E EXAMPLE

Consider a simple example where there are 10 products ($|\mathcal{P}| = 10$) and the SockFarm operator has 5 accounts to control ($|\mathcal{A}| = 5$)—moreover, $|T| = 10$. The initial state is $s_0 = (0, \{dp_{s_0}(a) \mid a \in \mathcal{A}\}, \emptyset, \emptyset)$ and $dp_{s_0}(a) = 0.2, \forall a \in \mathcal{A}$. The terminal states are states where either $t_s = 10$ or $dp_s(a) = 1$ for all $a \in \mathcal{A}$. The set of detected accounts at state s, i.e. $D(s)$, is empty at the initial state. The set of new requests added at state s, i.e. $\mathbf{REQ}^+(s)$, is also empty at the initial state.

After s_0 we move to state $s_1 = (1, \{dp_{s_1}(a) \mid a \in \mathcal{A}\}, \emptyset, \emptyset)$. No new requests have arrived and no reviews have been posted (because $A(s_0) = \emptyset$) so the detection probabilities remain unchanged, $\mathcal{TR}(s_0, \alpha = \emptyset, s_1) = 1$, $R(s_0, \alpha = \emptyset) = 0$, and $\pi(s_0) = \alpha = \emptyset$. Similarly, at $t = 2$, we reach $s_2 = (2, \{dp_{s_2}(a) \mid a \in \mathcal{A}\}, \emptyset, \emptyset)$ with everything else unchanged. At $t = 3$, the operator gets a request $\mathbf{r_1} = (p_1, 3, 5, 10)$ with probability 0.4. Then, s_2 with $\pi(s_2) = \alpha = \emptyset$ will transition to one of the following states:

$$s_{3,1} = (3, \{dp_{s_{3,1}}(a) \mid a \in \mathcal{A}\}, \{\mathbf{r_1}\}, \emptyset),$$
$$s_{3,2} = (3, \{dp_{s_{3,2}}(a) \mid a \in \mathcal{A}\}, \emptyset, \emptyset),$$

with $dp_{s_{3,1}}(a) = dp_{s_{3,2}}(a) = dp_{s_2}(a) = 0.2, \forall a \in \mathcal{A}$ and the following probabilities:

$$\mathcal{TR}(s_2, \alpha = \emptyset, s_{3,1}) = Pr(\{\mathbf{r_1}\}) \times 1 = 0.4,$$
$$\mathcal{TR}(s_2, \alpha = \emptyset, s_{3,2}) = (1 - Pr(\{\mathbf{r_1}\})) \times 1 = 0.6.$$

Consider now $s_{3,1}$ with policy $\pi(s_{3,1}) = \alpha_{3,1} = \{(a_1, p_1), (a_2, p_1), (a_3, p_1)\}$ and the conditional detection probabilities:

$$Pr((1.0, 0.6, 0.4, 0.2, 0.2) \mid (0.2, 0.2, 0.2, 0.2, 0.2), \alpha_{3,1}) = 0.3,$$
$$Pr((0.6, 0.6, 0.4, 0.2, 0.2) \mid (0.2, 0.2, 0.2, 0.2, 0.2), \alpha_{3,1}) = 0.7,$$

which are determined by the detection algorithm. Then we have the following reward function:

$$R(s_{3,1}, \alpha_{3,1}) = (0.3 \times 0 + 0.7 \times 10) - (0.3 \times 2 + 0.7 \times 0) - 3 = 3.4,$$

if the SockFarm operator posts each review with cost 1, has one detected account with cost 2 (with probability 0.3), and finishes the request with reward 10 (with probability 0.7).

At $t = 4$, we get $\mathbf{r_2} = (p_2, 3, 5, 10)$ with probability 0.4. Then, $\pi(s_{3,1}) = \alpha_{3,1}$ will make the operator move to one of the following states:

$$s_{4,1} = (4, (1, 0.6, 0.4, 0.2, 0.2), \{(p_1, 1, 5, 10), \mathbf{r_2}\}, \{(a_2, p_1), (a_3, p_1)\}),$$
$$s_{4,2} = (4, (1, 0.6, 0.4, 0.2, 0.2), \{(p_1, 1, 5, 10)\}, \{(a_2, p_1), (a_3, p_1)\}),$$
$$s_{4,3} = (4, (0.6, 0.6, 0.4, 0.2, 0.2), \{\mathbf{r_2}\}, \{\emptyset\}),$$
$$s_{4,4} = (4, (0.6, 0.6, 0.4, 0.2, 0.2), \emptyset, \emptyset\}),$$

where request $\mathbf{r_1}$ in $s_{4,1}$ and $s_{4,2}$ has not been finished because a_1 is detected, and a_1 can not be used anymore. The transition function is:

$$\mathcal{TR}(s_{3,1}, \alpha_{3,1}, s_{4,1}) = 0.4 \times 0.3 = 0.12,$$
$$\mathcal{TR}(s_{3,1}, \alpha_{3,1}, s_{4,2}) = 0.6 \times 0.3 = 0.18,$$
$$\mathcal{TR}(s_{3,1}, \alpha_{3,1}, s_{4,3}) = 0.4 \times 0.7 = 0.28,$$
$$\mathcal{TR}(s_{3,1}, \alpha_{3,1}, s_{4,4}) = 0.6 \times 0.7 = 0.42.$$

Consider now the following policies:

$$\pi(s_{4,1}) = \alpha_{4,1} = \{(a_4, p_1), (a_3, p_2), (a_4, p_2), (a_5, p_2)\}$$
$$\pi(s_{4,2}) = \alpha_{4,2} = \{(a_4, p_1)\}$$
$$\pi(s_{4,3}) = \alpha_{4,3} = \{(a_3, p_2), (a_4, p_2), (a_5, p_2)\}$$
$$\pi(s_{4,4}) = \alpha_{4,4} = \emptyset.$$

Suppose the detection probability for each account is not changed after taking these actions at the corresponding state. Then the requests are finished at the corresponding state with the following rewards:

$$R(s_{4,1}, \alpha_{4,1}) = 10 + 10 - 0 - 4 = 16,$$
$$R(s_{4,2}, \alpha_{4,2}) = 10 - 0 - 1 = 9,$$
$$R(s_{4,3}, \alpha_{4,3}) = 10 - 0 - 3 = 7,$$
$$R(s_{4,4}, \alpha_{4,4}) = 0.$$

After that, the operator gets no more requests, and will not post any reviews in the subsequent states.

Now suppose that, after $s_{3,2}$, $\mathbf{r_2}$ is finished successfully, and $V(s_{3,2}) = 2.8$. Then, the expected reward for the policy π is (with $\gamma = 1$):

$$\begin{aligned} V(\pi) = & R(s_0, \alpha_0) + R(s_1, \alpha_1) + R(s_2, \alpha_2) + 0.6 \times V(s_{3,2}) \\ & + 0.4 \times R(s_{3,1}, \alpha_{3,1}) \\ & + 0.4 \times (0.12 \times R(s_{4,1}, \alpha_{4,1}) + 0.18 \times R(s_{4,2}, \alpha_{4,2}) \\ & \quad + 0.28 \times R(s_{4,3}, \alpha_{4,3}) + 0.42 \times R(s_{4,4}, \alpha_{4,4})) \\ = & 0 + 0 + 0 + 0.6 \times 2.8 + 0.4 \times 3.4 \\ & + 0.4 \times (0.12 \times 16 + 0.18 \times 9 + 0.28 \times 7 + 0.42 \times 0) \\ = & 1.68 + 1.36 + 0.4 \times (1.92 + 1.62 + 1.96 + 0) \\ = & 5.24. \end{aligned}$$

F ADDITIONAL EXPERIMENTAL RESULTS

Figure 3 reports the F1-Scores obtained when we assumed 1000 total incoming requests. The results confirm the effectiveness of both SockAttack and SockDef:

- In 61 of 64 cases, SockAttack successfully reduced the F1-Scores of existing RFDs than the best baseline attack (Baseline-0.8).
- In all 64 cases, applying SockDef under SockAttack produced better F1-Scores.

On average, the impact of SockAttack and SockDef on F1-Scores was as follows:

- The SockAttack caused F1-Scores of baselines to drop by 21.6% on the Alpha dataset, 27.1% on OTC, 17.6% on Amazon, and 10.3% on Epinions.
- Using SockDef against the SockAttack increased F1-Scores by 69.2% on the Alpha dataset, 30.5% on OTC, 34.5% on Amazon, and 16.1% on Epinions.

Figure 4 reports the F1-Scores obtained with the STARS attack proposed in [16] compared with SockAttack (with 100 total incoming requests). The results show that, in 62 of 64 cases, SockAttack successfully reduced the F1-Score of the STARS attack. On average,

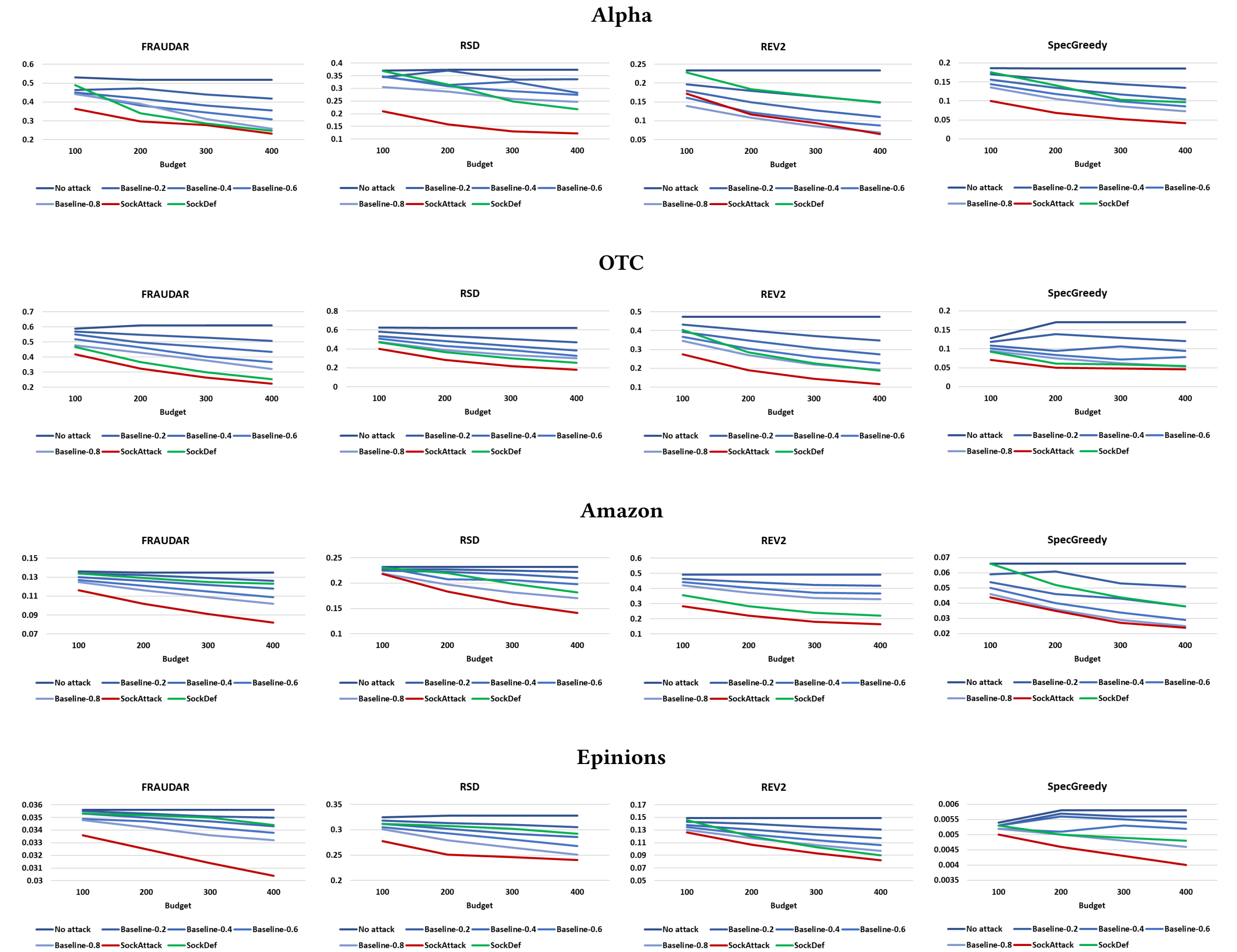

Figure 3: F1-Scores obtained with 1000 total incoming requests. Baseline-x means baseline attack with $\gamma = x$. The SockDef algorithm is evaluated under the SockAttack attack.

the SockAttack caused F1-Scores to drop by 25.6% on the Alpha dataset, 13.5% on OTC, 17.5% on Amazon, and 7% on Epinions.

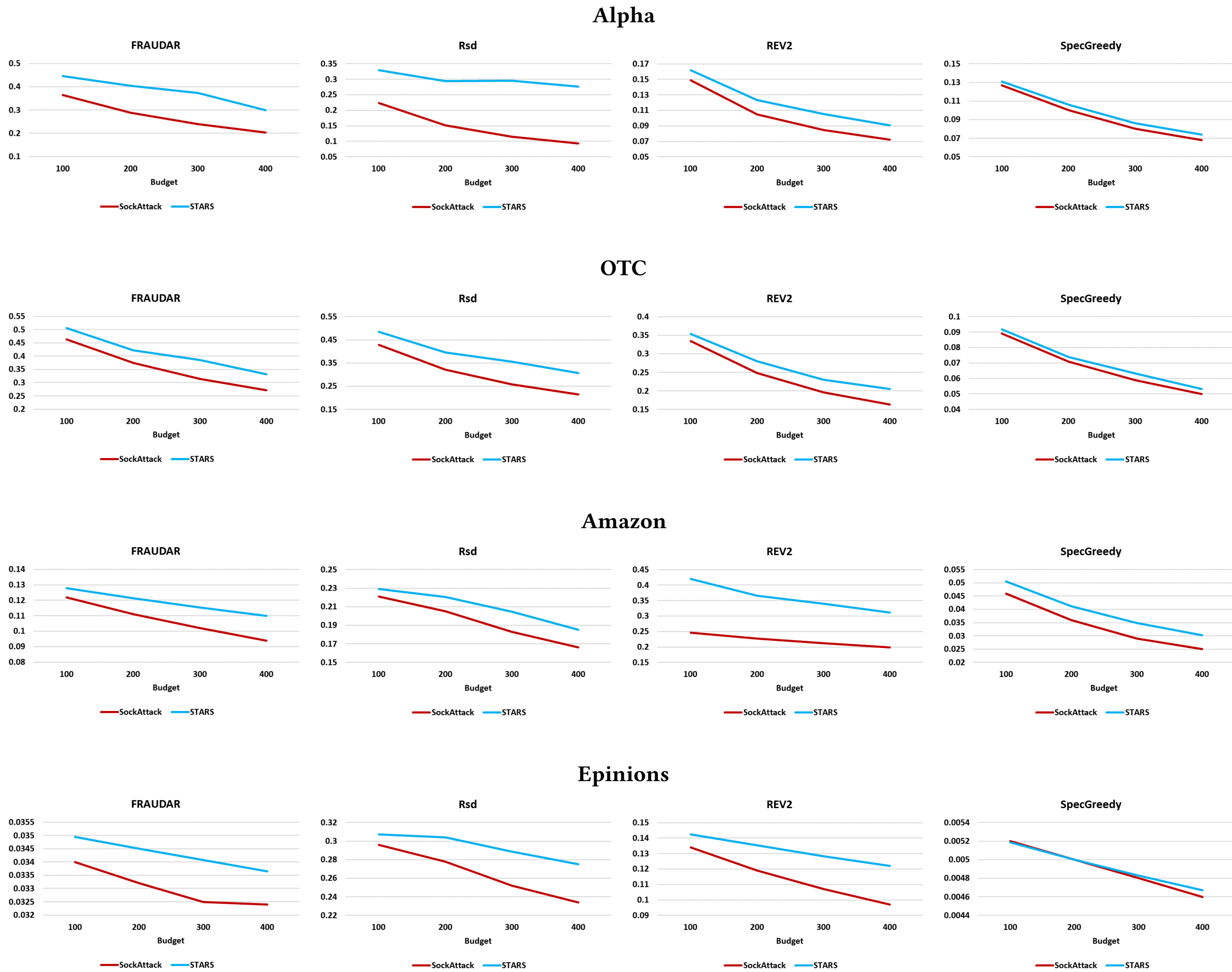

Figure 4: F1-Scores obtained by SockAttack and STARS.