Towards Certifiable Adversarial Sample Detection

Ilia Shumailov
ilia.shumailov@cl.cam.ac.uk
Computer Laboratory, University of Cambridge

Yiren Zhao
yiren.zhao@cl.cam.ac.uk
Computer Laboratory, University of Cambridge

Robert Mullins
robert.mullins@cl.cam.ac.uk
Computer Laboratory, University of Cambridge

Ross Anderson
ross.anderson@cl.cam.ac.uk
Computer Laboratory, University of Cambridge

ABSTRACT

Convolutional Neural Networks (CNNs) are deployed in more and more classification systems, but adversarial samples can be maliciously crafted to trick them, and are becoming a real threat. There have been various proposals to improve CNNs' adversarial robustness but these all suffer performance penalties or have other limitations. In this paper, we offer a new approach in the form of a certifiable adversarial detection scheme, the Certifiable Taboo Trap (CTT). This system, in theory, can provide certifiable guarantees of detectability of a range of adversarial inputs for certain l_∞ sizes. We develop and evaluate several versions of CTT with different defense capabilities, training overheads and certifiability on adversarial samples. In practice, against adversaries with various l_p norms, CTT outperforms existing defense methods that focus purely on improving network robustness. We show that CTT has small false positive rates on clean test data, minimal compute overheads when deployed, and can support complex security policies.

ACM Reference Format:
Ilia Shumailov, Yiren Zhao, Robert Mullins, and Ross Anderson. 2020. Towards Certifiable Adversarial Sample Detection. In *13th ACM Workshop on Artificial Intelligence and Security (AISec'20), November 13, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411508.3421381

1 INTRODUCTION

Convolutional Neural Networks (CNNs) give the best performance on visual applications [1, 24, 40] and are now being used in safety-critical applications, including autonomous vehicles [11], face recognition [42] and human action recognition [20]. However, perturbations can be crafted to trigger misclassifications that are not perceptible by humans [15]. Researchers have demonstrated adversarial samples that can exploit face-recognition systems to break into smartphones [4] and misdirect autonomous vehicles by perturbing road signs [10]. These adversarial samples can be surprisingly portable. Samples generated from one classifier transfer to others, making them a potentially scalable threat to real-life systems.

AISec'20, November 13, 2020, Virtual Event, USA

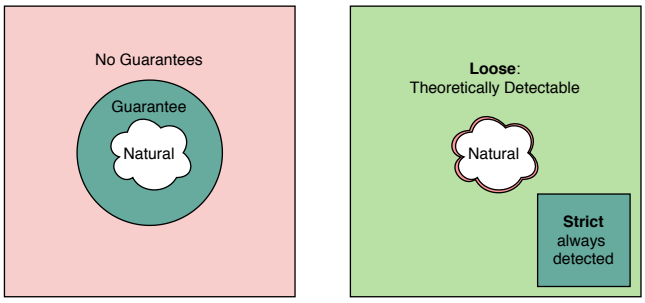
ACM ISBN 978-1-4503-8094-2/20/11…$15.00
https://doi.org/10.1145/3411508.3421381

Since most of these attacks use neural network gradient information to generate perturbations [15], the obvious defense is to improve the networks' classification robustness, such as by training classifiers with adversarial images. Such *adversarial training* significantly increases the performance of CNNs on adversarial samples but falls short in three ways. First, it assumes the defender has prior knowledge of the attacks; second, the defense is not certifiable; third, building a fully robust model is still an unsolved question [41]. In this paper, we look at a different defense strategy, namely adversarial sample detection. Researchers have shown that many adversarial samples are detectable, and detection methods can be built without prior knowledge of attacks [33, 46]. We built on the existing Taboo Trap detection scheme [46], whose focus is on finding overly excited neurons being driven beyond a pre-defined range by adversarial perturbations. We propose a mechanism, the Certifiable Taboo Trap (CTT), that combines the original Taboo Trap detection with numerical bound propagation, making the detection bounds on CNN activation values certifiable against certain input perturbation sizes. For input perturbations at a particular l_∞ value, CTT can verify detection, meaning that CTT guarantees the detected samples are adversarial inputs. As illustrated in Figure 1, certifiable detection provides a new angle to the problem of provable defense guarantees for adversarial samples compared to existing certifiable robustness research.

In this paper, we propose three versions of CTT: lite, loose and strict. CTT-lite requires no additional fine-tuning on a pretrained model, and can provide basic protection against weak adversaries. CTT-loose retrains on a random set of selected activations with propagated numerical interval bounds, and provides a loose guarantee that all samples detected are adversarial. Finally, CTT-strict fine-tunes with stricter numerical interval bounds, and thus is able to provide the same guarantee as CTT-loose on attackers with small l_∞ values; in addition, CTT-strict can verify detection on a pre-defined range of l_∞ values.

The contributions of this paper are:

- We introduce a novel certifiable detection scheme for adversarial samples.
- We release an open-source implementation with fully reproducible results.



(a) Certifiable Robustness with IBP

(b) Certifiable Detection with CTT

Figure 1: An intuitive comparison between certifiable robustness and detection. Certifiable robustness builds an l_p norm ball of provable performance around natural data samples and provides no guarantees outside of this ball. Certifiable detection on the other hand builds a very small l_p norm ball around natural data where we guarantee no detection. We also guarantee, for samples outside the ball, that CTT-loose detection is possible and CTT-strict detection will always happen in the declared range. The presented shape for CTT-strict is arbitrary; any regions that are non-overlapping with natural samples can be certified.

- We introduce *CTT-lite*, a new detection method that is fine-tuning free but is limited in its defense capability. We demonstrate how to optimise detection boundaries through fine-tuning, then introduce *CTT-loose* and *CTT-strict*, both detection schemes ensure that all detected samples are adversarial. CTT-strict guarantees detections on adversarial samples with a particular range of pre-defined l_∞ bounds.
- We provide detection results on all versions of CTT. For the first time, we empirically demonstrate how certifiable detection schemes (CTT-loose and CTT-strict) can have above 90% detection ratios on all attacks experimented on MNIST.

2 RELATED WORK

The field of adversarial machine learning has seen a rapid co-evolution of attack and defense since the discovery of adversarial samples [47]. The fast gradient sign method (FGSM) is an early adversarial attack that generates perturbations using the signs of the network gradients, and is still a simple yet effective way of finding adversarial samples [15]. The FGSM attack can be extended in an iterative way to look for smaller perturbations, giving the Projected Gradient Descent (PGD) Method or the Basic Iterative Method (BIM) [27, 32]. Many alternatives of BIM exist and focus on minimizing the l_1 or l_2 distances. DeepFool [36] is a more advanced iterative attack that linearizes misclassification boundaries of the network at each iteration and moving along the direction that gives the nearest misclassification. The Carlini & Wagner attack (CW) has improved the adversarial sample search by directly optimizing the difference between correct and incorrect logits [6]. However, a strong adversarial image, *i.e.* an adversarial sample satisfying a number of strong constraints such as perturbation size or classification confidence, is time-consuming to generate since it requires a large number of search iterations and binary search steps. Many of the attacks can change their optimization focus or be constrained on certain l_p norms in an iterative run. In our setup, we use the term l_p-bound attack to differentiate the same attack bounded by various l_p norms.

An interesting feature of adversarial samples is their transferability [15, 47, 53]. Adversarial samples that work well on a given neural network often transfer to a different type of network trained to solve a similar task. This makes black-box attacks possible. Another way of finding black-box attacks is using estimated instead of true gradients [2]. Estimation involves building an output distribution based on information queried from the target model.

Many defenses against adversarial attacks have been proposed, most of which aim at improving classification robustness. Adversarial training adds adversarial samples to the training process, helping the model to learn how to deal with an attacker [15, 27]. Pang et al. use an ensemble of models to increase decision robustness [38], while Mustafa et al. use class-wise disentanglement to restrict feature maps crossing the decision boundaries [37]. However, Schott et al. showed that even building robust classification on the small MNIST data remains an unsolved problem [41]. They also proposed the analysis and synthesis (ABS) method using class-conditioned data and demonstrate better robustness on the MNIST classification task.

Many researchers have tried to detect adversarial samples [31, 33, 34, 43]. Magnet detects adversarial samples by inspecting the reconstruction error of a trained autoencoder [33]. SafetyNet proposed SVM classifiers to recognize adversaries through neural activation patterns [31]. However, both of these detection methods rely on auxiliary components, which have two main problems. First, they impose a significant computational overhead. Second, an adversary might obtain a copy of the defense and devise an adversarial sample to defeat it [5, 7].

Another efficient detection scheme is the Taboo Trap [46], where a random subset of neurons are constrained in training and an alarm is set off when some threshold of them become overly excited. This imposes no extra runtime computational cost, and the constrained subset of neurons can be picked randomly, giving a key that can be different each time the network is trained. This makes Black-box attacks more challenging as there can be multiple independently-keyed networks each of which is vulnerable to different adversarial samples [44]. Our work builds on the Taboo Trap, and answers the question of how to make adversarial sample detection certifiable. It also establishes the optimal numerical range limit on neurons, and significantly improves the detection performance of Taboo Trap.

Our work can also be viewed as being related to certifiable robustness where the prediction of a data point x is verifiably constant with perturbations of a certain l_p norm. When queried with the input data x, x will be perturbed by isotropic Gaussian noise and multiple inference runs are executed on a base classifier f [8, 30], in this way, the returned classification provides the most probable prediction made by f with a Gaussian corrupted x. Meanwhile, certification of adversarial samples can be achieved using bound interval propagation, which is becoming established as a means of formal verification of neural networks [9, 13, 16, 21, 35, 48, 49].

Several prior works have studied efficient relaxation methods for computing tight bounds on the neural network outputs [48, 49]. Our Certifiable Taboo Trap uses bound-interval propagation, but its focus is on certifying out-of-bound values in a set of randomly sampled intermediate activations. The interval bounding is a simple integral bound so the computation overhead is minimised [35].

3 METHOD

3.1 Taboo Trap: A Practical View

The method shown here extends the Taboo Trap originally presented in [44, 46]. First, we will explain the Taboo Trap method and then demonstrate the extension made for producing a relaxed guarantee that a certain l_∞ bound attacker will always be detected.

Taboo Trap is based on the idea that neural network activations can be trained with extra regularisations to bound a set of activations inside a certain numerical range. No training set inputs drive this chosen set of activation values out of range. So if such a 'taboo' activation is seen, it signals that the current input may be adversarial. As different instances of the model can be trained with different taboo sets, the authors coined a term of a *transfer function*, which essentially served as a neural network key. In the original Taboo Trap, Shumailov et al. made use of the *n*th-max percentile activation bounds profiled from a trained network [46]. They later used polynomial keys [44]. Yet, the detection rates reported were less than ideal: the *n*th-max percentile function only detects weak attackers, while polynomial-based detectors show good detection rates on transfer attacks but perform worse under direct attack.

The Taboo Trap authors hypothesised that its performance is related to the choice of transfer functions, yet could not explain why some attackers could not be detected. While their experiments show a practical ability to detect adversaries, there is little theoretical understanding of how and why it worked.

3.2 Taboo Trap: A Theoretical View

In this section, we provide a theoretical understanding of how operating in a high dimensional activation space can detect adversarial samples. Assume that we have a linear function $f(x) = ax + b$ for simplicity. The simple integral bound of the linear function with input bounded between x_{min} and x_{max} is bounded by $f(x_{\text{min}})$ and $f(x_{\text{max}})$.

Figure 2a presents how the original Taboo Trap will instrument function f with a nth max percentile transfer function. x_{min} and x_{max} represent the minimum and maximum values x can take. Since network inputs are bounded, the intermediate layers should receive inputs that are also bounded, regardless of non-linearities. Being monotonic functions, $f(x_{\text{min}})$ and $f(x_{\text{max}})$ present the minimum and maximum values that the function f can naturally assume. If T_{high} represents the Taboo Trap threshold; we have:

$$\begin{cases} f(x) \leq T_{\text{high}} & \text{Benign} \\ f(x) > T_{\text{high}} & \text{Malicious} \end{cases} \tag{1}$$

We define an adversarial sample $\hat{x} = x + \epsilon$, with its l_∞ norm having the size of ϵ. With different detection thresholds (T_{high}), we can have natural samples becoming false positives or adversarial samples becoming undetectable. Figure 2b shows the scenario when $T_{\text{high}} < f(x_{\text{max}})$: there exists a clean sample x with an output $f(x)$ being in between T_{high} and $f(x_{\text{max}})$. This causes natural samples to be misclassified as adversarial (false positives). Figure 2c presents the case that $T_{\text{high}} > f(x_{\text{max}})$: adversarial samples $\hat{x}$ can generate output $f(\hat{x})$ smaller than T_{high} so that it becomes undetectable by the Taboo Trap framework. In summary:

$$\begin{cases} T_{\text{high}} > f(x_{\text{max}}) & \text{Missed detection} \\ T_{\text{high}} < f(x_{\text{max}}) & \text{False positives} \end{cases} \tag{2}$$

Consider $r = |f(x_{\text{max}}) - T_{\text{high}}|$, it means

- if r equals to zero, the adversarial samples will always get detected.
- for a given r it is easy to compute what type and how many of perturbations will go undetectable.
- as mentioned by Shumailov et al. , there is a direct measurable trade-off between false positives, accuracy and detection rate.

Using the method defined above, it becomes apparent that all monotonic transfer functions should theoretically work in Taboo Trap, and have a trade-off between accuracy, false positive and detection rates.

Perturbations can also exist in the range between x_{min} and x_{max}. The original Taboo Trap paper observed that better detector performance is achieved by setting a small threshold value, yet training becomes hard. Our hypothesis is that reducing the distance between x_{min} and x_{max} leads to a reduced number of perturbations in the natural image range.

It is also worth noting that in this paper detection occurs on post-ReLU activation values, and only the positive numerical range and the positive numerical threshold (T_{high}) are considered. In practice both T_{high} and T_{low} can be used with other activation functions such as LeakyReLU. For simplicity, we use T_l to represent a layer-wise threshold scalar in later descriptions.

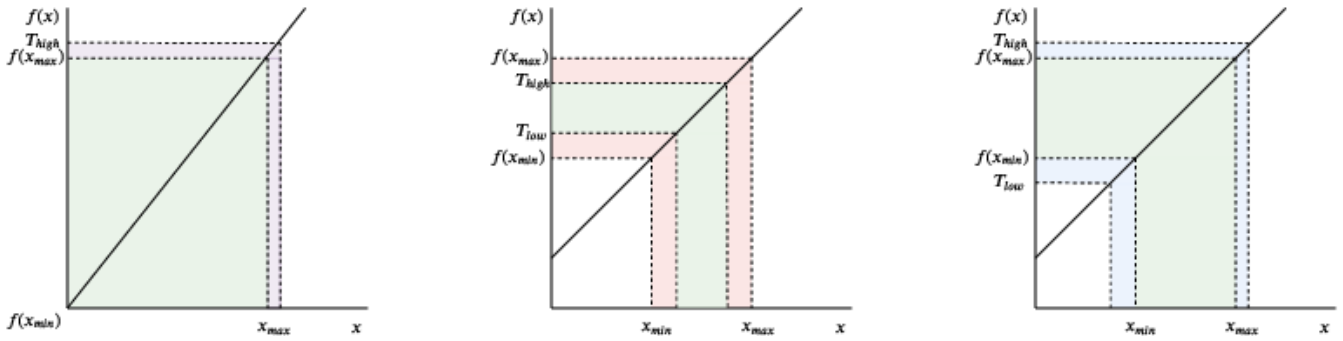
3.3 Interval Bound Propagation

For simplicity, we consider a feed-forward CNN F consisting of a sequence of convolution layers, where the l^{th} layer computes output feature maps $\mathbf{x}_l \in \mathbb{R}^{C_l \times H_l \times W_l}$. $\mathbf{x}_l$ is a collection of feature maps with C_l channels of $H_l \times W_l$ images.

The first stage of CTT is to compute the activation bounds from a pretrained network. In this paper we use Interval Bound Propagation (IBP) method [16]. Given the pretrained weights and the numerical bounds of inputs, CTT computes the numerical bounds for each layer in the CNN. Assuming the a set of lower and upper bound for layer l is $(B_l^{\text{low}}, B_l^{\text{up}})$, where B_l^{low} is the lower bound and B_l^{up} is the upper bound respectively, we have

$$\begin{aligned} B_{l+1}^{\text{low}} &= \text{Conv}_\text{b}(W_l, B_l^{\text{low}}) \\ B_{l+1}^{\text{up}} &= \text{Conv}_\text{b}(W_l, B_l^{\text{up}}) \end{aligned} \tag{3}$$

Notice B_0^{low} and B_0^{high} will be the boundaries on the input, obtained from profiling on the natural input data samples. Both B_l^{low}



(a) Original Taboo Trap. **(b) False positives (in red).** **(c) Undetectable range (in blue).**

Figure 2: Taboo Trap visualisation. If x is from the original data distribution, a strict bound causes them to be detected as adversarial samples, the red area shows the false positive samples (middle figure). If x is an adversarial sample, loose bounds cause detection fail on adversarial samples with small l_∞ values (blue part in figure on the right).

and B_l^{up} have the same dimensions as $\mathbf{x}_l$. The interval bound propagation in Equation (3) can be seen as a series of abstract interpolations in a convolution (Conv_b) [35, 49]. Given scalar bounds $m_l \leq m \leq m_h$ and $n_l \leq n \leq n_h$, we define an operation $(m_l, m_h) \cap (n_l, n_h)$ produces a tighter bound $p_l = \max(m_l, n_l)$ and $p_h = \min(m_h, n_h)$:

$$(m_l, m_h) \cap (n_l, n_h) = (\max(m_l, m_l), \min(n_h, n_h)) \tag{4}$$

This is equivalent to abstract interpolation in the interval domain (the box domain) [35]. Notice the bound propagation can also be performed for adversarial inputs, where the upper bound of the input becomes $\hat{B}_0^{\text{up}} = B_0^{\text{up}} + \epsilon$ and ϵ is the size of the l_∞ norm. We then define $(\hat{B}_l^{\text{low}}, \hat{B}_l^{\text{up}})$ to be a pair of upper bound and lower bound for layer l for adversarial inputs with an l_∞ budget of ϵ.

Considering the case in the middle layer l, we obtain a particular activation value x and its values across all input data distributions can be seen as a set X. Meanwhile, its values for all adversarial samples with an adversarial perturbation can be viewed as a set $\hat{X}$. For convenience, we call X the natural set and $\hat{X}$ the adversarial set. Figure 3 shows the placements of the detection threshold T_l. In the ideal case, if the distributions of the natural set (X) and the adversarial set ($\hat{X}$) are disjoint, the optimal placement of T_l is that $B_l^{\text{up}} <= T_l <= \hat{B}_l^{\text{low}}$. However, in practice, the natural set and the adversarial set might overlap (Figure 3b), meaning that there is only a sub-optimal placement option $B_l^{\text{up}} \leq T_l \leq \hat{B}_l^{\text{up}}$. For these two threshold placements, we conclude:

- Optimal placement of T_l ($B_l^{\text{up}} \leq T_l \leq \hat{B}_l^{\text{low}}$) ensures that all adversarial samples with l_∞ norm at the size of ϵ are detectable (Figure 3a).
- Both optimal and suboptimal placements ($B_l^{\text{up}} \leq T_l \leq \hat{B}_l^{\text{up}}$) of T_l ensure that all detected samples are adversarial regardless of the perturbation size (Figure 3b).

The above optimality claims are true if and only if the following assumption holds: *The test data distribution falls inside the training data distribution.* In other words, the test data fall in the range of the maximum and minimum bounds from the training dataset. Note that the above assumption does not mean that we mix training and test datasets. It only implies that there exists a perfect scenario in which there are no false positives and undetected adversarial samples. Yet in practice we find that we cannot place the threshold perfectly and because of that only capture a sub-set of adversarial examples. This could be a consequence of classification error, as it was previously shown that classification errors lead to existance of "local" adversarial samples [14]. We find that small increase in false positive rate helps networks to get significantly better at detecting adversarial samples, further strengthening the locality argument.

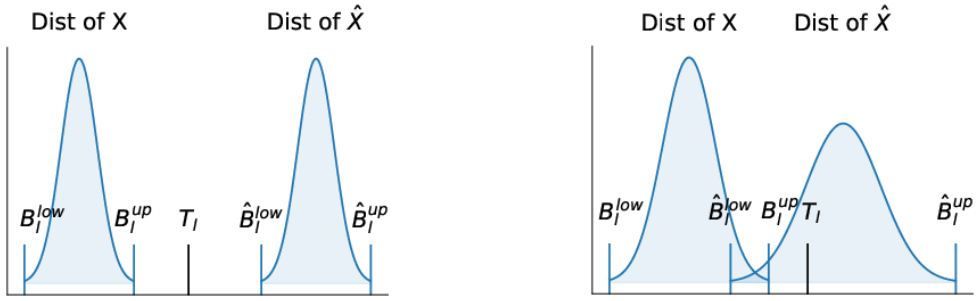
In Section 3.5, we show a training-free method of placing a non-optimal T_l. Section 3.6 discusses methodologies we used to ensure that placements of T_l are near-optimal.

3.4 Intuition Behind CTT

CTT is best understood in contrast with work on certifiable robustness such as that by Cohen et al. [8] and Gowal et al. [16]. Certifiable robustness aims at making natural sample behaviour stay in a pre-formed l_p norm ball, so that model behaviour is "stable" in a pre-defined range of ϵ values. CTT, on the other hand, aims at detecting illegal behaviours outside of the l_p ball, so the optimisation process of CTT encourages behaviour outside of the natural range to be more "unstable". In other words, CTT fine-tuning encourages the model to have a large Lipschitz constant, as opposed to regular robust training methods that decrease Lipschitz constant.

Figure 1 shows an intuitive comparison between Certifiable Robustness and Detection. Certifiable Robustness builds an l_p ball of provable performance around natural data samples and provides little to no guarantees on what happens outside of this ball. Certifiable Detection builds a very small ball around natural data where detection is impossible. With CTT-loose detection is possible, with CTT-strict samples in the predefined range will always be detected. Presented shape for CTT strict is arbitrary and we find that any regions non-overlaping with natural samples can be certified. It should be noted that certification of large continuous regions is hard as the volume of samples occupied by the region grows exponentially with number of dimensions.

The parameter ϵ can be thought of as a *detectability* certification of an adversarial sample. It defines the minimum theoretical perturbation size for which the detector *can* work. Rather than



(a) Optimal placement of T_l. **(b) Suboptimal placement of T_l.**

Figure 3: Placement of the detection threshold T_l with different boundaries from both the natural activations (X) and the adversarial activations ($\hat{X}$). This indicates that both the numerical value of T_l and the two distributions should be optimised using fine tuning.

generating adversarial samples as in adversarial training, we use natural samples perturbed by ϵ as in certifiable robustness. The CTT loss tries to ensure that when adversarial samples $X \pm \epsilon$ are considered, the detector neurons can be turned on.

The intuition is that the smaller you make natural sample l_p norm for CTT, the smaller the natural sample volume becomes. As the volume of natural samples becomes smaller, it becomes easier to detect adversarial behaviours. In practice, that should make CTT a lot less prone to invariance-based adversarial samples [18]. Instead of increasing the volume of natural samples as with certifiable robustness, certifiable detection reduces the volume of natural undetectable samples.

Algorithm 1 Certifiable Taboo Trap finetuning process

Inputs: $\alpha, \beta, \theta, f, x, y, \epsilon, E$
$\mathbf{m}^d$ = RandomMaskGen(β)
for $e = 0$ **to** $E - 1$ **do**
  L = CrossEntropy($y, f(x)$)
  $B = \emptyset, \hat{B} = \emptyset$
  for $l \in$ Layers(f) **do**
    $(B_l^{\text{up}}, B_l^{\text{low}})$ = BoundPropagate(l, x, f)
    $(\hat{B}_l^{\text{up}}, \hat{B}_l^{\text{low}})$ = BoundPropagate($l, x \pm \epsilon, f$)
    $B = B \cup (B_l^{\text{up}}, B_l^{\text{low}})$
    $\hat{B} = \hat{B} \cup (\hat{B}_l^{\text{up}}, \hat{B}_l^{\text{low}})$
  end for
  L_D, L_V = ComputeRegLoss($B, \hat{B}, f(x), \mathbf{m}^d$)
  α = Anneal(α, e)
  Opt$_\theta(L + \alpha(L_D + L_V))$
end for

3.5 Taboo Trap for Free

One major bottleneck in defending against adversarial samples is the training overhead. Classic methods like adversarial training increase model robustness by training with additional adversarial data points and thus significantly increase the training time. CTT can be deployed without any additional fine-tuning, and we name this detection mode CTT-lite.

We previously introduced the concept of a detection threshold value. Recall the definition of a particular layer's output activations $\mathbf{x}_l$, CTT uses a randomised binary mask $\mathbf{m}_l^d$ that is the same size of $\mathbf{x}_l$ to decided on which activation values to restrict on. Unlike [44] which used different transfer functions as keys, in this work we represent different keys as different subsets of neurons that are instrumented with CTT. We find that such a construction has all of the benefits described by [44]. Practically, CTT only detects on $\mathbf{x}_l \cdot \mathbf{m}_l^d$, where $\cdot$ is a Hadamard product (element-wise multiplication) between matrices.

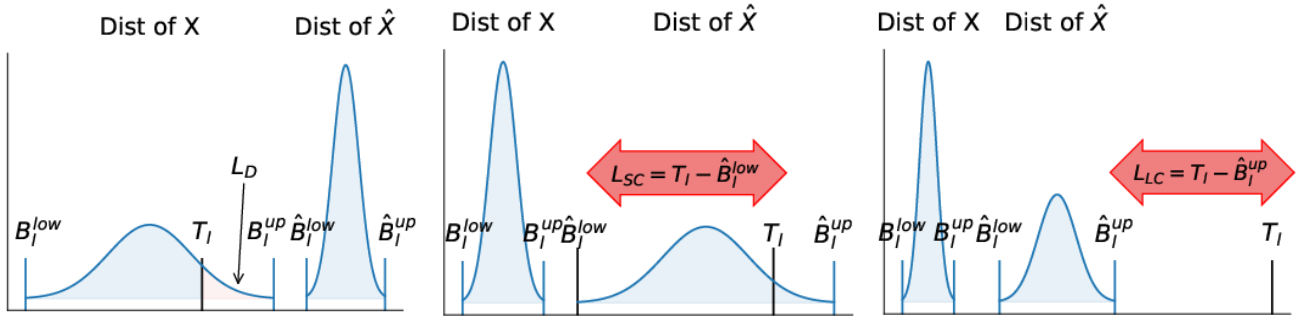
CTT-lite simply places T_l at the upper boundary of the natural set so that $T_l = B_l^{\text{up}}$. In the original Taboo Trap setup, as in Section 3.2, this effectively means $r = |f(x_{\max}) - T| = 0$. So the only additional computation is to perform the interval bound propagation for deducing the value of T_l in each layer, and no additional training is required. Note that as the bounds are computed for the training dataset, it will have false positives for the evaluation dataset.

We find that CTT-lite can detect very weak attackers such as FGSM with large epsilon, but struggles with attacks that produce small and mid-sized perturbations. As T_l placement is very far from optimal, CTT-lite should be considered a baseline detector. CTT-lite is pre-built into all networks by default and does not bring any additional costs – it is simply a natural upper bound of activations. We find that for LeNet5 with MNIST all adversarial samples with $l_2 > 10$ are detected by default.

3.6 Fine-tuning with CTT Losses

Fine-tuning networks further with CTT losses can introduce a better separation between the natural and adversarial sets. Unlike adversarial training, CTT fine-tuning operates on the original data; we do not generate any adversarial inputs to train with the model, so the training overheads are lower for CTT. We present three losses related to interval bounds that are considered as regularisations in our CTT detection. The three losses are presented in Figure 4, and they are: 1) Detection loss L_D, 2) Strict certification loss L_{SC}, 3) Loose certification loss L_{LC}.

Consider a masking function $\mathbf{m}_l = M(\mathbf{x}_l, T_l)$, the output $\mathbf{m}$ is a binary mask of which its elementwise entry is 1 if its corresponding elementwise entry in $\mathbf{x}$ is bigger than a scalar T_l, and otherwise is



(a) Detection loss L_D. **(b) Strict Certification loss L_{SC}.** **(c) Loose Certification loss L_{LC}.**

Figure 4: An illustration of CTT regularisation losses. The detection loss (L_D) ensures no natural samples are detected. Strict certification loss encourages the placement of T_l to be optimal, while loose certification loss helps T_l to achieve the suboptimal placement.

0. The detection loss L_D is a sum of all activation values picked by the taboo selection mask $\mathbf{m}_l^d$ that are greater than the detection threshold T_l. The verification losses are simply the distance between the detection threshold and the bound when the threshold is bigger than the bound. Considering a network with N layers, we have:

$$L_D = \sum_{l=0}^{N-1} \text{sum}(\mathbf{x}_l \cdot \mathbf{m}_l^d \cdot M(\mathbf{x}_l, T_l)) \tag{5}$$

$$L_{SC} = \sum_{l=0}^{N-1} \text{sum}(\mathbf{m}_l^d \cdot M(\hat{B}_l^{\text{low}}, T_l) \cdot (T_l - \hat{B}_l^{\text{low}})) \tag{6}$$

$$L_{LC} = \sum_{l=0}^{N-1} \text{sum}(\mathbf{m}_l^d \cdot M(\hat{B}_l^{\text{up}}, T_l) \cdot (T_l - \hat{B}_l^{\text{up}})) \tag{7}$$

The function sum produces the sum of all entries of a high dimensional tensor that is the result of convolutions (activations). Recall we previously defined the optimal and suboptimal placements of T_l in Section 3.3, the minimization of different combination of CTT regularisation losses provide:

- If $L_D = 0$ and $L_{SC} = 0$, we are achieving optimal placement of T_l. All adversarial inputs with l_∞ norm equal to ϵ are detectable, and all detected samples are adversarial samples regardless of the perturbation size. Given that, the test data fall into the training data distribution.
- If $L_D = 0$ and $L_{LC} = 0$, we are achieving suboptimal placement of T_l, all detected samples are adversarial samples regardless of the perturbation size. Given that, the test data fall into the training data distribution.

We present the detailed fine-tuning algorithm in Algorithm 1. The fine-tuning function takes a hyperparameter α, which controls how strong the regularisation is in the optimization procedure (Opt). In practice, it is necessary to anneal (Anneal) the value of α with respect to the number of epoch e. The other hyperparameter β is a probability between 0 to 1 that is later used to produce a set of masks $\mathbf{m}^d$ for each layer's activations. In the meantime, the fine-tune function considers a neural network f with trained parameters θ; x and y are the training data samples and their labels respectively. In addition, we need a pre-defined perturbation size ϵ for adversarial bound construction and E represents the maximum number of epochs for which we would like to fine-tune. The function CrossEntropy essentially computes the classification loss L based on the input training data.

Consider a neural network f parameterised by θ. For each layer in f, we perform the bound propagation as described in Section 3.3. The bounds for both the adversarial set of inputs and the natural set of inputs of each layer are accumulated for computing the regularisation loss using the function ComputeRegLoss. Note that the adversarial set represents the set of inputs with a particular l_∞ norm, so there is no actual generation of adversarial samples. The function ComputeRegLoss produces two losses L_D and L_C; the value of L_C can be calculated to be equal to whether L_{SC} or L_{LC} (Equation (6) and Equation (7)) depending on whether we use CTT-strict or CTT-loose. Since Algorithm 1 is only a high level overview, we did not distinguish between L_{SC} and L_{LC}, but call them in general L_C in Algorithm 1. It is worth to note that L_{SC} is a stronger regularisation than L_{LC}, so adding both regularisations is theoretically equivalent to adding only L_{SC}. The pre-defined parameter ϵ determines a trade-off between accuracy, detection ratios and adversarial accuracy. In practice, we determine the value of ϵ using a grid search spanning values from 10^{-5} to 10^{-1}, and determine its value based on the optimal performance in accuracy and detection ratio under a simple FGSM attack with fixed l_0. We explain this trade-off in detail in Section 4.4.

4 EVALUATION

4.1 Networks, Datasets and Attacks

We evaluate the proposed Certifiable Taboo Trap (CTT) on three different image datasets, MNIST [29], FashionMNIST [50] and CIFAR10 [23]. The MNIST dataset consists of images of hand-written digits and the number of output classes is 10. FashionMNIST is slightly harder than MNIST, and tries to classify pieces of clothing. The CIFAR10 dataset is a task of classifying 60000 images into 10 classes. We use the LeNet5 [28] architecture for MNIST, and evaluate an efficient CNN architecture (MCifarNet) from Mayo [52] that achieved a high classification accuracy using only 1.3M parameters.

Table 1: A comparison between CTT-lite, CTT-loose, CTT-strict, AdvTrain [27], Ensemble [38], MagNet reconstruction-based detector [33] and PCL [37] on the MNIST dataset. Acc means accuracy and Det means detection rate on adversarial samples.

Attack	Param	Baseline Acc	AdvTrain Acc	Ensemble Acc	PCL Acc	MagNet			CTT-lite			CTT-loose			CTT-strict		
						Det_{l_1}	Det_{l_2}	$Det_{l_1 \|\| l_2}$	Acc	Det	l_2	Acc	Det	l_2	Acc	Det	l_2
No Attack		99.1	99.5	99.5	99.3	1.75	1.93	2.93	99.1	1.9	-	98.5	1.6	-	98.9	1.1	-
FGSM	$\epsilon = 0.1$	66.7	73.0	96.3	96.5	54.49	54.59	54.80	70.9	1.4	2.08	25.0	100.0	1.98	61.1	100.0	1.99
	$\epsilon = 0.2$	25.7	52.7	52.8	77.9	85.20	85.31	85.31	21.9	1.0	4.14	15.0	100.0	3.89	32.7	100.0	3.90
BIM	$\epsilon = 0.1$	49.4	62.0	88.5	92.1	80.82	24.90	80.92	44.2	1.0	1.13	0.0	100.0	0.38	0.15	100.0	0.75
	$\epsilon = 0.15$	15.4	18.7	73.6	77.3	88.37	37.14	88.47	4.2	0.8	1.48	0.0	100.0	0.50	2.0	100.0	0.97
PGD	$\epsilon = 0.1$	59.4	62.7	82.8	93.9	83.78	77.96	83.78	51.0	1.2	1.50	1.0	100.0	1.24	13.4	100.0	1.35
	$\epsilon = 0.2$	1.83	31.9	41.0	80.2	98.27	98.27	98.27	0.0	1.1	2.73	0.0	100.0	2.43	0.9	100.0	2.53

Table 2: A comparison between CTT-loose, CTT-strict, AdvTrain [27], Ensemble [38], MagNet reconstruction-based detector [33] and PCL [37] on the Cifar10 dataset. Acc means accuracy and Det means detection rate on adversarial samples.

Attack	Param	Baseline Acc	AdvTrain Acc	Ensemble Acc	PCL Acc	MagNet			CTT-loose			CTT-strict		
						Det_{l_1}	Det_{l_2}	$Det_{l_1 \|\| l_2}$	Acc	Det	l_2	Acc	Det	l_2
No Attack		89.1	84.5	90.6	91.9	6.40	6.61	8.13	86.2	3.4	-	86.3	6.4	-	86.1	3.0	-
FGSM	$\epsilon = 0.02$	33.6	44.3	61.7	78.5	7.80	6.64	9.55	18.6	95.7	1.07	16.8	98.5	1.08	16.1	96.4	1.06
	$\epsilon = 0.04$	22.4	31.0	46.2	69.9	11.53	8.38	13.27	7.6	93.6	2.00	7.2	94.2	2.01	6.0	93.1	2.06
BIM	$\epsilon = 0.01$	13.5	22.6	46.6	74.5	6.98	6.52	8.61	0.5	9.0	0.15	0.0	14.1	0.16	1.1	10.9	0.16
	$\epsilon = 0.02$	1.5	7.8	31.0	57.3	6.64	6.52	8.50	0.0	14.2	0.21	0.0	25.9	0.20	0.0	17.2	0.21
PGD	$\epsilon = 0.01$	24.0	24.3	48.4	75.7	7.10	6.52	8.73	0.1	10.4	0.34	2.9	24.3	0.34	2.0	16.6	0.34
	$\epsilon = 0.02$	2.9	7.8	30.4	48.5	6.98	6.52	8.85	0.0	40.8	0.65	0.0	70.3	0.65	0.0	49.9	0.65

Table 3: A comparison between CTT-lite, CTT-loose, CTT-strict, Madry et al. [32], Sitatapatra [44], ABS and Binary ABS [41] on the MNIST dataset. For detection based defense, we show results in the form of $a(d)$, where a is accuracy and d is detection rate. GE represents gradient estimation.

	CNN	Madry et al.	Binary ABS	ABS	Sitatapatra	CTT-loose	CTT-strict
No Attack	99.1%	98.8%	99.0%	99.0%	99.2% (2%)	99.1% (0.5%)	98.8% (1.3%)
l_2-metric ($\epsilon = 1.5$)							
FGM	48%	96%	-	-	2% (3%)	4% (99%)	21%(100%)
FGM w/ GE	42%	88%	68%	89%	4% (7%)	0% (100%)	25%(100%)
Deepfool	18%	91%	-	-	12% (1%)	0% (100%)	77% (95.6%)
Deepfool w/ GE	30%	90%	41%	83%	6% (2%)	0% (100%)	76.5% (94.4%)
L2 BIM	13%	88%	-	-	0% (0%)	0% (100%)	0% (100%)
L2 BIM w/ GE	37%	88%	63%	87%	0% (3%)	0% (100%)	0% (100%)
l_∞-metric ($\epsilon = 0.3$)							
FGSM	4%	93%	-	-	2%(3%)	1% (99%)	2%(100%)
FGSM w/ GE	21%	89%	85%	34%	0%(2%)	0% (100%)	4% (100%)
BIM	0%	90%	-	-	0%(1%)	0% (100%)	0% (100%)
BIM w/ GE	37%	89%	86%	13%	0%(1%)	0% (100%)	0% (100%)

We consider gradient-based FGSM [15], FGM [15], BIM [27], PGD [27] and C&W [6] attacks with various attack parameters. These attacks can be seen as a collection of l_∞ and l_2 based attacks. In addition, we provide results in both White-box and Black-box settings. For Black-box attacks, we use gradient estimation with the coordinate-wise finite-difference method, similar to Schott et al. [41]; and a fully decision-based Boundary Attack [3]. The attack implementations are from Foolbox [39].

4.2 Parameter Choices

Most of the commonly-used optimisers are suitable for CTT training. However, there exists an interaction between the CTT penalty (the additional loss term introduced by CTT) and the weight decay of the optimizer. Although we have not evaluated this interaction formally, we find it easier to train models when the weight decay is either turned off or set to a very small value. The optimizer used in our experiments is RMSProp.

The annealing procedure (Anneal in Algorithm 1) for CTT parameters is important for convergence. The parameter α determines the strength of the CTT penalty, and we increase α iteratively by a factor of β every t training epochs. For both MNIST and FashionMNIST, we used $t = 6, \beta = 0.005$. For CIFAR10, we used $t = 30, \beta = 0.001$. In all of the networks we instrumented a proportion of the second layer with CTT. We find that the best way to train the models is to first optimise L_V, i.e. make sure that neurons have a bound larger than T_l and then start iteratively increasing α. We hypothesise that this works in line with recent findings that there exist a number of connected convergence clusters with similar performance [12] with a path between them. Iteratively increasing α allows us to keep convergence, while maintaining low L_V loss and decreasing the false-positive rate.

4.3 Attackers with Various Capabilities and Various Norm Bounds

Attacks can be evaluated very differently, and we offer two sets of evaluations for a thorough comparison with existing defense methods. In the first set, we run attacks with fixed parameters and a fixed number of iterations. In the second set, we enable early stopping for iterative attacks so that perturbation sizes are fixed. In addition, we also provide evaluation with Black-box attacks using gradient estimation. We used $\epsilon = 3 \times 10^{-3}$ for MNIST networks, and $\epsilon = 10^{-4}$ for CIFAR10 networks. These values were determined from a grid search, a detailed discussion of the grid search and an evaluation of using different ϵ is shown in Section 4.4. We show the detailed hyperparameter configurations of α, β, E and ϵ in Algorithm 1.

In Table 1 and Table 2, we present comparisons between CTT and various robust adversarial training schemes, including Adv-Train [27], Ensemble [38], MagNet [33] and PCL [37]. In this setup, we run attacks with fixed parameters and measure the accuracy, detection ratios and l_2 norms of the adversarial samples. BIM and PGD iterated for 10 times with a step size of $\epsilon/10$; Notice we present the baseline accuracy for the networks on which we evaluate. The baseline accuracy will be the same as CTT-lite, since it involves no re-training of the model. CTT-lite provides limited protections against adversarial attacks. CTT-loose and CTT-strict, however, show above 90% detection ratios across all examined attacks in Table 1. In addition, both detection schemes provide a degree of certifiability on the detected adversarial samples. The detection ratios when no attacks are applied are the false positives. There exists a trade-off between the false-positive rates and the detection ratios. As presented in Table 2, the two versions of CTT-loose have different false-positive rates, and offer different detection capabilities. Table 2 shows our detection scheme outperform robust networks on FGSM, however, provides relatively worse performance when

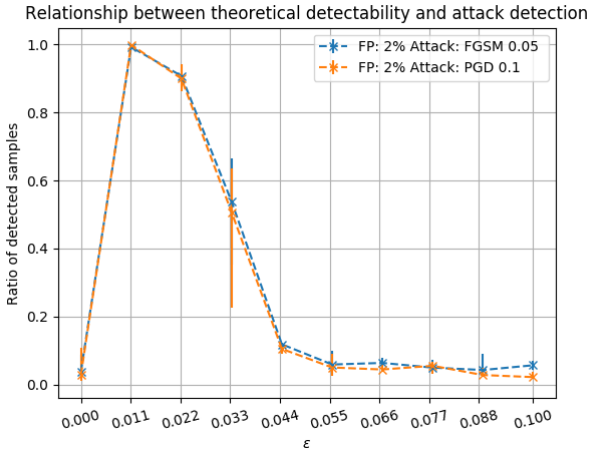


Figure 5: Trade-off between choices of ϵ and detector performance. There are five LeNet5 networks classifying MNIST, instrumented with CTT-loose with $T_l = 10^{-4}$ with a given ϵ. The networks are trained to a false-positive rate of 2%. Points show median performance, whereas error bars refer to standard deviations of the 5 networks.

l_2 norms are low. First, our detection offers certifiability which is not seen in any of the work compared. Second, the work compared does not report the l_2 norm, attacks with different random starts may cause a difference in l_2 norms and also the attacking quality.

To further evaluate the CTT system, we conduct a comparison to Madry et al., Sitatapatra [44], ABS and Binary ABS [41] under both White-box and Black-Box attacks on the MNIST dataset. We set a noise budget for each attack, and the Black-Box attacks are constructed using gradient estimation. We see almost all CTT-loose and CTT-strict results show above 90% detection of adversarial samples while keeping the false positives low. These results outperform all other competitors that focus solely on improving model robustness. As can be seen in Table 1, the robustness-based defenses have higher accuracy than to CTT on adversarial images. Intuitively, CTT enforces separation of natural and adversarial sets by the detection thresholds. The CTT models are thus more sensitive to adversarial samples – we observe selected neurons get suppressed for natural inputs but get non-zero values for adversarial ones. CTT-strict will show 100% detection on all attacks that are above a certain given l_∞ – which is exactly what we see in Table 1 with l_∞-based attacks. For l_2-based attacks, it is hard to ensure every pixel is under the given certifiable limit, but our method practically capture many adversaries with high detection rates.

4.4 False Positives Trade-off

In this section we show the impact of different false-positive rates on the CTT-loose instrumentation. We use 5 LeNet5 networks and train each of them with the same CTT-loose restrictions but stop at various training time so that networks achieve different false-positive rates. Figure 6 presents the false positive rate trade-off

Figure 6: Trade-off between choices of false positive rates and detector performance. There are five LeNet5 networks classifying FashionMNIST, instrumented with CTT-loose with $T_l = 10^{-4}$ with a given $\epsilon = 0.005$. Points show median performance, whereas error bars refer to standard deviations.

two specific attacks, with false positive rates on the x-axis and detectability on the y-axis.

The relationship between detector performance and false positive rates indicates a trade-off of interest when applying CTT-loose in practice. With a slight increase of false positive rates (1% to 3%), we increase the detector performance by around 20%. Intuitively, this suggests first, that there exist inefficiencies in the internal representations of the neural network, where the network struggles to separate natural and non-natural samples (similarly to [14]); and second, this trade-off between false positive rates and detectability is the result of of imperfections of the training dataset. If the training dataset involves imperfect, confusing images, this leaves a vague boundary between the natural and adversarial input sets. Although this relationship exists across different datasets and models, its scaling seems to be dataset-dependent.

4.5 FashionMNIST on LeNet5

MNIST is a popular benchmark, but is known to be relatively simple [29]. Xiao et al. proposed FashionMNIST [50], a more complex, yet still simple toy dataset. In this section we report on results of CTT-loose instrumentation of LeNet5 networks solving FashionMNIST with $\epsilon = 0.001$, meaning that the adversarial set includes perturbed images with an l_∞ size of 0.001.

In addition to the attacks presented in the previous section, we also show here the results for a decision-based attack [3]. The attack itself is particularly interesting as it is not based on any gradient information, so CTT detection is not network-information specific. For this attack, we use 25 trials per iteration and vary the number of iterations.

Table 4 shows the results of attacking LeNet5 instrumented with CTT-loose. In the evaluation section of the paper, we have shown that for MNIST, CTT detection was capable of capturing almost all of the adversarial samples. Unlike MNIST, CTT fails to detect all of the adversarial samples on FashionMNIST.

As already noted, there is a relationship between the attack perturbation size, dataset specifics, and the detectability of CTT. In the case of FashionMNIST, for the particular ϵ value, we find it shows relatively better detection rate for small l_2 values.

4.6 Runtime Overheads and Security Protocols

The proposed CTT system has low runtime overheads compared with other detection systems (SafetyNet [31] and MagNet [33]). It is similar to Sitatapatra [44], another derivative of Taboo Trap; CTT supports the concept of embedding keys in each neural network to diversify models under adversarial attack. The key is embedded via the mask and can support complex security protocols; a detailed analysis of key attribution and runtime overheads can be found in Shumailov et al. and these advantages are equally applicable to CTT.

Table 4: CTT-loose instrumented LeNet5 network classifying FashionMNIST. We show results in the form of $a(d)$, where a is accuracy and d is detection rate.

	θ	l_2	CTT-loose	
No Attack			90.6% (5.3%)	90.7% (3.2%)
FGSM	$\epsilon = 0.006$	0.14	86.55% (92.44%)	84.55% (97.06%)
	$\epsilon = 0.007$	0.17	84.63% (92.65%)	82.27% (98.08%)
	$\epsilon = 0.01$	0.24	78.08% (92.78%)	75.57% (95.81%)
	$\epsilon = 0.03$	0.71	42.49% (82.12%)	41.14% (80.69%)
	$\epsilon = 0.05$	1.17	22.82% (71.30%)	25.80% (69.53%)
	$\epsilon = 0.07$	1.63	16.38% (64.32%)	17.61% (63.59%)
Boundary	$i = 10$	2.92	0.00% (34.58%)	0.00% (42.39%)
	$i = 50$	2.35	0.00% (32.99%)	0.00% (36.93%)
	$i = 100$	1.88	0.00% (35.71%)	0.11% (39.59%)
	$i = 500$	0.51	0.00% (77.63%)	0.00% (81.48%)
	$i = 1000$	0.35	0.11% (84.84%)	0.00% (87.39%)
C&W	$c = 0.1$ $b = 1$	4.85	25.73% (15.20%)	24.66% (8.79%)
	$c = 0.5$ $b = 1$	0.10	65.79% (78.21%)	61.42% (79.89%)
	$c = 0.1$ $b = 5$	0.19	0.00% (79.68%)	0.00% (81.05%)
	$c = 0.5$ $b = 5$	0.23	0.88% (74.78%)	0.22% (77.11%)
	$c = 0.1$ $b = 10$	0.19	0.00% (80.81%)	0.00% (81.05%)
	$c = 0.5$ $b = 10$	0.22	0.88% (76.33%)	0.22% (79.78%)

5 DISCUSSION

5.1 Robustness and Detection

What does it mean for a neural network to be *robust*? Depending on who is asked the answers will range from interpretability i.e. understanding what influenced the decision, through to detection i.e. flagging up inputs that confuse the network, to resilience i.e. tolerance of perturbations of some particular size. Although each interpretation is useful, they all answer conceptually different questions and have large implications. We believe all three cases should be considered together.

When a previous version of this paper was submitted to a top machine learning conference, a number of reviewers questioned the usefulness of detection, arguing that for a car moving at speed, not making a decision while under attack might be as dangerous as

making a wrong decision. We disagree. Humans and many other animals have evolved an acute awareness of hostile intent, and for good reasons. Remaining alert for extended periods of time is exhausting. In systems security, situational awareness is critical in many real-life contexts; companies spend real money on threat intelligence, and monitor DNS to detect whether any machine in their network has been compromised. Academic security thinking has been influenced by cryptography, where one assumes a Dolev-Yao opponent (i.e. the enemy controls the phone company) and can use mechanisms that have security proofs to assure confidentiality and integrity of our communications. In most real-life applications, however, the costs of making a system resilient to all attacks are excessive. (Even in the cryptographers' model, there is no guarantee of availability: a hostile phone company can always deny us service.)

In the specific context of road vehicles, manufacturers must qualify all programmable electronics under ISO 26262, which involves careful hazard analysis leading to requirements for both safety functionality and safety integrity. These standards are about to be complemented by the draft ISO SAE/DIS 21434 on cybersecurity for road vehicles which extends hazard analysis to threat analysis. As a result, automotive machine-vision systems are significantly detuned to make adversarial attacks extremely difficult. Attack detection is also a valid means of response and may enable the use of better vision systems; in the event of an attack being detected, the vehicle can simply switch to its default safety behaviour of coming to a stop.

Next, it is often correct to be uncertain about a decision. When classifying cats and dogs, a giraffe should not be called either. Modern safety-critical systems typically have a number of fallback modes. In the case of road vehicles, the options include fallback to a limp-home mode with limited speed, and reversion to manual operation. The kind of DNN resilience on offer from adversarial training or certifiable robustness does not react to the attacks so much as remaining ignorant to their existence. This may seem ideal in isolation but is nowhere near ideal in many real-world applications.

If there is an actual attack, in which a malicious actor projects images on a wall with the intent of killing people in a car driving past it, it makes little difference whether the attacker has to use 8/255 or 35/255 perturbation, as long as the images work. If a car manufacturer or Tier-1 component supplier wishes to sell a DNN that is resilient to an ϵ-attacker, but only for easily achievable values of ϵ, we fail to see how such a product could usefully fit into the automotive safety ecosystem.

Resilience mechanisms may have some applications, but detection mechanisms probably have more. They must considered in the context of the design of larger systems, many of which already have other mechanisms for intrusion detection and situational awareness. Indeed two of the fastest-growing sectors in the cybersecurity ecosystem are *security orchestration and response* (SOAR) and *security incident and event management* (SIEM). On top of that, safety-critical systems of many kinds have their own mechanisms for resilience at the system level, involving redundancy, fallback and response.

5.2 Desirable Properties of the Detector

Detector mechanisms can be designed with different properties in mind. In this section we discuss properties that we consider important for real world deployable detectors.

5.2.1 Computational complexity. – when designing detectors it is important to consider them in context of real systems, where energy, latency, and memory constraints must be taken into consideration. For example, for simple MNIST, MagNet introduces an additional 20% and SafetyNet 2000–3000% computation overhead [46]. Such overheads can be used to create service-denial attacks [45]. Ideally, detection mechanisms should introduce zero run-time overheads in both computation and memory access. Detectors with this property include Taboo Trap [44, 46], CTT, introduction of an additional class [17], ODD [25], and Injected Attractors [51].

5.2.2 Hardware awareness. – a large number of neural networks now need to run on particularly constrained devices, such as security cameras or unmanned vehicles. This introduces additional complexity to the detection mechanisms – they need to be effective with highly quantised data representations, and to adapt to varying architectures produced by AutoML algorithms. This in turn suggests a need for detection mechanisms that are flexible and adaptive. CTT supports running on individual layers and even particular filter banks, making it flexible in deployment. Furthermore, detection can be very energy-efficient in hardware, since optimised detection is just numerical comparison of particular activation values. In addition, CTT can be seamlessly integrated into low-precision CNN inference hardware by just looking at overflow flags at the end of arithmetic operations.

5.2.3 Diversity and key size. – diversity is really important for detectors that may be attacked. In 1883, the cryptographer Auguste Kerckhoffs outlined a design principle that has stood the test of time: a system should withstand enemy capture, and it should remain secure if everything about it, except the value of a key, becomes public knowledge [22]. This suggests a need for controllable diversity of detectors, such that breaking one detector would not have an effect on another.

CTT randomises the set of detector neurons and uses them as a key which gives a significant advantage against Black-box attacks. Kupek et al. showed that it is possible to find detector thresholds of Taboo Trap in a White-box setting [26]. Exact model extraction techniques such as the one presented by Jagielski et al. may be the best known attack in a Black-box setting [19].

6 CONCLUSION

In this paper, we presented the Certifiable Taboo Trap (CTT), a new way for neural networks to detect adversarial samples. We discussed three different modes which provide different detection capabilities and levels of certifiability at different training costs. All variants have a small run-time overhead, and can be customised with the equivalent of cryptographic keys. The stronger variants have extra training but this is used to characterise propagation bounds rather than to defend against specific adversarial samples, yielding a more flexible and general defense mechanism.

7 ACKNOWLEDGMENTS

Partially supported with funds from Bosch Forschungsstiftung im Stifterverband.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561* (2015).

[2] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2018. Black-box attacks on deep neural networks via gradient estimation. *International Conference on Learning Representations Workshop (ICLR)* (2018).

[3] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SyZI0GWCZ

[4] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden Voice Commands. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association.

[5] Nicholas Carlini and David Wagner. 2017. Magnet and" efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478* (2017).

[6] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 39–57.

[7] Steven Chen, Nicholas Carlini, and David A. Wagner. 2019. Stateful Detection of Black-Box Adversarial Attacks. *CoRR* abs/1907.05587 (2019). arXiv:1907.05587 http://arxiv.org/abs/1907.05587

[8] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified Adversarial Robustness via Randomized Smoothing. In *Proceedings of the 36th International Conference on Machine Learning*.

[9] Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. 2018. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265* (2018).

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1625–1634.

[11] Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh. 2003. Road-sign detection and tracking. *IEEE transactions on vehicular technology* 52, 5 (2003), 1329–1341.

[12] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. 2019. Deep Ensembles: A Loss Landscape Perspective. *arXiv preprint arXiv:1912.02757* (2019).

[13] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. 2018. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.

[14] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. 2018. Adversarial spheres. *arXiv preprint arXiv:1801.02774* (2018).

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)* (2015).

[16] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715* (2018).

[17] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick D. McDaniel. 2017. On the (Statistical) Detection of Adversarial Examples. *CoRR* abs/1702.06280 (2017). arXiv:1702.06280

[18] Jörn-Henrik Jacobsen, Jens Behrmann, Nicholas Carlini, Florian Tramèr, and Nicolas Papernot. 2019. Exploiting Excessive Invariance caused by Norm-Bounded Adversarial Robustness. *CoRR* abs/1903.10484 (2019). arXiv:1903.10484 http://arxiv.org/abs/1903.10484

[19] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High Accuracy and High Fidelity Extraction of Neural Networks. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/usenixsecurity20/presentation/jagielski

[20] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2013), 221–231.

[21] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*. Springer, 97–117.

[22] Auguste Kerckhoffs. 1883. La cryptographie militaire. *Journal des sciences militaires, vol. IX*, 161–191.

[23] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2014. The CIFAR-10 dataset. (2014).

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[25] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Remi Le Priol, and Aaron Courville. 2020. Out-of-Distribution Generalization via Risk Extrapolation (REx). *arXiv preprint arXiv:2003.00688* (2020).

[26] Tobias Kupek, Cecilia Pasquini, and Rainer Böhme. 2020. On the Difficulty of Hiding Keys in Neural Networks. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security* (Denver, CO, USA) *(IH&MMSec '20)*. Association for Computing Machinery, New York, NY, USA, 73–78. https://doi.org/10.1145/3369412.3395076

[27] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial machine learning at scale. (2017).

[28] Yann LeCun et al. 2015. LeNet-5, convolutional neural networks. (2015), 20.

[29] Yann LeCun, Corinna Cortes, and CJ Burges. 2010. MNIST handwritten digit database. 2 (2010).

[30] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2018. Certified Robustness to Adversarial Examples with Differential Privacy. In *IEEE S&P 2019*.

[31] Jiajun Lu, Theerasit Issaranon, and David A Forsyth. [n.d.]. SafetyNet: Detecting and Rejecting Adversarial Examples Robustly.

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. (2018).

[33] Dongyu Meng and Hao Chen. 2017. MagNet: A Two-Pronged Defense Against Adversarial Examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. ACM, New York, NY, USA, 135–147.

[34] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On Detecting Adversarial Perturbations. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*.

[35] Matthew Mirman, Timon Gehr, and Martin Vechev. 2018. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*.

[36] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: a simple and accurate method to fool deep neural networks. (2016).

[37] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. 2019. Adversarial Defense by Restricting the Hidden Space of Deep Neural Networks. In *The IEEE International Conference on Computer Vision (ICCV)*.

[38] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. 2019. Improving Adversarial Robustness via Promoting Ensemble Diversity. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). 4970–4979.

[39] Jonas Rauber, Wieland Brendel, and Matthias Bethge. 2017. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131* (2017).

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[41] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. 2019. Towards the first adversarially robust neural network model on MNIST. *International Conference on Learning Representations Workshop (ICLR)* (2019).

[42] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[43] Shawn Shan, Emily Willson, Bolun Wang, Bo Li, Haitao Zheng, and Ben Y. Zhao. 2019. Gotta Catch 'Em All: Using Concealed Trapdoors to Detect Adversarial Attacks on Neural Networks. *CoRR* abs/1904.08554 (2019). arXiv:1904.08554 http://arxiv.org/abs/1904.08554

[44] Ilia Shumailov, Xitong Gao, Yiren Zhao, Robert Mullins, Ross Anderson, and Cheng-Zhong Xu. 2019. Sitatapatra: Blocking the Transfer of Adversarial Samples. (2019).

[45] Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. 2020. Sponge Examples: Energy-Latency Attacks on Neural Networks. *arXiv preprint arXiv:2006.03463* (2020).

[46] Ilia Shumailov, Yiren Zhao, Robert Mullins, and Ross Anderson. 2018. The Taboo Trap: Behavioural Detection of Adversarial Samples. *arXiv preprint arXiv:1811.07375* (2018).

[47] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR* abs/1312.6199 (2013). arXiv:1312.6199

[48] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. 2018. Efficient formal safety analysis of neural networks. In *Advances in Neural Information Processing Systems*. 6367–6377.

[49] Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. 2018. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*. 8400–8409.

[50] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv:cs.LG/1708.07747 [cs.LG]

[51] Jiyi Zhang, Ee-Chien Chang, and Hwee Kuan Lee. 2020. Detection and Recovery of Adversarial Attacks with Injected Attractors. *arXiv preprint arXiv:2003.02732* (2020).

[52] Yiren Zhao, Xitong Gao, Robert Mullins, and Chengzhong Xu. 2018. Mayo: A Framework for Auto-generating Hardware Friendly Deep Neural Networks. (2018).

[53] Yiren Zhao, Ilia Shumailov, Robert Mullins, and Ross Anderson. 2018. To compress or not to compress: Understanding the Interactions between Adversarial Attacks and Neural Network Compression. *arXiv preprint arXiv:1810.00208* (2018).