Securing Behavior-based Opinion Spam Detection

Shuaijun Ge*, Guixiang Ma†, Sihong Xie‡ and Philip S. Yu§
*dorisgebupt@gmail.com, †gma4@uic.edu, ‡six316@lehigh.edu, §psyu@uic.edu

***Abstract*—Fake opinion reviews, known as opinion spams, are prevalent and can undermine the trustworthiness of reputation systems such as Amazon and Yelp. Spams generated by spammers (human or bots) using certain assumed static spamming strategy can be detected quite effectively using existing detection techniques. Nonetheless, in reality, spammers can evade the deployed detectors with spams that are well-covered and can violate the assumptions of the detectors, leading to detection failures. Evasions against behavior-based detectors have received less attention, compared to those against text and graph-based detectors, leading to vulnerabilities in spam detection systems. We close this gap by first proposing a general computational model EMERAL (Evasion via Maximum Entropy and Rating sAmpLing) that uses maximum entropy model and sampling to generate evasive spams that can bypass certain existing detectors. Instances of EMERAL are derived for spammers with different goals and levels of knowledge about the detectors, targeting at both the early and late stages of the review periods of target products. We show that only a few evasion types are meaningful to the spammers, and a spammer cannot evade too many detection signals all at once. We reveal that some evasions are quite insidious and can fail all detection signals. We then propose DETER (Defense via Evasion generaTion using EmeRal), based on re-training on data augmented with diverse evasion samples generated by EMERAL. Experiments on real-world data confirm that DETER can lead to more accurate detection of both time windows with spamming activities and individual spamming reviews within those windows. In terms of security, DETER is versatile enough to be vaccinated against diverse and unexpected evasions, and is agnostic about evasion strategy and can be released without privacy concern.**

I. Introduction

More and more opinionated reviews are posted on online commerce websites such as Amazon and Yelp, which also serve as reputation systems that rate and rank items therein. The posted opinions can help consumers find high-quality products, and make products and services become more visible and boost their sales via word-of-mouth [12], [11], [9]. However, such a mechanism has also attracted many dishonest businesses to hire professional spammers to post ungrounded reviews (called "opinion spams") to manipulate product reputations [24], [46], [2], [53]. Victims of the spams include customers who are misled to low quality products, honest businesses that suffer from unfair competition, and the whole reputation system that are rendered less trustworthy.

To combat opinion spams, prior works have proposed abundant different detection models based on texts [33], [23], [24], [56], user-behaviors [33], [55], [13], network structures [1], [52], [29] and ensemble methods [33], [1]. However, more resourceful spammers can exploit information about the detectors available through publications, spam-spotting guidance and detection websites (Fakespot:https://www.fakespot.com/ and ReviewMetahttps://reviewmeta.com/), to craft insidious spamming campaigns that can evade certain detectors. Some detection models consider spams that are specially crafted to evade graph-based and text-based detectors [8], [18]. Evading ensembles of classifiers, in general, has also been studied in [10], [4], [45], [43]. From the perspective of detection, knowing how to generate evasive samples can significantly improve defense via model re-training [47], [31], [25], [22]. However, adversarial evasions against behavior-based detectors complementary to text and graph-based detectors have so far received less attention. This leads to potential vulnerabilities in spam detection systems that integrate behavior-based detectors.

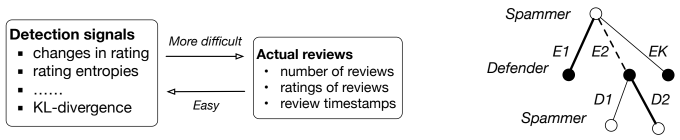


Fig. 1: (left) extracting detection signals from reviews is easier than inferring actual reviews from signal values; (right) a spammer adopts an evasion strategy, say E1, to evade a defense strategy, say D2, which is optimal for another evasion strategy, say E2, rather than E1.

Addressing this gap is non-trival, however. First, evasions against behavior-based detectors are distinct from previous evasions and are more difficult to obtain quantitatively and computationally. Existing evasion attacks assume closed-form objective functions over graphs [18], [8], or differentiable detection models [34], [3], [38] for evasion generation. There are also existing algorithms to directly create evading instances that embed PDF files with malwares [57], [44], [35], [47]. These methods are not applicable as they cannot quantitatively map the detection signal values to actionable spamming behaviors. For example, for a spammer to boost monthly rating without exceeding 4.5 stars, and to make the month's rating distribution close to last month's, it can post different numbers of fake reviews with different ratings to achieve the same set of goals. While genetic algorithms can heuristically modify previous spamming campaigns to fool detection signals, such approaches are not scalable [47]. This computational challenge is not pertaining to any higher level detection models that use the signals as features, and evading a particular classification model such as logistic regression or tree ensemble [3], [25] is not relevant.

Second, strategically, any deployed detectors are subject to adversarial probing. For example, as the game tree in Figure 1 shows, the defender fits a model using spams labeled during a period when the spammers used evasion strategy E2, resulting in defense strategy D2. A later spammer can look for patterns in the reviews deleted by D2 (for example, Yelp releases the detected reviews as "not recommended reviews"), reverse-engineer the detector [37], and switch to strategy E1 to evade D2, at which moment D1 is better for the defender. Further, diverse spamming strategies can be adopted simultaneously by multiple spammers for different targets. These scenarios lead to spamming-detecting strategy asymmetry — the defense strategy is not optimal with respect to the actual spamming strategy, and a detector assuming a fixed evasion strategy [47], [5], [34] is more vulnerable. Model retraining is promising, only if evasion samples can be generated and put in the training data, which is not possible without a computation models. Ideally, a detector has to be agnostic of any spamming strategies, but the simple solution of blindly reacting to any spamming strategies can produce too many false positives (see the experiments). The detector shall also be able to withstand any probing from the spammers, and releasing the detector shall not jeopardize the detection system.

To address the first challenge, we first identify a set of state-of-the-art detection signals [59], [36], [37], [40], [55], [13] that characterize spammer behaviors such as the frequency, amount, and rating distribution of spamming reviews, which are also the evasion targets of the spammers. We then propose a maximum entropy formulation to obtain an algorithm called "EMERAL" that encodes the spammers' knowledge and generates spamming behaviors that can evade detection based on rating distribution, amount and frequency of spams. The model, for the first time, captures the quantitative dependencies between the spammers' knowledge and actionable spamming behaviors, allowing the spamming behaviors to be computed from detection signals via optimization and sampling. The model is general, as multiple types of evasions against behavior-based detection signals can be included as objectives or constraints during both the early and late review stages of a product.

With EMERAL, a spammer can technically evade exponentially many subsets of the signals and aggravate the "strategic asymmetry". Nonetheless, we empirically show that (see Figures 2c) evading larger subsets of detection signals in a single spamming campaign is infeasible for a spammer, due to the overly constrained solution space. Instead, a rational spammer will focus on evading smaller but more valuable subsets of signals, and indeed, we discover such a dominating evasion strategy capable of evading any single detection signal in practical spamming success metrics. While one can devise a specific defense against the dominating evasion strategy, an evasion agnostic defense without assuming the strategy adopted by the spammer is more useful, since in reality, there are usually multiple spammers with diverse goals and knowledge, and the spammer can easily change its strategy. For the purpose, we propose a novel defense, DETER, based on retraining, where training data containing possible future evasive spams are first generated by EMERAL and then used to train more effective detector without assuming a fixed evasion strategy. Based on the weights learned by DETER and the properties of evasion generation, DETER can be released to the spammer without security concern. Experimentally, the new defense is shown to be superior to any fixed single detection signals, simple signal aggregation and even ensembles of multiple classifiers trained on the same adversarial examples.

TABLE I: Abnormal reviewer behavior detection signals.

Signal names	Suspicious when	Descriptions
NR (ΔNR)	H (H)	Number of reviews and change of NR in a window [55].
ΔCAR	H (H)	Change in Cumulative Average Rating
CAR-DEV	H	Deviation of CAR from its predicted value [59].
NPR (ΔNPR)	H	Number of positive reviews and its changes.
EN (ΔEN)	L (H)	Entropy of ratings (and its change) in each window.
KL-DIV	H	KL-divergence between rating distribution of a window and historic distribution.

II. Detection and threat models

A review system has a set of accounts $\mathcal{U} = \{u_1, \dots, u_n\}$, items $\mathcal{V} = \{v_1, \dots, v_m\}$, and reviews $\mathcal{R} = \{r_{ij} : i \in \{1, \dots, n\}, j \in \{1, \dots, m\}\}$, where r_{ij} is the review posted by account u_i to item v_j. r_{ij} contains its text contents $c(r_{ij})$, its rating $s(r_{ij})$ and its posting time $t(r_{ij})$. We focus on detection model based on aggregated rating behaviors over time [14], [55], [13], [26], [59]: reviews in $\mathcal{R}$ are grouped into windows and for each window, numeric detection signals in Table I are computed to obtain window suspicious scores. These window-wise signals are unique and not available in detection on the review, reviewer and item level, and can help detect individual reviews [14]. We focus on spammers, be it human or bots, with the goal of promoting the target products' long and short term reputation, measured in cmulative average rating (CAR) and current month ranking (CMR, defined as the ranking of a business, among all businesses, based on the current month's average rating [48]). CAR and CMR are shown to be vulnerable to spammers' manipulations [28], [49], [15]. The demoting spams can be handled similarly by the proposed models. We first introduce behavior-based detection signals defined by previous work.

A. Time series based detection signals

Normal review traffic shall arrive in a smooth manner while spamming reviews usually arrive in a more abrupt pattern [55], [13]. Besides, to effectively promote product reputation, spammers also aim at lifting the average rating of the targets significantly [12], [11], [9]. Time series-based detection constructs and monitors time series to spot such changes in review volume and rating. A time series is a sequence of temporally ordered random variables $\boldsymbol{x} = [X_1, X_2, \dots, X_t, \dots]$, and $\boldsymbol{x}_m^n = [X_m, \dots, X_n]$ denotes the portion from time window m to n. For the t-th window (we also refer t to the window or the timespan of the window), the signals NR (number of reviews) and CAR (cumulative average rating) can be calculated to obtain two time series:

$$\text{NR}(t) = |\{r : t(r) \in t\}|, \quad \text{CAR}(t) = \frac{\sum_{t(r) \leq t} s(r)}{N_t},$$

where N_t is the number of reviews ever posted up to window t. These two series can capture the large volume of spamming reviews and inflated average ratings. Changes in NR and CAR, denoted by ΔNR and ΔCAR, can capture the abrupt changes in the volume of reviews and accumulated average rating:

$$\Delta\text{NR}(t) = \text{NR}(t) - \text{NR}(t-1),$$
$$\Delta\text{CAR}(t) = \text{CAR}(t) - \text{CAR}(t-1).$$

The deviation of the actual time series value from the value predicted by a model that assumes smoothness of the series, such as auto-regressive models, can capture unexpected changes in the time series. In particular, an order d auto-regressive model (AR(d)) predicts X_t using historic data $\boldsymbol{x}_{t-d}^{t-1}$ and a linear model $\boldsymbol{\theta}^{(t)}$

$$X_t = \sum_{i=1}^{d} \theta_i^{(t)} X_{t-i} = \left\langle \boldsymbol{\theta}^{(t)}, \boldsymbol{x}_{t-d}^{t-1} \right\rangle. \tag{1}$$

The deviation of the predicted CAR ($\widehat{\text{CAR}}(t)$) from the actual CAR, can be used for detection (only promotion is considered):

$$\text{CAR-DEV}(t) = \max\{\widehat{\text{CAR}}(t) - \text{CAR}(t), 0\}.$$

The larger the CAR-DEV, the more suspicious the window.

B. Distribution-based detection signals

A spammer needs to post a large number of positive fake reviews to promote the target. Thus if the percentage of positive reviews within a window is abnormally high, there are likely spamming activities. The signal PR (Positive Ratio) [40], [37] is calculated based on this intuition:

$$\text{PR}(t) = \frac{|r : s(r) \geq 4 \text{ and } t(r) \in t|}{n_t},$$

where n_t is the number of reviews within window t. Second, the overall rating distributions of the t-th window $\boldsymbol{p}(t) = [p_1(t), \ldots, p_5(t)]$, with $p_i(t)$ be estimated by $|r : s(r) = i \text{ and } t(r) \in t|/n_t$, can be perturbed by spamming ratings and deviate from the background rating distribution. Such distortion in rating distribution can be used as for spam detection [14], [36], [40]. Let $\boldsymbol{p} = [p_1, \ldots, p_5]$ be the rating distribution of all historic ratings up to time t: $p_i = |r : s(r) = i \text{ and } t(r) \leq t|/N_t$. The KL divergence between these two distributions detects distortion in rating distribution:

$$\text{KL-DIV}(\boldsymbol{p}(t) \parallel \boldsymbol{p}) = \sum_{i=1}^{5} p_i(t) \log \frac{p_i(t)}{p_i}$$

The larger the KL-DIV, the farther $\boldsymbol{p}(t)$ is away from $\boldsymbol{p}$, and thus the more suspicious the t-th window. Third, define the rating entropy

$$\text{EN}(\boldsymbol{p}(t)) = -\sum_{i=1}^{5} p_i(t) \log p_i(t)$$

If the rating entropy of a window is low, then the ratings therein are highly concentrating on a certain value while a normal distribution shall have a certain level of dispersion across multiple values [40] (such as a U-shape [19]). A related signal is the change in rating entropy $\Delta\text{EN} = \text{EN}(t) - \text{EN}(t-1)$. The window t is suspicious if $\Delta\text{EN} < 0$.

C. Threat model

A threat model captures what knowledge about the defense system a spammer can learn about, and how the spammer can exploit the knowledge to evade the system [8], [47], [50], [44]. As their core service, review websites have to make available a large number of reviews, including account and item profiles, review ratings and timestamps, to all users, including spammers. This information is released after filtering by the spam detection models of these websites and thus represents what the websites regard as "normal". Detection signals and algorithms based on time series smoothness and rating distribution are likely deployed by certain review websites and then published with great details [37]. Spam spotting guides (such as Fakespot and ReviewMeta) also leak detection signals to the spammers. As a result, a spammer can reconstruct the detection signals from the review data and infer when a spamming campaign is likely to bypass the detection. Regarding hyper-parameter of the detection signals, we empirically show that a spammer does not need to have exact knowledge for effective evasions (Figures 2e and 2j). Regarding the amount of review data needed for evasion, the proposed evasion model is applicable to cases with either a small or large amount of historic review data: new or less popular products receive less reviews than old and popular products. Regarding labeled data, while Yelp releases spams flagged by their system, labeled data are not widely available in other websites and is not required by our threat model. Signals based on review texts or graphs are orthogonal to the behavior-based signals, and a spammer does not need knowledge about these signals to conduct successful evasions.

The defender needs to aggregate multiple detection signals to generate the final suspicious scores for detection. Spammers can have different levels of knowledge about the aggregation. A spammer with minimal knowledge is aware of the signals but not how they are aggregated, and it can evaluate an evasion strategy against individual signals. A spammer with moderate knowledge is aware of simple aggregation of the detection signals, such as taking a uniform linear combination, selecting a random signal, or taking the signal with a maximal suspicious score. The spammer can therefore evaluate evasion strategies against these simple aggregation methods. Lastly, a spammer with perfect knowledge can learn about how the signals are aggregated for detection. The proposed defense DETER aggregates the signals in a linear but adaptive way and is agnostic about evasion strategies, and more importantly, requires no protection of its parameters (see Section V).

III. EMERAL: AN EVASION GENERATOR

In Sections III-A we present EMERAL (Evasion via Maximum Entropy and Rating sAmpLing) to generate evasions against detection signals using rating behaviors. The model is general and can easily incorporate evasions against multiple signals (including but not restricted to those in Table I) as constraints. The resulting optimization problem allows effective and efficient evasion generation (Section III-C).

A. Evading behavior-based signals

The signals KL-DIV, EN, ΔEN and PR rely on rating distribution, and in desirable spam campaigns, a spammer needs to know the exact ratings of each of spams so that the target's reputation can be promoted while evading the signals. The idea is to first find a rating distribution to evade these signals and then sample from the distribution to create the actual spams. We use the above 4 signals as examples in our formulation, but in general, signals based on rating distribution can be incorporated similarly.

To evade KL-DIV, all ratings, including fake and normal ratings, in the current time window should have a distribution $\boldsymbol{p}$ that is close to the rating distribution $\bar{\boldsymbol{p}}$ that the defender considers normal. Specifically, let $R \in \{1, 2, \ldots, 5\}$ be a random variable of ratings such that $p(R = i) = p_i \geq 0$ and $\sum_{i=1}^{5} p_i = 1$. For a target business with long rating history, the spammer can estimate $\bar{p}_i$ using the proportion of reviews with rating i for the business. The spammer can find for the t-th window an evasive rating distribution $\boldsymbol{p}$ with minimal KL-divergence to $\bar{\boldsymbol{p}}$

$$\min_{\boldsymbol{p}} \quad \mathrm{KL}(\boldsymbol{p}||\bar{\boldsymbol{p}}) = \textstyle\sum_{i=1}^{5} p_i \log \frac{p_i}{\bar{p}_i}. \tag{2}$$

The spammer's ultimate goal is to move CAR to $\tilde{x}_t = x_t + \delta_t^*$,

$$\tilde{x}_t - \epsilon \leq \frac{N_{t-1}x_{t-1} + (n_t + n_\delta)\mathbb{E}_{\boldsymbol{p}}[R]}{N_t + n_t + n_\delta} \leq \tilde{x}_t, \tag{3}$$

where x_t is the CAR at time t and $\tilde{x}_t$ is the spammer's target CAR value, which will be optimized in the next section. N_{t-1} is the number of ratings accumulated up to time t, n_t is the number of existing ratings at time t without the spamming ratings, n_δ is the number of spamming ratings to be added. $\mathbb{E}_{\boldsymbol{p}}(R) = \sum_{i=1}^{5} ip_i$ is the expectation of R. $\frac{N_{t-1}x_{t-1}+(n_t+n_\delta)\mathbb{E}_{\boldsymbol{p}}[R]}{N_t+n_t+n_\delta}$ is the manipulated CAR at time t after the attack. $\epsilon > 0$ is a small positive number to allow some fluctuation in the target CAR. In sum, the spammer needs the manipulated CAR to be close to but not to exceed the target $\tilde{x}_t$. In addition, the spammer can evade signals ΔEN and NPR by adding more constraints to the above problem, leading to the following inequality-constrained KL-divergence minimization problem:

$$\begin{aligned}
\min_{\boldsymbol{p}} \quad & \mathrm{KL}(\boldsymbol{p}||\bar{\boldsymbol{p}}) \\
\text{s.t.} \quad & \mathbb{E}_{\boldsymbol{p}}[R] \leq U \triangleq \frac{(N_t + n_t + n_\delta)(x_t + \delta_t^*) - N_{t-1}x_{t-1}}{n_t + n_\delta}, \\
& -\mathbb{E}_{\boldsymbol{p}}[R] \leq B \triangleq \frac{(N_t + n_t + n_\delta)(x_t + \delta_t^* - \epsilon) - N_{t-1}x_{t-1}}{n_t + n_\delta}, \\
& -H(\boldsymbol{p}) \leq -(H_{t-1} + H_\delta) \triangleq -H, \\
& p_4 + p_5 \leq P, \qquad \textstyle\sum_i p_i = 1.
\end{aligned} \tag{4}$$

The first two constraints are derived from Eq. (3), and the third enforces the entropy of the rating distribution at time t, denoted by $H(\boldsymbol{p})$, to be no less than $H_{t-1} + H_\delta$ to evade the detection of ΔEN. The constraint $p_4 + p_5 < P$ ensures that after spamming, the ratio of positive reviews (4 and 5 star ratings) will not exceed P to evade the detection of PR (ratio of positive reviews). The optimization can be solved using Lagrangian multiplier method:

$$\boxed{\begin{aligned}
\max_{\alpha,\beta,\gamma,\lambda} \quad & L(\alpha, \beta, \gamma, \lambda) \\
\text{s.t.} \quad & \alpha \geq 0, \beta \geq 0, \gamma \geq 0, \lambda \geq 0
\end{aligned} \quad (5)}$$

$$L(\alpha, \beta, \lambda, \gamma) = -(1+\gamma)\log Z - (1+\gamma) - \alpha U + \beta B - \lambda p + \gamma H$$

and $Z = \sum p_i = \sum_i \exp(S_i/(1 + \gamma))$ with

$$S_i = \log \bar{p}_i - (\alpha - \beta)i - 1 - \lambda\mathbb{I}(i) - \gamma \tag{6}$$

We can use gradient ascent to find the optimal Lagrangian multipliers α^*, β^*, γ^* and λ^* with non-negativity constraints. Evading EN is similar and can be done by setting the target distribution $\bar{\boldsymbol{p}}$ to the uniform distribution:

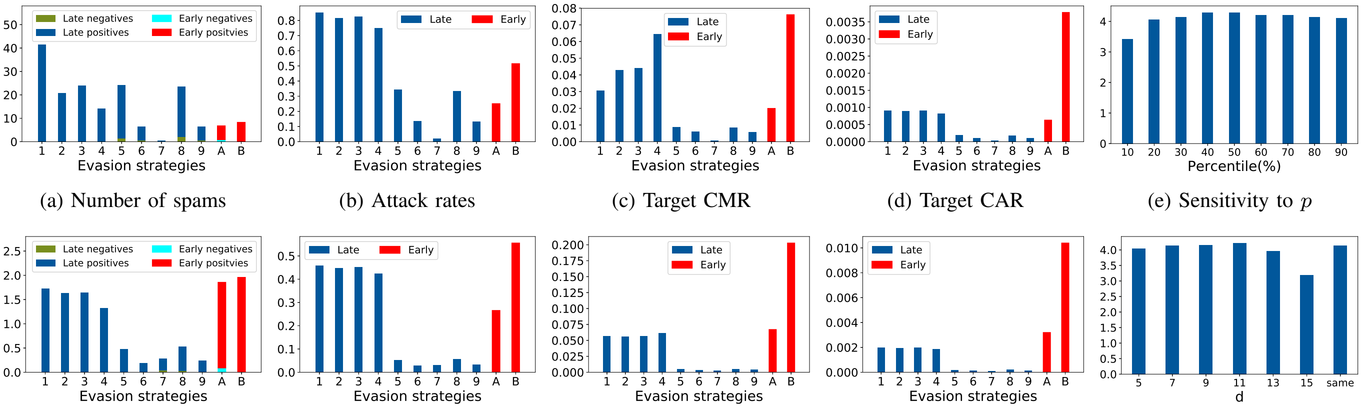
The above optimization problem assumes that the number of spamming reviews (n_δ) and the target CAR value ($\tilde{x}_t$) are given. We further set these parameters to evade ΔCAR, CAR-DEV and ΔNR that look for abrupt changes in the corresponding time series. By assuming that the defender adopts a degree d AR model $\boldsymbol{\theta}$ to capture CAR deviation, the spammer sets δ_t, the increment in CAR in window t, to meet the following goals: 1) increase CAR from x_t to $\tilde{x}_t = x_t + \delta_t$ so that $\tilde{x}_t$ is as high as possible; 2) evade the detection of CAR-DEV by ensuring $|\tilde{x}_t - \hat{x}_t| < \epsilon$, where ϵ is a small number; 3) gear the next AR model $\boldsymbol{\theta}^{(t+1)}$ to predicting a high CAR (denoted by $\hat{x}_{t+1}(\delta_t)$), to leave more room for δ_{t+1} to be added to x_{t+1}, so that $|x_{t+1}(\delta_t) + \delta_{t+1} - \hat{x}_{t+1}| < \epsilon$. The goals can be formulated as the following optimization problem:

$$\begin{aligned}
\max_{\delta} \quad & \delta + \hat{x}_{t+1}(\delta) \\
\text{s.t.} \quad & 0 \leq \delta, \quad |\hat{x}_t - (x_t + \delta)| < \epsilon, \quad x_t + \delta < U,
\end{aligned} \tag{7}$$

where U is an upper bound of the time series ($U = 5$ for CAR), Assuming the spammer uses online gradient descent to train $\boldsymbol{\theta}$ with learning rate η, then Eq. (7) becomes the following constrained quadratic programming problem:

$$\boxed{\begin{aligned}
\max_{\delta} \quad & \left[1 + \theta_1^{(t+1)} + \eta \left(\boldsymbol{x}_{t-d}^{t-1}\right)^\top \boldsymbol{x}_{t-d+1}^{t}\right] \delta + \eta x_{t-1} \delta^2 \\
\text{s.t.} \quad & \max\{0, \hat{x}_t - x_t - \epsilon\} \leq \delta \\
& \min\{U - x_t, \hat{x}_t - x_t + \epsilon\} \geq \delta
\end{aligned}}$$

The optimal δ is denoted by δ_t^* and is used to set $\tilde{x}_t = x_t + \delta_t^*$ in Eq. (3). The spammer also wants to evade detection based on burst detection [55], [13]. If CAR goes up suddenly in window t, ΔCAR(t) is large and the spamming campaign is likely to be detected. We can add the constraint $|x_{t-1} - (x_t + \delta)| < \epsilon$, to reduce ΔCAR. To evade the detection of ΔNR, a spammer samples $n_\delta \geq 0$ ratings from the distribution obtained from Eq. (4), such that n_δ is below the p-percentile of all positive historical increments in NR. To optimize δ, the spammer needs to know both d, the degree of AR model used by the defender, and the p-percentile of the CDF (cumulative distribution function) of the detection signals used by the defender. We empirically show that the spammer has a wide range of choice for d and p to conduct effective and evasive spamming campaigns.



(a) Number of spams (b) Attack rates (c) Target CMR (d) Target CAR (e) Sensitivity to p

(f) Number of spams (g) Attack rates (h) Target CMR (i) Target CAR (j) Sensitivity to d

Fig. 2: First 4 columns (top: Amazon, bottom: Yelp) show, from left to right: the number of total/negative/positive spams posted, successful rate of evasion, average promotions in CMR and CAR under late and early evasions. Last column: sensitivity of ranking promotion per spam on Amazon regarding parameters p and d of the defender (other metrics are similar on both datasets).

Algorithm 1 EMERAL: Evasion via Maximum Entropy and Rating sAmpLingi (for a target with a long rating history)

```
Input: Reviews of a target; maximum number of trials $M$.
Output: Ratings of spamming reviews to be posted.
Select $n_\delta$ and $\delta_t^*$ based on historic reviews.
Set rating distribution: $p_5 = 1$ and $p_i = 0, i = 1, \dots, 4$.
if Evade rating distribution-based signals then
    Use $n_\delta$ and $\delta_t^*$ to solve problem (4) to find $\boldsymbol{p}(t)$, with options of evading one of {KL-DIV, KL-DIV+ΔEN, KL-DIV+NPR, EN, EN+ΔEN}.
    Exit without spams if no evasive rating distribution found.
end if
while Not succeed and number of trials $< M$ do
    Sample $n_\delta$ ratings from $\boldsymbol{p}(t)$ satisfying constraints $\delta_t^*$.
    Return sampled ratings if no constraint violated.
end while
Exit without spams.
```

The overall evasion procedure EMERAL is described in Algorithm 1. EMERAL requires the target to have a reasonably long history of reviews to calculate the evasion parameters. Note that the algorithm may fail to find an evasive spamming plan for a window, and in that case, the spammer will not attempt to attack. Note also that EMERAL can create spamming plans to evade all combination of the detecting signals. However, evading certain subsets of signals, especially the larger subsets result in over-constrained optimization problem that produce no feasible evasion plans (no spams can be posted without evading the specified combination of detection signals), as confirmed by our experiments (not shown in this paper). We only focus on the evasion of 9 particular combinations of detection singals, denoted by E1 to E9 [1].

A running example of late spamming a spammer is trying to promote an app (id:B00G5LQ5MU) in a week while bypassing detection. First, it uses historical ratings of the app to calculate the CDF of each signal. It finds that the 80% percentile of the CDF of NR is 893 reviews, which allows it to post 685 reviews on top of 208 normal reviews. If the spammer is concerned about the large number of spams, it can lower the 80% percentile to a lower number and the ranking promotion is not much affected (see Figure 2e). Second, it finds that the 80% percentile of the CDF of CAR(ϵ) is 0.0022. Third, it plugs ϵ into Eq. (7) to get $\delta_t^* = 0.0027$. If the spammer does not care about rating distribution, it can at most post 59 5-star reviews to promote CAR without exceeding 0.0027. If the spammer also wants to evade KL-DIV, it needs to post some reviews with other ratings. It can plug $n_\delta = 685$ and $\delta_t^* = 0.0027$ into Eq. (4) to get the evasive rating distribution p. Lastly, it samples from p to get a specific number of spamming ratings. In this case, 9 1-star, 13 2-star, 102 3-star, 224 4-star and 337 5-star ones, will be posted.

B. EMERAL for early spamming: evasion E-A and E-B

It is shown that dishonest businesses have a strong motivation to conduct promotional spamming early on when their products are open for review [26], [40]. We adapt EMERAL to generate evasive spams for such situations. A new product will have a smaller number of reviews for a spammer to probe evasion parameters from the CDFs of the signals. However, a spammer can leverage the CDFs of the signals based on the early reviews of other products and estimate the evasion parameters. In particular, a spammer can obtain NR, ΔNR, ΔCAR and rating distribution of the early time windows of all available products, and then tries maximize the entropy while safisfying constraints over NR, ΔNR and ΔCAR (E-A). Evasion E-B tries to post a maximum number of 5 star reviews to evade NR, ΔNR and ΔCAR at the same time.

C. Empirical properties of EMERAL on late spamming

We use datasets collected from Amazon and Yelp — two popular websites hosting millions of opinion reviews regarding products and restaurants. These datasets are used in previous spam detection literature [17], [40]. To spam targets with long

[1] The 9 combinations are: E1=[NR], E2=[NR, ΔNR], E3=[NR, CAR-DEV], E4=[NR, CAR-DEV, ΔCAR], E5=[NR, CAR-DEV, KL-DIV], E6=[NR, CAR-DEV, KL-DIV, ΔEN], E7=[NR, CAR-DEV, KL-DIV, NPR], E8=[NR, CAR-DEV, EN], E9=[NR, CAR-DEV, EN, ΔEN]

review histories, we filter products on Amazon with less than 1000 reviews or having less than 37 weeks of reviews, and restaurants on Yelp having less than 37 months of reviews (a month or a week is referred to as a "time window"). Due to space limit, we did not study other cutoffs, which are believed to generate similar results. The results are 383 products with 1175088 reviews on Amazon, and 327 restaurants with 247117 reviews on Yelp. The evasions are created on each target for the last 5 consecutive time windows based on knowledge obtained from all previous time windows (32 in total). We compute evasive spamming campaigns with strategies E1 to E9, assuming that the spammer aims to keep each detection signal lower than the 80 percentiles of the corresponding signals' CDFs after the campaigns.

The average numbers of total/negative/positive spams posted in all test windows by each evasion on the two datasets are shown in Figure 2a and 2f. One can observe that all evasions post much more positive spams than negatives to promote business ratings and rankings. Interestingly, if a spammer decides to evade rating distribution related signals, as with evasions E5 to E9, some negative reviews have to be posted, while with evasions E1 to E4, there is no negative reviews. Since EMERAL does not guarantee that an evasive rating distribution can be found, Figures 2b and 2g show the percentages of windows that an evasion is possible. Evasions 1-4 are successful in most of the windows (more than 70%) while Evasions 5-9 are more conservative due to constraints over rating distribution. Figures 2c and 2d on the top row show the promotions in the target's CMR and CAR per spamming review, averaged over all targets and test windows, on the Amazon dataset. We can see that evasions 5-9 are less profitable to the spammers as the promotions are rather small, and evasions 1-4 can promote the target rather effectively. We tried to evade other combinations of the signals using EMERAL but found out that it is hard to find evasions valuable to the spammers. As a result, the defender needs not to consider evasions against other combination of signals in Table I.

D. Empirical properties of EMERAL on early spamming

The early windows of the datasets that are not used for late spamming are used for early evasions. In the same set of figures for late spamming, we use the last two bars in each subfigure to demonstrate the properties of early evasive spamming. In Figures 2a and 2f, we can see that average numbers of total/negative/positive spams post in each early windows. There is not large difference in the total number, but E-A creates a small amount of negative reviews due to entropy maximization. In Figures 2b and 2g, we can see that E-B has a successful rate two times of the rate of E-A, leading to higher per spam utility in CMR and CAR promotions, shown in Figures 2c, 2h, 2d and 2i. We conclude that early spamming is very attractive to spammers and advanced defense against early spamming need to be deployed, as we will do next.

Spammer knowledge requirements From Section III-A, for evasive late spamming, it seems that the spammer needs to know the degree the AR model and the p-percentile of the CDF of historic CAR to find out ϵ for solving Eq. (8). For evasion E4, which requires d and p, the two parameters can be selected from wide ranges so that they can be different from the values used by the defender. Figures 2e and 2j show the ranking promotion brought by E4 after detection based on the spammers' assumptions is not much affected by the inaccurate knowledge of these two hyper-parameters. For example, different d values achieve similar spamming utility as when d is the same as the value set by the defender. There is no parameter d in early spamming.

IV. A GAME THEORETICAL ANALYSIS

The above discussions do not consider the situations where a motivated spammer can probe the detection system to learn about any deployed detection algorithm and then select the best evasion strategy. The defender can also run EMERAL to identify its weakness and look for better defense, given that the defender knows what the spammer knows. We find the optimal strategies of both parties in this a non-cooperative game.

Spammers' assumptions It is relatively easy for a spammer to learn about a few simple defense strategies, including the individual detection signals and how to combine them for detection. In general, following [40], a defender estimates the cumulative distribution function CDF_k ($\text{CDF}_k(x) = \text{Prob}(X_k \leq x)$) of the signal X_k, $1 \leq k \leq K$, shown in Table I. For the value x_k of the k-th signal, compute

$$f_k = \begin{cases} 1 - \text{CDF}_k(x_k), & \text{if the higher the more suspicious} \\ \text{CDF}_k(x_k), & \text{otherwise} \end{cases} \tag{8}$$

A lower f_k value indicates a more suspicious time window. For each time window of a business, all the signal values $x_1, \ldots, x_K$ are calculated and tranformed to $f_1, \ldots, f_K$, which are combined using $\mathbf{w} = [w_1, \ldots, w_K] \in \mathbb{R}_+^K$: $S = 1 - \sqrt{\sum_{k=1}^K w_k f(x_k)^2 / \sum_{k=1}^K w_k}$. Windows are then ranked by their S scores for detection.

$\mathbf{w}$ controls the contribution of each signal to the suspicious scores, and different $\mathbf{w}$ may lead to various defense strategies. A spammer with moderate knowledge can assume that the k-th signals is used and set w_k to be much higher than the remaining ones Denote this detector by $\mathbf{w}^k$. There are also "aggregated detectors": $\mathbf{w}^a$ sets all the weights to be the same, $\mathbf{w}^m$ assigns the maximum of $1 - f(x_k), k = 1, \ldots, K$ to each window, and $\mathbf{w}^r$ assigns random weights to the signals.

Assumed measure of evasion success The spammer's goal is to promote a target business' reputation. A reasonable spamming utility is the difference in the target's reputation before and after the spamming campaign, after some spams are detected and removed. The spammer would assume a positive correlation between window suspicious score and spam removal rate, and want to avoid high suspicious score to maintain the spamming effect. This assumption is realistic as any reasonably effective detectors shall positively correlate suspicious score with spam removal rate. Here the spammer is only approximating the true defense strategy and is thus different from the previous work [47], [50], [3], [34], [21] that assume the adversarial knows the *true* defender's objective for evasion evaluation. In particular, the spammer removes a certain percentage between 20% and 80% of the spams in a window, linearly dependent on the suspiciousness of the window. After spam removal, we calculate the increase in CAR, CMR, and the business' ranking among non-target businesses based on CMR:

$$\Delta\psi(t) = \frac{\max\{\tilde{\psi}(t) - \psi(t), 0\}}{\#\{\text{Spam posted in window } t\}}, \tag{9}$$

where $\tilde{\psi}$ is CAR/CMR/ranking after the spammer posts the spamming reviews and defender removes spams, and ψ is the same metric before spamming. The normalization leads to the unit spam utility.

Game theoretical optimal evasion Let the spammer's action space $\mathcal{S}$ be the 9 evasions (E1 to E9) and the defender's action space $\mathcal{D}$ be the 13 defenses, in the order of CAR-DEV, ΔCAR, ΔNR, ΔEN, NR, EN, PR, KL-DIV, ΔPR, maximal atomic signal, a logistic classifier trained on evasion datasets generated with the corresponding evasion strategy, uniform weighting the signals, and randomly selecting a signal. A player chooses one action in a window. A pure evasion (defense) strategy i (j) is when the spammer (defender) selects Evasion i (Defense j). The pair (i, j) is called a strategy profile. The spammer utility function is $u_s(i, j) : \mathcal{S} \times \mathcal{D} \mapsto \mathbb{R}$, mapping from the profile (i, j) to a spamming utility $\Delta\psi$. Since the spammer does not know the *true* defender objective (to maximize both recall and precision, or just one of them), the spammer can assume that $u_d(i, j) = -u_s(i, j)$, leading to a zero-sum game. We will measure a more realistic defender utility function in the next Section from the defender's perspective. A Nash equilibrium is a strategy profile (i^*, j^*) such that $u_z(i^*, j^*) \geq u_z(i^*, j)$ and $u_z(i^*, j^*) \geq u_z(i, j^*), \forall z = s, d$, $\forall j \neq j^*$ and $i \neq i^*$. Evasion i^* is a dominating strategy if $u_s(i^*, j) > u_s(i, j)$ for all i and j. Thatch is, evasion i^* is best for the spammer regardless of defense strategy.

For each strategy profile, we compute the spammer's utility, shown in Figure 3. We found that Evasion 4 is the optimal atomic evasion strategy. In particular, on both datasets, with lower window suspicious scores and fewer spams caught, Evasion 4 (against NR, CAR-DEV and ΔCAR) generates more spams while effectively bypassing all the defense strategies. On the Amazon dataset, by a large margin, Evasion 4 outperforms the runner-up (Evasion 3) by additionally evading ΔCAR, indicating that ΔCAR is critical to evade. Evasions 5-9 are not helpful to the spammers due to the small number of spams generated and the camouflaging negative reviews. In sum, Evasion 4 finds the spammer the right amount of positive reviews to promote CAR without getting caught.

V. DETER: Evasion agnostic defenses

The defender may assume a fixed evasion strategy that is optimal for the spammer and then devise a detection model accordingly. For example, based on the above analysis, the defender can assume that a rational spammer will only use Evasion 4 to spam late review windows. In reality, multiple spammers can choose different evasion strategies and a spammer can change its strategy as well. We propose such a defense called DETER (Defense via Evasion generaTion using EmeRal) that works well regardless of which evasion strategy is adopted by the spammers.

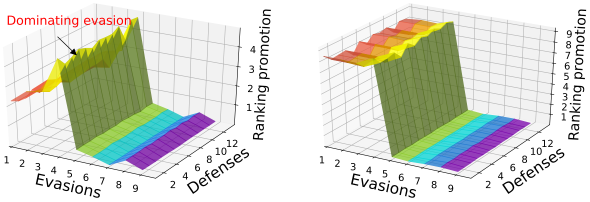


Fig. 3: Utility of the evasions under different detection strategies (Left: Amazon. Right: Yelp). Evasion 4 is the dominating one: no matter what detection strategies the defender adopts, the spammer gains the most by using Evasion 4. The CMR and CAR metrics have similar results.

DETER is based on defense model re-training [47], [31], [25], [22]. However, the re-training relies on a evasion generator, which is not available without EMERAL. For a target with long review history (with more than 30 windows), earlier windows (the first 30 windows) are used to train an EMERAL model, which generates 9 types of evasions on later windows (after the 30-th windows, group 1) of the targets. For early spamming, all targets are partitioned into two subsets. We pool all early windows (the first 30 ones) of the targets in the first subset together to train a single EMERAL model, which generates two types of early evasions (Section III-B) on all early windows of the targets in the second subset. Detection signals are computed for each window where evasion are attempted. Labels are assigned accordingly ("spammed" (or "not spammed") if EMERAL finds an evasion (or fails to generate any spam)). For the two groups of windows with attempted evasion, respectively, the defender pools the labeled windows from all targets for all evasion types within the group to train a logistic regression model (using sklearn with the default hyper-parameters) to detect windows spammed with unknown evasion strategy during late or early review periods.

A. *Effectiveness of DETER for late spam detection*

Suspicious window detection AUC is used as the defender's metric [2]. As two baselines, the defender can train a classifier using data obtained from each evasion type, and during testing, detect spammed windows by pooling all classifier outputs using the MAX or AVG function (denoted by EN_M and EN_A, short for ENsemble Max and ENsemble Average, respectively). DETER, EN_M and EN_A are all based on retraining and agnositic about evasion strategies, as opposed to $\mathbf{w}^i, i = 1, \ldots, 9$, $\mathbf{w}^a$, $\mathbf{w}^m$ and $\mathbf{w}^r$, which are only best for a single evasion strategy.

Two randomized evasion strategies are created to confirm that DETER works without knowing the evasion strategy. The first one ("Rand1") assumes that each window is spammed

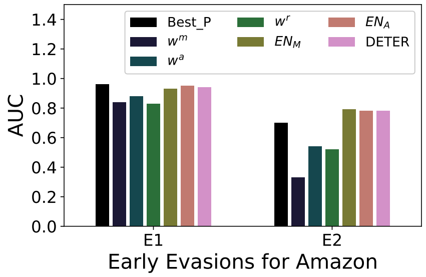
[2] the defender cares about both precision and recall, while a spammer cares only about promotion effect, which is related to recall but not precision.

TABLE II: Detection AUC under different strategy profiles with late/early spamming. Top: Amazon, buttom: Yelp.

(a) AUC of detection of late spamming on Amazon (rows: evasions, columns: defenses).

Evasions	Best_P	$\mathbf{w}^m$	$\mathbf{w}^a$	$\mathbf{w}^r$	EN_M	EN_A	DETER
Rand1	0.91	$0.91_{\pm 0.007}$	$0.91_{\pm 0.007}$	$0.89_{\pm 0.008}$	$0.74_{\pm 0.007}$	0.90	$\mathbf{0.93}_{\pm 0.006}$
Rand2	0.86	0.84	0.84	$0.83_{\pm 0.006}$	$0.73_{\pm 0.011}$	$0.84_{\pm 0.006}$	$\mathbf{0.87}_{\pm 0.006}$
E1	0.91	0.85	0.86	$0.84_{\pm 0.007}$	0.91	0.89	**0.92**
E2	0.90	0.84	0.85	0.83	**0.91**	0.88	0.90
E3	**0.89***	0.82	0.83	$0.81_{\pm 0.008}$	0.87	0.85	0.88
E4	**0.75***	0.67	0.69	0.67	0.71	0.72	0.74
E5	0.97	0.96	0.95	0.94	0.71	0.97	**0.98**
E6	0.97	0.93	0.92	$0.90_{\pm 0.007}$	0.62	0.97	**0.98**
E7	**0.98***	$0.96_{\pm 0.005}$	0.96	$0.95_{\pm 0.011}$	$0.72_{\pm 0.011}$	$0.97_{\pm 0.007}$	0.98
E8	0.97	0.96	0.96	0.94	0.72	0.97	**0.98**
E9	**0.96***	0.92	0.91	$0.89_{\pm 0.014}$	$0.52_{\pm 0.009}$	$0.92_{\pm 0.006}$	0.96

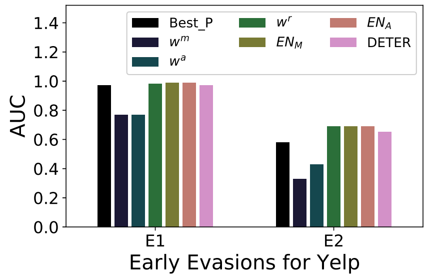
(b) AUC of detection of early spamming on Amazon



(c) AUC of detection of late spamming on Yelp (rows: evasions, columns: defenses).

Evasions	Best_P	$\mathbf{w}^m$	$\mathbf{w}^a$	$\mathbf{w}^r$	EN_M	EN_A	DETER
Rand1	$0.77_{\pm 0.012}$	$0.71_{\pm 0.012}$	$0.73_{\pm 0.011}$	$0.71_{\pm 0.016}$	$0.67_{\pm 0.015}$	$0.70_{\pm 0.017}$	$\mathbf{0.80}_{\pm 0.015}$
Rand2	$0.73_{\pm 0.013}$	$0.65_{\pm 0.012}$	$0.67_{\pm 0.012}$	$0.65_{\pm 0.016}$	$0.73_{\pm 0.044}$	$0.73_{\pm 0.051}$	$\mathbf{0.76}_{\pm 0.016}$
E1	0.73	0.63	0.66	$0.65_{\pm 0.007}$	**0.82**	0.81	0.78
E2	0.73	0.63	0.66	0.65	**0.82**	0.81	0.78
E3	0.73	0.63	0.65	$0.64_{\pm 0.007}$	**0.81**	0.80	0.77
E4	0.69	0.58	0.61	$0.59_{\pm 0.006}$	**0.78**	0.77	0.71
E5	**0.99***	0.97	0.96	$0.95_{\pm 0.011}$	$0.71_{\pm 0.009}$	$0.93_{\pm 0.009}$	$0.95_{\pm 0.007}$
E6	**0.99***	$0.95_{\pm 0.008}$	$0.95_{\pm 0.008}$	$0.94_{\pm 0.008}$	$0.75_{\pm 0.007}$	$0.93_{\pm 0.007}$	0.97
E7	**0.99***	0.95	0.95	$0.93_{\pm 0.009}$	0.73	0.95	0.98
E8	**0.99***	0.96	0.96	$0.95_{\pm 0.012}$	$0.74_{\pm 0.008}$	$0.95_{\pm 0.007}$	0.97
E9	**0.99***	0.94	0.94	$0.92_{\pm 0.019}$	$0.70_{\pm 0.012}$	$0.90_{\pm 0.007}$	0.95

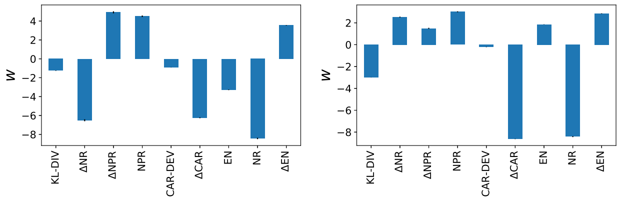
(d) AUC of detection of early spamming on Yelp



with one of the 9 pure strategies with equal probability, and the second ("Rand2") assumes that half of the windows are spammed with Evasion 4, while the remaining windows are spammed with the other strategies with equal probability. Overall, there are 11 evasion strategies (9 pure: E1 to E9, plus 2 mixed: Rand1 and Rand2) and 15 defense strategies (9 pure: $\mathbf{w}^i, i = 1, \ldots, 9$, plus $\mathbf{w}^m$, $\mathbf{w}^a$, $\mathbf{w}^r$, EN_M, EN_A and DETER), resulting in 11×15 strategy profiles. Rand1, Rand2, E5 - E9 are randomized algorithms and we repeat each evasion and detection for 10 times, and the means of the AUCs under each strategy profile are reported in Table II. Standard deviation of AUC greater than $5e-3$ are reported as the subscripts of the means. Evasion strategies E1 - E4 are deterministic and only one experiment is needed. Due to space limit, we show the best AUC of $\{\mathbf{w}^i, i = 1, \ldots, 9\}$ (Best_P).

From the table, we have the following observations. First, under strategies Rand1 and Rand2, DETER has the highest AUC than all the remaining defenses. Among the agnostic defenses, by averaging, EN_A is the runner-up beating EN_M, indicating that taking the maximum of the output is a reasonable defense but can be over-sensitive. Second, Best_P is always better than $\mathbf{w}^m$, $\mathbf{w}^a$ and $\mathbf{w}^r$, and we conclude that if the defender knows the exact evasion strategy, it can pick a single detection signal, rather than guessing using $\mathbf{w}^r$, which is inferior to DETER. Third, under E3, E4, E7 and E9 on the Amazon dataset, and E5 to E9 on the Yelp dataset, Best_P outperforms all agnostic strategies (indicated by bold fonts with asterisks). However, such performances are based on the unrealistic assumption that all windows are spammed with the specific evasion strategies, and cannot be achieved in reality. According to Figure 2, E5 to E9 are not effective in promoting target reputations (unprofitable for spammers) and a spammer is less likely to select them, although DETER outperforms or is comparable to Best_P. The take-away is that, by evasive spamm generation, data pooling and detection model retraining, a defender can achieve state-of-the-art detection performance.

One may question the security of DETER: what if a spammer reads this paper and then implements and evades DETER? Figure 4 shows the weights learned by DETER over the 9 detection signals on two datasets. We can see that CAR-DEV is not used much by DETER, but ΔCAR and NR are always active to prevent the dominating evasion E4. With other few medium weights watching rating distribution entropy, it would be quite difficult for a spammer to evade this set of detection signals, while evading a larger set of signals will significantly reduce reputation promotion (see Section IV especially Figure 3). The strategy profile consisting of the trained DETER model and any evasion strategies is a Nash equilibrium when the defender aims at detection AUC and spammers aim at promotion.



(a) Amazon

(b) Yelp

Fig. 4: Weights over detection signals learned by DETER.

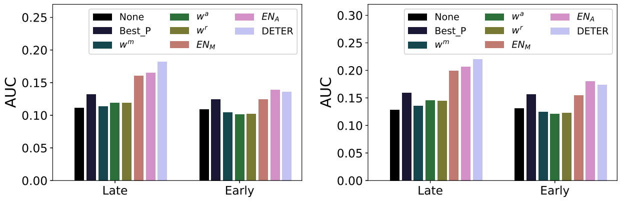


Fig. 5: Detection of spams in early and late evasion on two Yelp datasets. (Left:YelpChi, right: YelpNYC)

B. Effectiveness of DETER for early spam detection

For early spamming, the spammer can choose from two evasion strategies, E-A and E-B. As shown in Figures IIb and IId, when dealing with evasion type E-A, the 3 adaptive detectors based on EMERAL and re-training (EN_M, EN_A and DETER) have comparable or even better performance than the best pure detection strategy (Best_P). When dealing with E-B, these three detectors significantly outperform Best_P. In sum, EMERAL provides sufficient knowledge about a wide spectrum of spams to vaccinate the defender in the face of whatever evasion strategy.

C. Effectiveness of DETER for spamming review detection

We adopt the state-of-the-art spam detector, SpEagle [40], which combine the features of the reviews, reviewers and products with the reviewer-product graph. We show that the window suspicious scores generated by those window detectors based on re-training can help SpEagle identify individual spamming reviews. We run evasion E4 for late spamming and evasion E-B for early spamming on YelpChi and YelpNYC datasets [40], respectively, generating spams to be detected by SpEagle. The above evasions provide the rating distributions of the spams in test windows, and the actual spams are posted by a random subset of the existing accounts at some randomly picked time during the test window. To rank reviews based on their suspicious scores, we multiply the review posteriors produced by SpEagle by the suspicious score of the window where the review sits in. The detection AUC are shown in Figure 5. It is clear that those window detectors based on re-training using EMERAL (EN_M, EN_A and DETER) outperform the remaining ones. In particular, DETER outperforms EN_M and EN_A in the late spamming cases and is comparable to EN_A in the early spamming cases.

VI. Related Work

Opinion spams are different from social spams [20], [16], [30], [58], web spams [54], email spams [41] in terms of spamming goal and detection mechanism, and we focus on opinion spams. Graph-based approaches leverage the relationships between reviewer accounts, reviews and products to detect spams, suspicious accounts and dishonest businesses [51], [32], [40], even with evasive camouflages [18] Text-based approaches identify spamming reviews based on the contents of the reviews, using linguistic features and psychological features [39], topic model [42], semantic analysis [27], etc. Behavior-based approaches [55], [13], [59], [26], [33], [23] look for abnormal patterns in the the volume and distribution of user ratings, which are complementary to graphs and texts based approaches. To the best of our knowledge, no previous work has considered generative models for evading and securing behavior-based opinion spam detection.

Randomized defenses help to obfuscate the details of the defender and prevent attackers from taking advantage of any static defense strategies [6], [50]. DETER does not need randomization for privacy purpose, since it prevents the spammers from creating campaigns that are *both* evasive and effective. If privacy is indeed a concern when using DETER, randomization can be implemented via differential privacy [7]. Randomized evasion is handled by DETER, demonstrated by two randomized evasion strategies.

Generating adversarial examples is critical to secure and robust machine learning models: if the models can see (foresee) most/all of the adversarial examples during training [31], [47], then during test time, most adversarial examples crafted by the attackers can be correctly detected. Adversarial example can be generated in either feature spaces [34], [21], [50] or problem spaces [50], [57], [44]. Generation in the feature space usually admits a convex and differentiable optimization problem whose solutions can be efficiently found as adversarial examples. However, the generated vectors usually cannot be mapped to realistic examples in the problem space and often tend to be over-pessimistic. Example generation in problem spaces requires domain knowledge and usually involves non-convex and non-differentiable optimization problems. The work here is the first step towards rigorous, efficient and realistic adversarial spam generation in the problem space.

Game theory has been used in secure machine learning [6], [34], [5]. They assume that the attacker and defender know each other's objective function and try to use game theory to arrive at a Nash equilibrium so that both parties do not seek other solutions. We use the concept of game theory to analyze the behaviors of a rational and well-informed spammer, instead of using game theory to find a secure defense solution. In fact, a Nash equilibrium may be too strong an assumption in the context of spam detection, as multiple spammers can adopt different strategies or a spammer may have no knowledge about the defender's strategy.

VII. Conclusion

We proposed a flexible and general computational evasion model ("EMERAL") against state-of-the-art spam detection techniques for both early and late stage review periods of the targets. The spamming campaigns generated are effective in reputation manipulation and detection evasions, and require only public available datasets and published detection methods without knowing the exact hyper-parameter values. EMERAL does not require differentiable models or heuristic search. We showed that a spammer can only evade a handful of signals but has a dominating evasion strategy representing the worst case for the defender. We considered more realistic scenarios with mixtures of evasion strategies, and devised DETER, an evasion-agnostic defenses based on model retraining. Experiments showed that data pooling is the best defense, among other ensemble methods.

References

[1] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. ICWSM, 2013.

[2] BBC. Samsung probed in taiwan over 'fake web reviews'. http://www.bbc.com/news/technology-22166606.

[3] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. ECML/PKDD, 2013.

[4] Battista Biggio, Giorgio Fumera, and Fabio Roli. *Multiple Classifier Systems for Adversarial Classification Tasks*. 2009.

[5] Michael Bruckner and Tobias Scheffer. Nash equilibria of static prediction games. NIPS, 2009.

[6] S. Rota Bul, B. Biggio, I. Pillai, M. Pelillo, and F. Roli. Randomized prediction games for adversarial machine learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[7] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. NIPS, 2009.

[8] Yizheng Chen, Yacin Nadji, Athanasios Kountouras, Fabian Monrose, Roberto Perdisci, Manos Antonakakis, and Nikolaos Vasiloglou. Practical attacks against graph-based clustering. CCS, 2017.

[9] Judith A. Chevalier and Dina Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354, 2006.

[10] Deepak Chinavle, Pranam Kolari, Tim Oates, and Tim Finin. Ensembles in Adversarial Classification for Spam. CIKM, 2009.

[11] C. Dellarocas, N. Awad, and M. Zhang. Using online ratings as a proxy of word-of-mouth in motion picture revenue forecasting. 2005.

[12] Wenjing Duan, Bin Gu, and Andrew B. Whinston. The dynamics of online word-of-mouth and product salesan empirical investigation of the movie industry. *Journal of Retailing*, 84(2):233 – 242, 2008.

[13] Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Exploiting burstiness in reviews for review spammer detection. ICWSM, 2013.

[14] Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. Distributional footprints of deceptive product reviews. ICWSM, 2012.

[15] Anindya Ghose, Panagiotis G. Ipeirotis, and Beibei Li. Examining the impact of ranking on consumer behavior and search engine revenue. *Management Science*, 60(7):1632–1654, 2014.

[16] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @spam: The underground on 140 characters or less. CCS, 2010.

[17] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. WWW, 2016.

[18] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *KDD*, 2016.

[19] Nan Hu, Jie Zhang, and Paul A. Pavlou. Overcoming the j-shaped distribution of product reviews. *Commun. ACM*, 52(10):144–147, October 2009.

[20] Xia Hu, Jiliang Tang, and Huan Liu. Online social spammer detection. In *AAAI*, 2014.

[21] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin I P Rubinstein, and J D Tygar. Adversarial Machine Learning. AISec, 2011.

[22] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. ICLR, 2015.

[23] Nitin Jindal and Liu Bing. Analyzing and detecting review spam. ICDM, 2007.

[24] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *WSDM*, 2008.

[25] Alex Kantchelian, J D Tygar, and Anthony D Joseph. Evasion and Hardening of Tree Ensemble Classifiers. ICML, 2016.

[26] Santosh KC and Arjun Mukherjee. On the temporal dynamics of opinion spamming: Case studies on yelp. WWW, 2016.

[27] Seongsoon Kim, Hyeokyoon Chang, Seongwoon Lee, Minhwan Yu, and Jaewoo Kang. Deep semantic frame-based deceptive opinion spam analysis. In *CIKM*, 2015.

[28] Theodoros Lappas, Gaurav Sabnis, and Georgios Valkanas. The impact of fake reviews on online visibility: A vulnerability assessment of the hotel industry. *Information Systems Research*, 27(4):940–961, 2016.

[29] Hady W. Lauw, Ee-Peng Lim, and Ke Wang. Bias and controversy: Beyond the statistical deviation. KDD, 2006.

[30] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: Social honeypots + machine learning. SIGIR, 2010.

[31] Bo Li, Yevgeniy Vorobeychik, and Xinyun Chen. A general retraining framework for scalable adversarial classification. NIPS Workshop on Adversarial Training, 2016.

[32] Huayi Li, Zhiyuan Chen, Bing Liu, Xiaokai Wei, and Jidong Shao. Spotting fake reviews via collective positive-unlabeled learning. In *ICDM*, 2014.

[33] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *CIKM*, 2010.

[34] Daniel Lowd and Christopher Meek. Adversarial Learning. KDD, 2005.

[35] Davide Maiorca, Igino Corona, and Giorgio Giacinto. Looking at the bag is not enough to find the bomb: An evasion of structural methods for malicious pdf files detection. ASIA CCS, 2013.

[36] Amanda J Minnich, Nikan Chavoshi, Abdullah Mueen, Shuang Luan, and Michalis Faloutsos. TrueView: Harnessing the Power of Multiple Review Sites. WWW, 2015.

[37] Arjun Mukherjee, Vivek Venkataraman, Bing Liu 0001, and Natalie S. Glance. What yelp fake review filter might be doing? ICWSM, 2013.

[38] Blaine Nelson, Benjamin I. P. Rubinstein, Ling Huang, Anthony D. Joseph, Shing-hon Lau, Steven J. Lee, Satish Rao, Anthony Tran, and J. Doug Tygar. Near-optimal evasion of convex-inducing classifiers. In *AISTATS*, 2010.

[39] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *ACL: HLT*, 2011.

[40] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*, 2015.

[41] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, 1998.

[42] Vlad Sandulescu and Martin Ester. Detecting singleton review spammers using semantic similarity. In *WWW*, 2015.

[43] Charles Smutz and Stavrou Angelos. When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors. In *NDSS*, 2016.

[44] Nedim Šrndic and Pavel Laskov. Practical Evasion of a Learning-Based Classifier: A Case Study. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, SP. IEEE Computer Society, 2014.

[45] David Stevens and Daniel Lowd. On the Hardness of Evading Combinations of Linear Classifiers. AISec, 2013.

[46] New York Times. Charges settled over fake reviews on itunes. http://www.nytimes.com/2010/08/27/technology/27ftc.html.

[47] Liang Tong, Bo Li, Chen Hajaj, and Yevgeniy Vorobeychik. Feature conservation in adversarial classifier evasion: A case study. 2017.

[48] TripAdvisor. Tripadvisor popularity ranking: Key factors and how to improve. https://www.tripadvisor.com/TripAdvisorInsights/w722.

[49] Catherine Tucker and Juanjuan Zhang. How does popularity information affect choices? a field experiment. *Management Science*, 57(5):828–842, 2011.

[50] Yevgeniy Vorobeychik and Bo Li. Optimal randomized classification in adversarial settings. AAMAS, 2014.

[51] Guan Wang, Sihong Xie, Bing Liu, and S Yu Philip. Review graph based online store review spammer detection. In *ICDM*, 2011.

[52] Guan Wang, Sihong Xie, Bing Liu, and Philip S. Yu. Identify online store review spammers via social review graph. *ACM Trans. Intell. Syst. Technol.*, 3(4):61:1–61:21, 2012.

[53] WebMD. Health care reform:health insurance & affordable care act. http://www.webmd.com/health-insurance/insurance-basics/using-doctor-ratings-sites.

[54] Baoning Wu and Brian D. Davison. Identifying link farm spam pages. WWW, 2005.

[55] Sihong Xie, Guan Wang, Shuyang Lin, and Philip S. Yu. Review spam detection via temporal pattern discovery. KDD, 2012.

[56] Qiongkai Xu and Hai Zhao. Using deep linguistic features for finding deceptive opinion spam. In *COLING*, 2012.

[57] Weilin Xu, Yanjun Qi, and David Evans. Automatically evading classifiers: A case study on pdf malware classifiers. NDSS, 2016.

[58] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y. Zhao, and Yafei Dai. Uncovering social network sybils in the wild. IMC, 2011.

[59] Junting Ye, Santhosh Kumar, and Leman Akoglu. Temporal Opinion Spam Detection by Multivariate Indicative Signals. In *ICWSM*, 2016.