that Nash-Detect can find the optimal detector configuration (q*) by playing the minimax game between the spammer and defender. With $D(\text{q}^*)$, the performance of $A(\text{p})$ should be worse than the worst-case of each single attack against a single detector [36]. It will show that the detectors configured by Nash-Detect could defend the attacks better than any single detector. The performance of attacks and defenses are both measured by the practical effect (Eq. (2)).
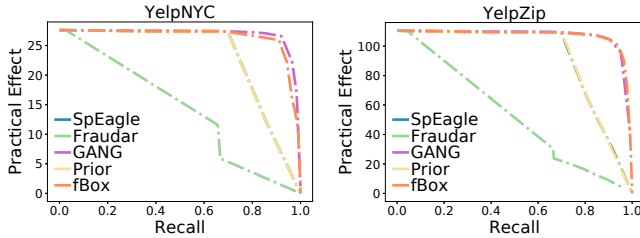
Figure 2: Practical Effect vs. Recall for different detectors against ensemble attacks on YelpNYC and YelpZip.

4.2 Practical Effect vs. Accuracy

To verify the advantage of the proposed practical goal, we plot the detection results of five single detectors against five attacks under three datasets. Figure 1b and Figure 2 show the Practical Effect-Recall curves of five detectors against multiple attacks on three Yelp datasets. The practical effect is calculated according to Eq. (2) and recall score is obtained by using the suspicious scores as a detector. Given the review suspicious scores, as we decrease the classification threshold from 100% to 0%, more spams will be detected, the recall score increase from 0 to 1, and the practical effect should gradually decrease. However, most detectors (GANG, fBox, SpEagle, Prior) only reduce a small practical effect of attacks when their recall scores increase from 0 to 0.8. It demonstrates that the practical goal proposed by us could capture the practical effect of the spamming campaign, and a high accuracy metric does not mean a good practical detection performance. We also employ the practical effect to show the vulnerability of individual detectors. Table 2 shows the practical effect of individual detectors against individual attacks. One can see that if the spammer knows that a specific detector (such as Fraudar and fBox) is adopted by the defender, the spammer can adopt the spamming strategy (such as Random or IncBP) that leads to the most practical effect with respect to the known detector. Therefore, a detector ensemble configuration $D(\text{q})$ is necessary.

Table 2: The practical effect of detectors against attacks under YelpChi.

	GANG	SpEagle	fBox	Fraudar	Prior
IncBP	4.8916	4.9052	4.9125	1.4203	4.9099
IncDS	4.9010	4.9052	4.9110	4.8959	4.9099
IncPR	4.9010	4.9052	4.9105	3.0716	4.9099
Random	4.9010	4.9052	4.9092	4.8962	4.9099
Singleton	0.5300	0.5865	0.5783	0.5771	0.5912

4.3 Nash-Detect Training Process

In the training phase, we run Nash-Detect for 50 episodes on three datasets, and the results are shown in Figure 3.

Spamming strategy updating. Figure 3a shows the updates that the spamming strategy mixture parameters (p) undergo. We can see that Nash-Detect successfully identifies reviews posted by new accounts using `Singleton` as insignificant to the practical effect. A similar observation is shown in Table 2 where the `Singleton` attack's practical effect is much smaller than the other spamming strategies against all detectors. This observation is consistent with the practical influence of non-elite accounts discussed in Section 3.2. In Figure 3a, the fluctuations in the relative importance (p_k values) of the remaining four spamming strategies indicate that it is hard to find a single useful spamming strategy to evade the defense strategy, which is also under training and evolving.

Detector importance updating. By juxtaposing Figures 3a and 3b, we can see that no matter how the spamming strategy evolves, the detectors' weights (q) move smoothly toward the optimal configuration. Meanwhile, the loss of detectors (Eq. (11)) converges as training episodes increases (Figure 3f). Note that in Figure 3b, though the relative importances of GANG and Fraudar (q) differ across datasets, the two detectors both perform better than other detectors under the worst cases individual attack under each dataset. This means that the Nash-Detect converges to the optimal configuration.

Practical effect. According to Figures 3b and 3d, the learned detector importance (q) do not use a single detector but with a stable distribution that guarantees the worst-case performance is better than the worst-case when using a single detector (dash lines at the top). By comparing Figure 3c and 3e, we see that the number of evaded spams are not positively correlated with the total rewards of the spamming attacks on the target products. This observation further confirms that the promotions mainly come from the small number of elite accounts that evaded the detection, even with many singleton spams detected. Similar observations can be seen from Figures 3c and 3d, as the practical effect will be backpropagated to individual products as their rewards. The bumping of the number of evaded reviews on the YelpNYC dataset is due to the suspicious scores of posted fake reviews are around the filtering threshold.

Interpretability. Since the fake reviews posted by elite accounts contribute much more than other reviews to practical effect (Eq. (2)), Nash-Detect favors detectors that can spot more elite accounts. From the graph perspective, the elite accounts and target products are likely to form dense blocks when fake reviews are added between the accounts and the products. Since Fraudar is designed to spot dense blocks, more elite spams should be detected and Nash-Detect successfully assigns higher importances to Fraudar, as
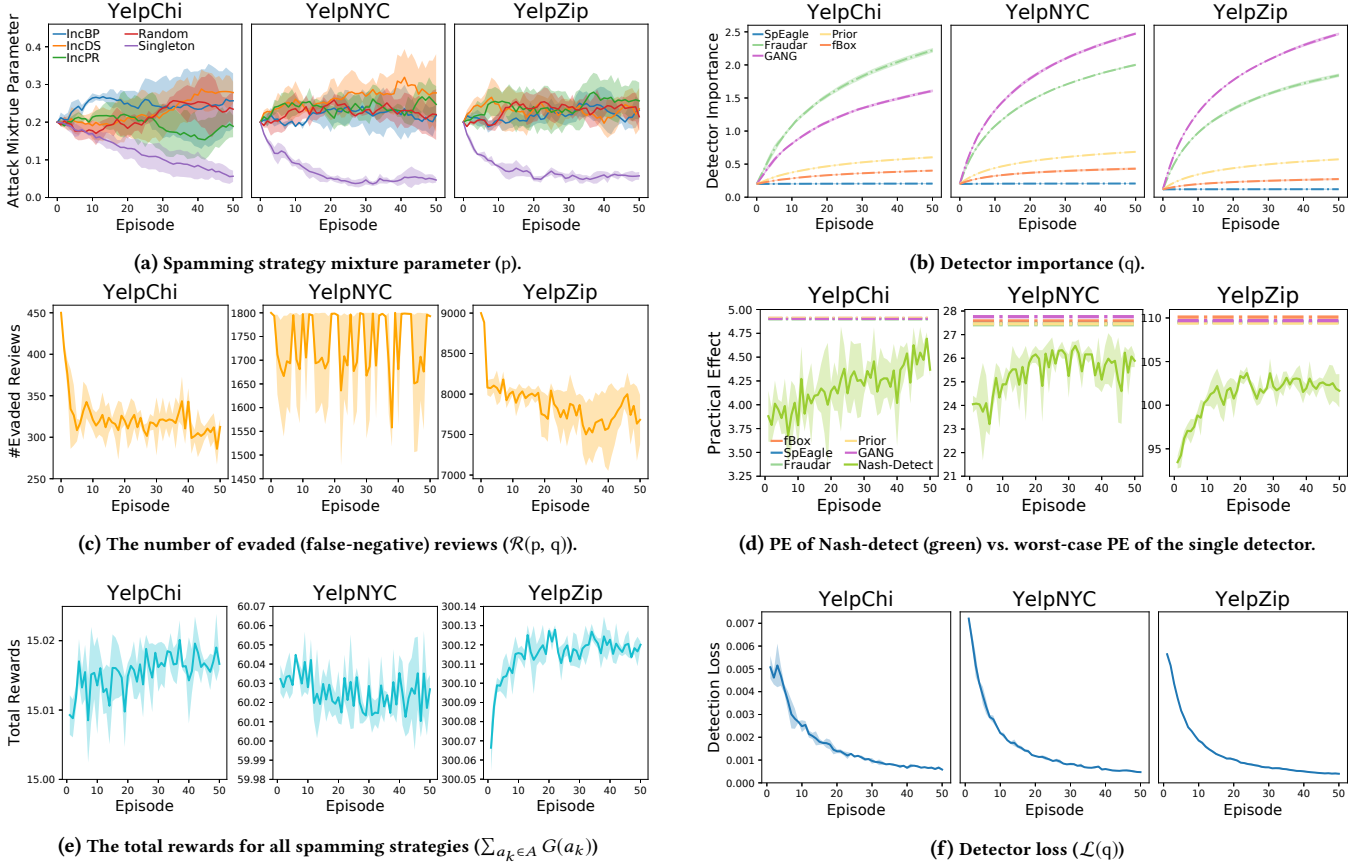
(a) Spamming strategy mixture parameter (p).

(b) Detector importance (q).

(c) The number of evaded (false-negative) reviews ($\mathcal{R}$(p, q)).

(d) PE of Nash-detect (green) vs. worst-case PE of the single detector.

(e) The total rewards for all spamming strategies ($\sum_{a_k \in A} G(a_k)$)

(f) Detector loss ($\mathcal{L}$(q))

Figure 3: The training process of Nash-Detect with 50 episodes.

shown in Figure 3b. Contrarily, fBox is designed to detect "small" attacks and can't handle elite spams. Since singleton spams are insignificant to the practical influence, Nash-Detect assigns lower importance to fBox.

4.4 Performance in Deployment

In the testing phase, we select new sets of controlled accounts following the same settings in the training phase but keep the same set of target products. For each target product, we sample base spamming strategies uniformly to test the performance of various detectors in suppressing practical spamming effects when deployed. We repeat the sampling for 10 times. In Figure 4 we show the means and standard deviations of the practical effects when the detectors are deployed. Equal-Weights is a baseline where all base detectors have the same importance, and Nash-Detect represents the optimal detector configuration (q*) found during the training phase. From Figure 4, we can see that Nash-Detect (in red bars) has lower mean practical effects, compared to other detectors on YelpChi and YelpNYC. On YelpZip, the performance of Nash-Detect is not as good as Fraudar's, since the selected accounts in the testing phase have a higher percentage of elite accounts with more reviews (> 100). As a result, Fraudar can easily spot the large dense blocks and has a much lower practical effect on YelpZip. This suggests that one should incorporate diverse accounts and products during

training to guarantee the robustness of the resulting detectors. More experiment results are presented in the Supplement A.2.

5 RELATED WORK

Spam detectors. Besides text-based detectors [13, 24, 27, 39], there are three major types of spam detectors: behavior-based detectors, graph-based detectors, and detectors using both behavioral and graph structural information. Behavior-based detectors [23, 24, 27, 40] calculate review suspicious scores based on different behavior features. Graph-based detectors adopt representation learning approaches, such as SVD [30], node2vec [14], and more recently graph neural networks [20]. Some graph-based detectors identify dense blocks [10] or dense flows [18] of the spamming accounts and targets from review graph. Examples of detectors using both behavioral and graph structural information include the SpEagle [27], GANG [38], and GraphConsis [20], etc.

Marketing research on spam campaigns. Online review systems and the associated spam campaigns have drawn much attention from researchers in marketing. [8, 21, 43] study how online business ratings influence consumers' purchase decisions. [43] shows that a ten percent increase in traveler review ratings will boost online bookings by more than five percent. [8] suggests that both the reviews and the accounts' social status will contribute
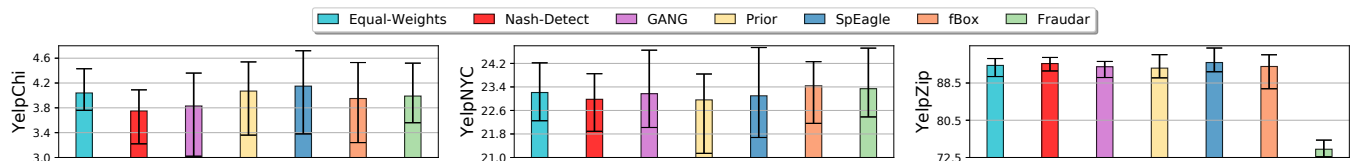
Figure 4: The practical effect of Nash-Detect and baselines during testing.

to revenue growth. Inspired by the above works, we propose to incorporate economic incentives and accounts' social status into the spamming practical goal. In the real world business system, there are other factors like marketing operation status and business category that may affect the business revenue as well. However, it is infeasible for us to model such a complicated relationship using only the online review data. Therefore, in this paper, we mainly focus on the relationship between rating and revenue.

6 CONCLUSION

In this work, we propose a practical metric for review systems with product revenue promotion in mind. We investigate the practical performance of mainstream spam detectors against several goal-oriented spamming attacks in an adversarial setting. We formulate a game theoretical model and reinforcement learning algorithm to find a robust detection strategy against a diverse set of spamming strategies. Empirical evaluations on three large review datasets demonstrate that the proposed algorithm can indeed generate detectors that can effectively tame the practical spamming goal of product revenue manipulation. It remains our future work to adopt the proposed model and algorithms to adversarial misinformation attack in other reputation systems, such as rumor detection on social networks.

ACKNOWLEDGMENTS

REFERENCES

[1] L. Akoglu, R. Chandy, and C. Faloutsos. 2013. Opinion fraud detection in online reviews by network effects. In *ICWSM*.
[2] B. Biggio, G. Fumera, and F. Roli. 2013. Security evaluation of pattern classifiers under attack. In *ICDE*.
[3] A. Breuer, R. Eilat, and U. Weinsberg. 2020. Friend or Faux: Graph-Based Early Detection of Fake Accounts on Social Networks. In *WWW*.
[4] S. Bubeck and N. Cesa-Bianchi. 2012. Regret Analysis of Stochastic and Non-stochastic Multi-armed Bandit Problems. *FTML* (2012).
[5] Y. Chen, Y. Nadji, A. Kountouras, F. Monrose, R. Perdisci, M. Antonakakis, and N. Vasiloglou. 2017. Practical Attacks Against Graph-based Clustering. In *CCS*.
[6] W. Dai, G. Z. Jin, J. Lee, and M. Luca. 2012. Optimal aggregation of consumer ratings: an application to yelp. com. *NBER WPS* (2012).
[7] C. Elkan. 2001. The foundations of cost-sensitive learning. In *IJCAI*.
[8] C. Forman, A. Ghose, and B. Wiesenfeld. 2008. Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets. *ISR* (2008).
[9] S. Ge, G. Ma, S. Xie, and P. S. Yu. 2018. Securing Behavior-based Opinion Spam Detection. In *IEEE Big DATA*.
[10] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos. 2016. FRAUDAR: Bounding Graph Fraud in the Face of Camouflage. In *KDD*.
[11] J. Hu and M. P. Wellman. 2003. Nash Q-learning for General-sum Stochastic Games. *JMLR* (2003).
[12] P. Kaghazgaran, M. Alfifi, and J. Caverlee. 2019. TOmCAT: Target-Oriented Crowd Review Attacks and Countermeasures. In *ICWSM*.
[13] P. Kaghazgaran, M. Alfifi, and J. Caverlee. 2019. Wide-Ranging Review Manipulation Attacks: Model, Empirical Study, and Countermeasures. In *CIKM*.
[14] P. Kaghazgaran, J. Caverlee, and A. Squicciarini. 2018. Combating crowdsourced review manipulators: A neighborhood-based approach. In *WSDM*.
[15] C. Kim, G. Lin, and H. Bang. 2015. Discovering Yelp Elites: Reifying Yelp Elite Selection Criterion. *UCSD* (2015).
[16] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and VS Subrahmanian. 2018. Rev2: Fraudulent user prediction in rating platforms. In *WSDM*.
[17] T. Lappas, G. Sabnis, and G. Valkanas. 2016. The impact of fake reviews on online visibility: A vulnerability assessment of the hotel industry. *ISR* (2016).
[18] X. Li, S. Liu, Z. Li, X. Han, C. Shi, B. Hooi, H. Huang, and X. Cheng. 2020. FlowScope: Spotting Money Laundering Based on Graphs. In *AAAI*.
[19] M. L Littman. 1994. Markov Games As a Framework for Multi-agent Reinforcement Learning. In *ICML*.
[20] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng. 2020. Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection. *SIGIR*.
[21] M. Luca. 2016. Reviews, reputation, and revenue: The case of Yelp. com. *HBS Working Paper* (2016).
[22] M. Luca and G. Zervas. 2016. Fake it till you make it: Reputation, competition, and Yelp review fraud. *Management Science* (2016).
[23] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. 2013. Spotting opinion spammers using behavioral footprints. In *KDD*.
[24] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance. 2013. What Yelp Fake Review Filter Might Be Doing?. In *ICWSM*.
[25] K. Murphy, Y. Weiss, and M. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *UAI*.
[26] M. Rahman, N. Hernandez, R. Recabarren, S. I. Ahmed, and B. Carbunar. 2019. The Art and Craft of Fraudulent App Promotion in Google Play. In *CCS*.
[27] S. Rayana and L. Akoglu. 2015. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In *KDD*.
[28] Y. Ren, H. Zhu, J. ZHang, P. Dai, and L. Bo. 2019. EnsemFDet: An Ensemble Approach to Fraud Detection based on Bipartite Graph. *arXiv preprint arXiv:1912.11113* (2019).
[29] ReviewMeta. 2019. How It Works. https://bit.ly/3edwG3x
[30] N. Shah, A. Beutel, B. Gallagher, and C. Faloutsos. 2014. Spotting Suspicious Link Behavior with fBox: An Adversarial Perspective. In *ICDM*.
[31] L. Sun, Y. Dou, C. Yang, J. Wang, P. S. Yu, and B. Li. 2018. Adversarial Attack and Defense on Graph Data: A Survey. *arXiv preprint arXiv:1812.10528* (2018).
[32] R. S. Sutton and A. G. Barto. 1998. *Introduction to Reinforcement Learning*.
[33] J. Swearingen. 2017. Amazon Is Filled With Sketchy Reviews. Here's How to Spot Them. https://slct.al/2TBXDpT
[34] M. Tan. 1993. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *ICML*.
[35] TheVerge. 2013. Samsung fined $340,000 for faking online comments. https://bit.ly/2WXKfOV
[36] J. Uesato, B. O'Donoghue, P. Kohli, and A. Oord. 2018. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. In *ICML*.
[37] B. Wang and N. Z. Gong. 2019. Attacking Graph-based Classification via Manipulating the Graph Structure. In *CCS*.
[38] B. Wang, N. Z. Gong, and H. Fu. 2017. GANG: Detecting Fraudulent Users in Online Social Networks via Guilt-by-Association on Directed Graphs. In *ICDM*.
[39] X. Wang, K. Liu, and J. Zhao. 2017. Handling cold-start problem in review spam detection by jointly embedding texts and behaviors. In *ACL*.
[40] S. Xie, G. Wang, S. Lin, and P. S. Yu. 2012. Review Spam Detection via Temporal Pattern Discovery. In *KDD*.
[41] C. Yang, H. Wang, L. Sun, and B. Li. 2020. Secure Network Release with Link Privacy. *arXiv preprint arXiv:2005.00455* (2020).
[42] Y. Yao, B. Viswanath, J. Cryan, H. Zheng, and B. Y. Zhao. 2017. Automated crowdturfing attacks and defenses in online review systems. In *CCS*.
[43] Q. Ye, R. Law, B. Gu, and W. Chen. 2011. The influence of user-generated content on traveler behavior: An empirical investigation on the effects of e-word-of-mouth to hotel online bookings. *CHB* (2011).
[44] Yelp. 2019. *What is Yelp's Elite Squad?* https://bit.ly/2A1dqYi
[45] H. Zheng, M. Xue, H. Lu, S. Hao, H. Zhu, X. Liang, and K. Ross. 2018. Smoke screener or straight shooter: Detecting elite sybil attacks in user-review social networks. *NDSS* (2018).

Yingtong Dou, Guixiang Ma*, Philip S. Yu, and Sihong Xie

[46] B. Zhou, Y. Yao, and J. Luo. 2014. Cost-sensitive three-way email spam filtering. *JIIS* (2014).
[47] D. Zügner, A. Akbarnejad, and S. Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *KDD*.