A. Theoretical Analysis

Let the time needed for a single round optimization is t. For simplicity, we further assume the number of rounds of optimization needed to generate the final trigger for the target label is R and the objective function has p probability choosing the target label. Let $p_s = (1-\epsilon)\cdot p + \epsilon \cdot \frac{1}{K}$ be the probability the target label is scheduled.

Efficiency. Since the selective optimization terminates only when the target label is optimized $R-2$ times, it follows the *Negative Binomial Distribution* (Eggenberger & Pólya, 1923) that models the probability of the number of failure events before a given number of successful events happen, when the probability of one successful event is given. The expected time cost of K-Arm is hence the following.

$$\mathbb{E}[T_{km}] = 2\cdot K\cdot t + \frac{(R-2)\cdot t}{(1-\epsilon)\cdot p + \epsilon\cdot\frac{1}{K}} \tag{6}$$

The first term is the time for the warm-up phase in which all the K labels go through 2 rounds of optimization. The second is the time for the selective optimization. The denominator is the probability of choosing the right label. From the equation, We have the following observations.

- When $R \gg K$, such as for TrojAI round 1 models (i.e., $R = 50$ and $K = 5$). The cost is dominated by the second term. Therefore, we have $\mathbb{E}[T_{km}] = \mathcal{O}(R\cdot t)$. Since the cost for NC is $\mathbb{E}[T_{nc}] = \mathcal{O}(K\cdot R\cdot t)$, the speed-up over NC is determined by K.
- When $R \ll K$, e.g., in ImageNet models with $K = 1000$. The cost is dominated by the first term. $\mathbb{E}[T_{km}] = \mathcal{O}(K\cdot t)$ and the speed-up is determined by R.

Effectiveness. We analyze the effectiveness of our method by comparing with NC and NC+pre-selection the likelihood of finishing optimizing the target label within a time bound. The analysis is done by comparing the expected time of finishing optimizing the target label. Note that if the time bound is fixed, the smaller expected value means a higher probability of finishing successfully.

NC vs. K-Arm. Since NC optimizes all labels in order, the expected finishing time is the following.

$$\mathbb{E}[T_{nc}] = R\cdot t\cdot(1\cdot\frac{1}{K}+2\cdot\frac{1}{K}+...K\cdot\frac{1}{K}) = \frac{(K+1)\cdot R\cdot t}{2}$$

In practice, due to the objective function design, the probability of K-Arm scheduling the target label $p_s = (1-\epsilon)\cdot p + \epsilon\cdot\frac{1}{K}$ is usually much higher than $2/K$ when K is not small. Together with Eq. (6), we have $\mathbb{E}[T_{km}] < 2\cdot K\cdot t + \frac{(R-2)\cdot t}{\frac{2}{K}} = \frac{K\cdot R\cdot t}{2} + K\cdot t < \mathbb{E}[T_{nc}]$. Note R is usually larger than $2\cdot K$.

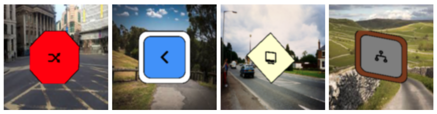


Figure 7. Example images from TrojAI datasets

NC+pre-selection vs. K-Arm. NC+pre-selection makes deterministic decision to select the m smallest triggers after the initial optimization. If the target label is not among the m smallest, pre-selection will never succeed. In practice, the failure probability is not low. Here, we only focus on comparing K-Arm with NC+pre-selection when the target label is among the m smallest. We have the expected time of pre-selection $\mathbb{E}[T_{ps}] = 2\cdot K\cdot t + \frac{(m+1)(R-2)t}{2}$, similar to $\mathbb{E}[T_{nc}]$. When $p_s > \frac{2}{m}$ (which holds in practice), following the reasoning similar to above, we have $\mathbb{E}[T_{km}] < \mathbb{E}[T_{ps}]$.

B. Details of TrojAI Competition Datasets

Round1 Dataset. The round1 training set contains 1000 CNN models for classification tasks, in which 532 models are trojaned and 468 are benign. Each model has 5 labels and IARPA provides 100 labeled clean images with size 224x224x3 for each class. A clean image is generated by combining a foreground object and a background image. The foreground objects are traffic signs with different shapes. The background images are road scene data drawn from KITTI (Fritsch et al., 2013), Cityscapes (Cordts et al., 2016) and Swedish Roads (Larsson & Felsberg, 2011). Note that these samples were not used to train the models, but drawn from the same distribution. Sample images are shown in Fig. 7. There are 3 different model architectures for round1 models: ResNet-50 (He et al., 2016), Inception-v3 (Szegedy et al., 2016), DenseNet-121 (Huang et al., 2017). There are only universal triggers in the round1 trojaned models. The triggers are polygons with 3 to 12 sides and a randomly selected color. In each malicious image, a trigger is stamped on an unknown area inside the foreground object. The size of trigger varies from 2 $\sim$ 24% of the foreground object. Fig 8 illustrates the generation process of trojan images.

Round2 Dataset. The round2 training set contains 1104 CNN models for classification tasks, with 552 trojaned models and 552 benign models. Compared to round1, round2 models have more labels ranging from 5 $\sim$ 25. The clean images provided for each label are fewer (20 per class). It includes universal triggers, label specific triggers, and also Instagram filter triggers. There are 23 different model architectures. More description related to the TrojAI datasets can be found in (IARPA, 2020).

Round3 Dataset. The round3 training set contains 1008 CNN models for image classification tasks with 504 trojaned models and 504 benign models. Same as round2, the

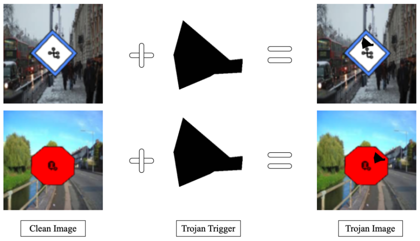
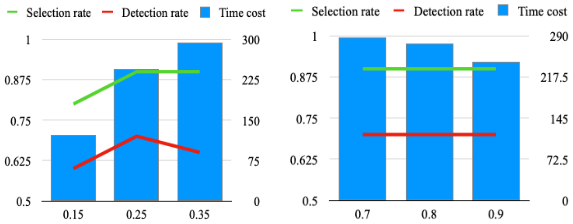


Figure 8. Trojan Image Generation

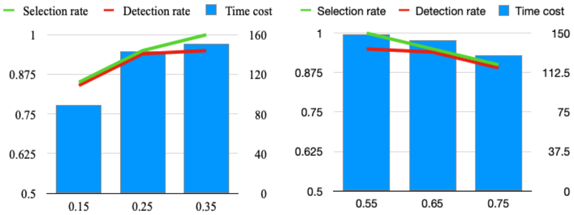


(a) γ Comparison (b) θ Comparison

Figure 9. Label-specific trigger detection under different hyper-parameters.

number of classes for each model is $5 \sim 25$, and the clean images provided for each label are $10 \sim 20$. Different from round2 models, all round3 models are enhanced through adversarial training (Madry et al., 2017; Wong et al., 2020). The adversarial attack has 3 different strength levels based on the perturbation size $(\frac{4}{255}, \frac{8}{255}, \frac{16}{255})$ and 2 different levels based on the ratio $(0.1, 0.3)$, i.e. what percentage of the batches are attacked. The number of iterations used in PGD attacks is set as 4 different values $(2, 4, 8, 16)$. More details can be found in (IARPA, 2020).

Round4 Dataset. The round4 training set contains 1008 CNN models with 504 trojaned models and 504 benign models. As the most challenging round, round4 models have more classes ($15 \sim 44$), less samples ($2 \sim 5$ per class). Unlike previous rounds, round4 models can have many concurrent conditional triggers. Such triggers can cause the misclassification only when they fulfill the conditions. There are three different conditions: spatial, spectral and class. The spatial trigger requires the trigger exists within a certain area to cause the misclassification behaviour. The spectral trigger can only lead the misclassification when the trigger has certain color. The class context requires the trigger must be stamped on the correct class. Besides, the universal triggers are removed in round4. There are only label specific triggers. Such comprehensive settings make the backdoor detection more difficult. Table 5 summarizes the configurations cross all trojAI 4 rounds.



(a) γ Comparison (b) θ Comparison

Figure 10. Universal trigger detection under different hyper-parameters.

C. Impact of Hyper-parameters

From Fig. 11a, we observe that K-Arm has stable detection accuracy and time cost in a large range of β (from 10^2 to 10^6). When β is small, K-Arm might get stuck with a few labels that seem promising (based on the objective function). Thus the time cost slightly increases. From Fig. 11b, when ϵ is large, K-Arm pays more attention to exploring random labels, which leads to more time consumption. From Fig. 11c , when τ is small, many real (back-door) triggers are considered benign, causing accuracy degradation. When τ is in 300-500, we can achieve a stable high accuracy (around 91%) to distinguish the trojaned and benign models.

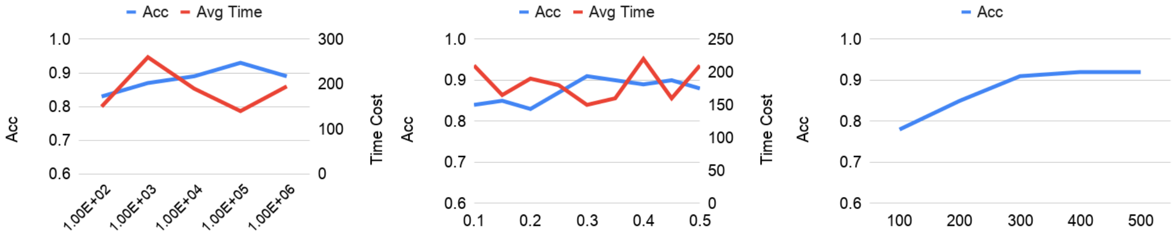
We evaluate the effect of remaining two hyper-parameters θ and γ. Recall that θ and γ are used in the arm pre-processing phase. In particular, we consider a label promising if its logits value ranks among the top $\gamma\%$ labels in at least $\theta\%$ of all benign samples of a label (for label-specific trigger scanning) or various labels (for universal trigger scanning). Intuitively, γ should be small and θ should be large. For scanning universal triggers, we set 3 different values for γ $(15, 25, 35)$ and 3 different values for θ $(55, 65, 75)$. For scanning label specific triggers, we test the same values of γ and choose θ from $(70, 80, 90)$. Given 20 randomly selected round2 models with global triggers and 20 with label specific triggers, we report the accuracy for selecting the correct target label successfully under different settings, the average time cost and the detection accuracy. From Fig. 9a and Fig. 10a, we can see that a small γ value causes some target labels omitted as the arm size is reduced. This further leads to detection accuracy degradation. On the other hand, when γ is large, although the selection rate increases, the time cost goes up. Compared to γ, arm pre-processing is less sensitive to θ. From Fig. 9b and Fig. 10b, the detection accuracy and time cost are more stable with different θ values.

D. Study of K-Arm Failing Cases

In this section, we study 2 K-Arm failing cases and explain the reasons.

Table 5. TrojAI Dataset

Rounds	# of Models	# of Classes	# of Samples per Class	# of Model Architectures	# of Triggers	Global Trigger	Label-specific Trigger	Polygon Trigger	Instagram Filter Trigger	Adv.Training
Round1	1000	5	100	3	1	✓	✗	✓	✗	✗
Round2	1104	5∼25	10∼20	23	1	✓	✓	✓	✓	✗
Round3	1008	5∼25	10∼20	23	1	✓	✓	✓	✓	✓
Round4	1008	15∼44	2∼5	16	1∼2	✗	✓	✓	✓	✓

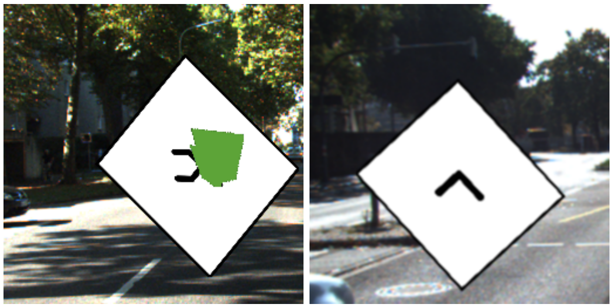


(a) β Comparison (b) ϵ Comparison (c) τ Comparison

Figure 11. K-Arm accuracy and time cost under different parameter value settings

Case I: Pre-screening fails to select the correct target-victim pair. According to the Figure 9a, the pre-screening can not achieve 100% selection accuracy. Therefore, for some trojaned models, the correct victim-target pair is filtered out during the pre-selection stage and cause the detection fail. For instance, model #18 in round4 is a trojaned model with a label-specific polygon trigger. The victim label is 14 and target label is 8. When we apply the pre-screening by the default setting ($\gamma = 25, \theta = 90$) on this model, we find that 13 out of 342 pairs are selected. However, the right pair is not in the list. In fact, there are only 60% samples from the victim label, in which the target label's logits value rank on the top 25% among all labels. Since the right pair is pruned out, the following K-Arm optimization cannot find a trigger smaller than the threshold τ and report the model as benign.

Case II: Symmetric K-Arm fails when victim and target labels are similar. Recall that the Symmetric K-Arm performs the trigger optimization in two opposite directions and considers the ratio of objective functions to distinguish the real trigger and natural features. However, when the ground truth trigger is stamped on a victim class which is similar to the target label, Symmetric K-Arm will avoid selecting such a pair to optimize due to the small ratio, and eventually it causes the detection to fail. Figure 12 shows the victim label#13 image stamped with trigger and target label#12 image for model#22 in round2. As shown in the figure, the victim class is very similar to the target class. The sign at the center of the image is the only difference between the two classes. Fig. 13 further illustrates the trigger size variation in two opposite directions. We can see that the trigger sizes reduce in the same pace for both directions. Therefore, the ratio of objective functions is closed to 1. This pair can rarely be selected to optimize in K-Arm. In this case, K-Arm actually selects the victim-target pair (#5-#1) in most rounds, and eventually reports the model as benign since the optimized trigger is larger than τ.



(a) Victim + Trigger (b) Target

Figure 12. R2 model#22

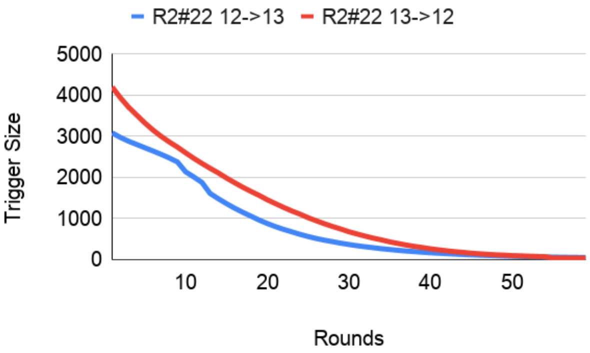


Figure 13. Trigger size variation in two opposition directions.