– Independent Work Report Spring 2019 –

Robust OOD Detection in Secure Open-World Learning

Matteo Russo
Adviser: Professor Prateek Mittal
PhD Supervisors: Arjun Bhagoji, Vikash Sehwag

Abstract

In recent years, the paradigm of Deep Learning has revolutionised prediction techniques in extremely diverse fields of knowledge, ranging from autonomous driving to medical diagnosis. Oftentimes, Deep Learning models are tested against a pre-defined distribution of samples with a fixed set of labels. This does not account for the fact that, in real world settings, samples are collected from an open-world environment, where input data is partially if not fully out-of-distribution (OOD). Not considering this as a possibility is, in fact, one of the major vulnerabilities of Deep Neural Networks, especially if exploited by adversaries in evasion attacks, as shown in recent work. This poses a real threat to the deep learning applications, whether because of malicious attacks or because of the nature of the data. In this paper, we propose two methods for robust OOD detection: the first based on Gradients Magnitude Analysis of test data and the second based on the use of Hessian Spectral Norm Analysis to determine how the curvature of the loss surface differs for in-distribution and out-of-distribution samples. In addition, we observe that benign OOD samples are detected with high degree of accuracy and confidence. On the other hand, for adversarially generated OOD samples, the detection rate is not as high. Finally, we test these models against a variety of datasets, demonstrating the importance of such detection methods for Secure Open-World Learning.

1. Introduction

1.1. Motivation and Goal

Statistical learning is the branch of inferential and probabilistic statistics that solves the general problem of estimating the best predictive function only through the knowledge of the data. Let us define $\mathcal{D} := \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{n}$ to be the finite set of ordered sample-class pairs we can infer a predictive function from. Hereby, let us name sample space $\mathcal{X}$ and class space $\mathcal{Y}$ the spaces such that $(\boldsymbol{x_i}, y_i) \sim \Omega(\mathcal{X} \times \mathcal{Y}),\ \forall\ i \in \{1, ..., n\}$, where the symbol "$\sim$" denotes the sampling from probability distribution Ω over the space obtained by the Cartesian Product of the sample and class spaces respectively. The problem of statistical learning is that of finding a function $f_* : \mathcal{X} \to \mathcal{Y}$ such that the following holds:

$$f_* \in arg \min_{f \in \mathcal{X} \times \mathcal{Y}} \quad \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x_i})) \tag{1}$$

where $\mathcal{L}$ is the loss function, i.e. an error function measuring the difference between the true class value y_i and the predicted class value $f(\boldsymbol{x_i})$. Considering the confidence function $g : \mathcal{X} \to \mathbb{P}(\mathcal{Y})$ (where $\mathbb{P}(\mathcal{Y})$ represents the set of probability distributions over $\mathcal{Y}$ [2]) (1) can be rewritten as follows:

$$f_*(\boldsymbol{x}) \in arg \max_{y \in \mathcal{Y}} \quad g(\boldsymbol{x})(y) \tag{2}$$

Given the finiteness of the data at hand and the desire to prevent overfitting of the predictive function to the possibly noisy samples $\boldsymbol{x_i}$, we split $\mathcal{D}$ into a training set $\mathcal{D}_{train}$ over which we estimate the function f_* and a testing set $\mathcal{D}_{test}$ where we evaluate the performance of the estimated function on virtually unseen samples. In the context of Closed-World Learning (the realm most modern Machine Learning research and work has leaned towards), it is always assumed that samples in the training and testing sets come from the same distribution. However, in a much more realistic scenario, testing samples are gathered in an

open environment and thus, it is much more likely that the testing set would be composed of samples that belong to the same distribution of the training samples (in-distribution or IN) and samples that belong to a different distribution than the one we have trained our model f_* on (out-of-distribution or OOD). This induces the following notation: $\boldsymbol{x_{IN}}$ will denote a proxy for any in-distribution testing sample and $\boldsymbol{x_{OOD}}$ a proxy for out-of-distribution ones. In particular, in the context of classification, where the set of classes is prefixed and finite ($|\mathcal{Y}| = C < \infty$), the model f_* is forced to output one of the C classes for both $\boldsymbol{x_{IN}}$ and $\boldsymbol{x_{OOD}}$ which is completely nonsensical for the latter given that there exists no class $y \in \mathcal{Y}$ such that $f_*(\boldsymbol{x_{OOD}}) = y$. However, we do not know *a priori* which samples are IN or which samples are OOD. **The goal of this project is to design a detector h that is able to distinguish IN from OOD samples as illustrated in Figure 1.**

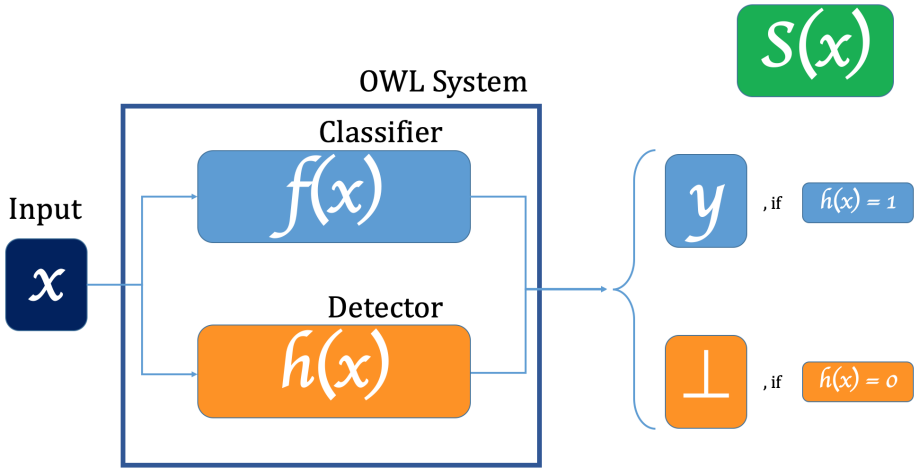


Figure 1: The Open-World Learning system $S(\boldsymbol{x})$ composed of a classifier or predictive function $f_*(\boldsymbol{x})$ and a detector $h(\boldsymbol{x})$ which equals 0 when $\boldsymbol{x}$ is an OOD input and 1 when $\boldsymbol{x}$ is an IN input [2].

We have not yet considered the possibility to be dealing with not only OOD but also adversarial points. An adversarial point $\tilde{\boldsymbol{x}}$ is defined as follows:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta} \quad \textbf{s.t.} \quad f_*(\tilde{\boldsymbol{x}}) = \omega \quad \textbf{with} \quad d(\tilde{\boldsymbol{x}}, \boldsymbol{x}) < \varepsilon \quad \textbf{for} \quad \boldsymbol{\delta} \succ \mathbf{0},\ \varepsilon > 0 \tag{3}$$

Hereby, $\boldsymbol{\delta}$ is the adversarial perturbation added on $\boldsymbol{x}$, ω is the target class we would like $\tilde{\boldsymbol{x}}$ to be mislabeled in and d is a distance metric. The combinatorial nature of the above posed

problem may lead to computational inefficiencies. Madry et al. in [21] achieve a state-of-the art performance with the approximate adversarial perturbation derived from a technique named PGD (Projected Gradient Descent). It consists of the following procedure for each iteration $t \in \{1, ..., T\}$:

$$\tilde{\boldsymbol{x}}^{\boldsymbol{t}} = \Pi_{\mathcal{B}}(\tilde{\boldsymbol{x}}^{\boldsymbol{t-1}} - \alpha \cdot sign(\nabla_{\tilde{\boldsymbol{x}}^{t-1}} \tilde{\mathcal{L}}(\omega, f_*(\tilde{\boldsymbol{x}}^{\boldsymbol{t-1}})))) \tag{4}$$

Hereby, Π denotes projection operator constrained by the set $\mathcal{B} = \{\boldsymbol{\delta} : ||\boldsymbol{\delta}||_\infty \leq B\}$ and $\tilde{\mathcal{L}}$ is the adversarial loss function based on the confidence function g through f_*. In words, PGD takes a benign OOD sample $\boldsymbol{x_{OOD}}$, it repeatedly projects it, and converges to an adversarial OOD sample $\tilde{\boldsymbol{x}}_{\boldsymbol{OOD}}$ after a certain number of iterations.

1.2. Overview

The problem of Out-Of-Distribution sample detection is a newly considered problem that differentiates itself substantially from anomaly detection, insofar as one crucial statistical assumption in the anomaly detection problem is that the outliers are drawn from the same distribution, but, likely, due to large fourth moments (kurtosis), those look very much different from the canonical samples that are not outliers. As the acronym suggests, OOD detection is something that pushes to the extreme this very concept: we need to detect whether a particular unlabelled sample belongs or not to the distribution of data points we have at hand. The task is evidently harder than the anomaly detection one insomuch as we have no perfect knowledge of the distribution where the test samples may come from; indeed, it could be any distribution. Unlike the anomaly problem, the field of OOD detection is relatively new and, hence, there are several opportunities for novel research results. Hitherto, as summarised by the papers, the main approaches to OOD detection follow one of the following two line of thought: the first assumes perfect knowledge of the Out-Of-Distribution data manifold and trains the model on part of the OOD data, making the model more robust for inference time. This approach resembles closely anomaly detection and is, therefore, unrealistic for

real-world scenarios where we cannot anticipate what the OOD distribution or distributions will be. This motivates the second approach where the classifier model assumes a partial notion of labelled OOD data in a Semi-Supervised Learning fashion. This approach has provable guarantees for performances as per [20], if we assume notion of a fraction α of the data. Our approach is different from both the previous two. It is solely based on the estimated empirical distribution of the gradients, on one hand, and the Hessians norms, on the other, of the loss function with respect to the weights at the end of the training phase on the in-distribution and out-of-distribution data.

1.3. Summary of Approach, Implementation and Results

In this research paper, we would like to propose two increasingly performing methods for robust OOD detection: the first based on *Gradients Magnitude Analysis* or GMA and the second based on *Hessian Spectral Norm Analysis* or HNA of test data. In particular, for both gradient norms and Hessian spectral norms, we consider in-distribution samples and compute the norm of the gradient (or Hessian) evaluated at those points after the Deep Neural Network training phase and we do the same with OOD points. Subsequently, we study the properties of the Gradient Norms distributions for IN and OOD samples and we do the same for Hessian Spectral Norms. We find that it is easier to distinguish between the two distributions when those are extracted from Hessian Spectral Norms rather than gradients. In addition, we have analytically shown that Hessian spectral Norms upper bound gradient ones and thus allow us to have more degrees of freedom to distinguish between IN and OOD samples in worst-case scenarios. In order to thoroughly test the designed detectors, we have generated adversarial OOD samples at inference time through the PGD attack. We have, thus, tested our detectors against benign OOD data and adversarially perturbed OOD test samples, across a variety of different datasets, with naive and robustly trained datasets. The rationale behind the choice of the HNA method comes from the fact that, after training, the Gradient Descent algorithm(canonical method utilized to find the empirical loss minimizing

function) will lead to an approximate local minimum over the loss function surface when evaluated at IN points. This implies that the ε-neighbourhood around that approximate minimum will result to be considerably flat, and the gradient norms evaluated at IN points will be very small. Moreover, we have empirically found that as much as the IN samples, the OOD ones result in very low gradient norms as well.

2. Problem Background and Related Work

Despite being a relatively recent problem to be addressed, in the past few years, OOD detection has become a focus of attention for the Machine Learning and Pattern Recognition community.

ODIN: Liang et al. in [19] design a system called ODIN to detect OOD samples. ODIN slightly perturbs the input in order for the network to result more robust to OOD injections at inference time. In particular, ODIN determines, in a semi-supervised learning fashion, a threshold on the maximum output confidence g to detect OOD samples.

Deep Generative Models: Schlegl et al. and Chalapathy et al., respectively in [23] and [4], use Deep Generative Models to detect Out-Of-Distribution samples: the first uses a Generative Adversarial Network (GAN) to learn the "distance" between the distribution of the IN data and the OOD data. The second uses an Autoencoder for the same exact purposes. Furthermore, in [22], Ruff et al. design a method called Deep-SVDD which learns, through an autoencoder, a hypersphere that is capable to function as a discrimen between IN and OOD samples.

Mahalanobis Detector: Lee at al, in [18], create the so-called Mahalanobis detector, which generates class-conditional Gaussians from convolutional features. For a new input, the Mahalanobis distance between the features for that input and the closest class-conditional distribution is calculated in order to detect OOD.

Network Agnostophobia: In [7], Dhamija et al. use an additional loss term which is aimed at maximizing the entropy in the softmax output of OOD samples, so to obtain a distinguishing feature for the latter set of samples.

Several other publications such as Erfani et al.'s in [8] or Jiang et al.'s in [13] design seemingly robust OOD detection systems. However, in both [24] and [2], the authors showed that all these detection techniques ([4], [7], [8], [13], [18], [19], [22], [23]) are inefficient and utterly vulnerable to OOD evasion attacks.

OOD Detector	Benign	Adversarial to Naive	Adversarial to Adv.Robust
ODIN	24.7%	100%	49.2%
Network Agnostophobia	22.2%	97.1%	57.3%
Mahlanobis Detector	7.3%	88.6%	57.7%
Trust Score	90.5%	82.5%	94.2%
AE Detector	53.5%	82.5%	53.5%
Deep-SVDD	58.6%	99.5%	96.7%

Table 1: Report of FPR(% False Positive Rate) on CIFAR-10 trained OOD Detector with ImageNet as source of OOD data(this table of results is taken from [2]). The higher the FPRs, the more successful the evasion of the OOD detector will have been.

As we may observe from the table, benign OOD samples (second column) result in a relatively low FPR for at least the first three detectors. Once, the OOD samples become adversarial, the naively trained detectors result in huge FPR rates. Although when adversarially trained, this detectors result in better (lower) FPRs, the effect of adversarial perturbation is still determinant in the performance of that specific detector. For more detailed results on other OOD datasets for the same detectors, please refer to section 5 of [2].

3. Approach

As mentioned, our approach will be novel insofar as it will be based on the two following detection methods: *Gradient Magnitude Analysis*, explored by Vodrahalli et al. in [25] in the context of the correlation between gradient magnitudes and how hard the corresponding samples are to learn for a classifier or regressor, and *Hessian Spectral Norm Analysis*, both of which we hereby describe. For the remaining of the description, in the following section, let us consider a loss function $\mathcal{L}$ as described in 1.1. In order to give more intuition about the two methods and their geometric interpretation, let us explain them in a more qualitative perspective. Beginning with *Gradient Magnitude Analysis* (GMA), "training" a model f to f_* means to have minimized a loss metric. Thus, if the function f is written as a form similar to $f(\boldsymbol{w}^T\boldsymbol{x})$, then, we expect the gradient of the loss function to be close to zero for samples that are far from the decision boundary as they are the easier ones to learn, whereas, we expect it to be higher for samples close to the boundary. Indeed, in [25], Vodrahalli et al. have used this very technique to distinguish between easy and hard samples to learn at training and inference time. We would expect the OOD samples to be amongst close to the hard samples to learn and that is why we have initially used this technique. Even if this is likely to be true, however, we have theoretically and empirically shown that it does not mean that the gradient might be a distinguishing factor for IN vs. OOD samples. On the other hand, *Hessian Spectral Norm Analysis* (HNA) leads us to the following question: what does the curvature of the loss surface for a minimized $\boldsymbol{w}$ evaluated at a specific sample tell us about the sample itself? We explore this question in depth both from a simplified theoretical perspective and from a thorough empirical evaluation on Deep Convolutional Neural Networks.

4. Implementation

The general approach to the problem is best summarised by Figure 1. Let us now give a formal definition to the problem and the useful facts that could give us deep insights for the theoretical underpinnings of the problem we are trying to solve. Below, the acronyms GMA and HNA, as mentioned previously, stand for Gradient Magnitude Analysis and Hessian Spectral Norm Analysis respectively.

4.1. Problem Statement: GMA and HNA

Definition 4.1. The gradient of the loss function $\mathcal{L}$ with respect to the last DNN layer weight matrix $\boldsymbol{w}$, learned after the training phase, evaluated at a specific sample point $\boldsymbol{x}$ is

$$\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}) := \frac{\partial}{\partial \boldsymbol{w}}\mathcal{L}(\boldsymbol{x}) \tag{5}$$

Thus, the gradient norm is

$$||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| := \Big[\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})\Big]^T\Big[\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})\Big] \tag{6}$$

Definition 4.2. The Hessian of the loss function $\mathcal{L}$ with respect to the last DNN layer weight matrix $\boldsymbol{w}$, learned after the training phase, evaluated at a specific sample point $\boldsymbol{x}$ is

$$\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}) := \frac{\partial}{\partial \boldsymbol{w}\boldsymbol{w}^T}\mathcal{L}(\boldsymbol{x}) \tag{7}$$

Let us, thus, define Hessian Spectral Norm as the square root of its Gram matrix maximum eigenvalue λ_{max} or equivalently its largest singular value σ_{max}

$$||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| := \sqrt{\lambda_{max}\bigg(\Big[\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})\Big]^T\Big[\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})\Big]\bigg)} = \sigma_{max}\Big(\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})\Big) \tag{8}$$

Let us now consider a learning function f and its optimally trained counterpart, as introduced

in 1.1, f_*, which is optimal in terms of the learned weight matrix $\boldsymbol{w}$, then, we will have that the test set $\mathcal{D}_{test}$ is the union of the disjoint IN test set $\mathcal{D}_{test,IN}$ and OOD test set $\mathcal{D}_{test,OOD}$ $\mathcal{D}_{test} = \mathcal{D}_{test,IN} \cup \mathcal{D}_{test,OOD}$. For all $\boldsymbol{x}_{IN} \in \mathcal{D}_{test,IN}$ and for all $\boldsymbol{x}_{OOD} \in \mathcal{D}_{test,OOD}$, we have the two following cases:

- *Gradient Magnitude Analysis* (GMA): let us compute $||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}_{IN})||$ and $||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}_{OOD})||$, hence, obtaining two empirical distributions for the gradient norms, one for in-distribution data and the other for out-of-distribution: $\hat{\Omega}_{||\nabla||,IN}$ and $\hat{\Omega}_{||\nabla||,OOD}$. We seek to find a threshold $\tau_{||\nabla||}$ that could optimally separate the two distributions, meaning that the threshold satisfies the following minimization problem, where, without loss of generality, we have assumed that the $\hat{\Omega}_{||\nabla||,IN}$ empirical mean $\hat{\mathbb{E}}[\hat{\Omega}_{||\nabla||,IN}]$ is at most as large as the $\hat{\Omega}_{||\nabla||,OOD}$ empirical mean $\hat{\mathbb{E}}[\hat{\Omega}_{||\nabla||,OOD}]$, $\hat{\mathbb{E}}[\hat{\Omega}_{||\nabla||,IN}] \leq \hat{\mathbb{E}}[\hat{\Omega}_{||\nabla||,OOD}]$

$$\tau_{||\nabla||} \in arg\min_{\tau \in \mathbb{R}} \int_{-\infty}^{\tau} \hat{\Omega}_{||\nabla||,OOD}(z)dz + \int_{\tau}^{\infty} \hat{\Omega}_{||\nabla||,IN}(z)dz \tag{9}$$

- *Hessian Spectral Norm Analysis* (HNA): let us compute $||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}_{IN})||$ and $||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}_{OOD})||$, hence, obtaining two empirical distributions for the Hessian Spectral norms, one for in-distribution data and the other for out-of-distribution: $\hat{\Omega}_{||\mathcal{H}||,IN}$ and $\hat{\Omega}_{||\mathcal{H}||,OOD}$. We seek to find a threshold $\tau_{||\mathcal{H}||}$ that could optimally separate the two distributions, meaning that the threshold satisfies the following minimization problem, where, without loss of generality, we have assumed that the $\hat{\Omega}_{||\mathcal{H}||,IN}$ empirical mean $\hat{\mathbb{E}}[\hat{\Omega}_{||\mathcal{H}||,IN}]$ is at most as large as the $\hat{\Omega}_{||\mathcal{H}||,OOD}$ empirical mean $\hat{\mathbb{E}}[\hat{\Omega}_{||\mathcal{H}||,OOD}]$, $\hat{\mathbb{E}}[\hat{\Omega}_{||\mathcal{H}||,IN}] \leq \hat{\mathbb{E}}[\hat{\Omega}_{||\mathcal{H}||,OOD}]$

$$\tau_{||\mathcal{H}||} \in arg\min_{\tau \in \mathbb{R}} \int_{-\infty}^{\tau} \hat{\Omega}_{||\mathcal{H}||,OOD}(z)dz + \int_{\tau}^{\infty} \hat{\Omega}_{||\mathcal{H}||,IN}(z)dz \tag{10}$$

In the following subsection, we will establish upper and lower bounds on the magnitude of the two different norms in the case of logistic loss.

4.2. GMA vs. HNA: the Logistic Loss Case

Let us consider a simple binary classification problem, which aims at finding the optimal hyperplane which separates two classes: $\boldsymbol{x} \in \mathbb{R}^p$ and $y \in \{0, 1\}$.

Note. In the case of Deep Neural Networks, rather than having a regular input vector $\boldsymbol{x}$, we would rather have a concatenation of its feature maps representations $\boldsymbol{\Phi}(\boldsymbol{x})$.

Let us make the following assumptions, where $\boldsymbol{x}_{IN}, \boldsymbol{x}_{OOD} \in \mathbb{R}^p$.

Assumption 1. $\boldsymbol{x}_{IN} \preceq \boldsymbol{M}_{IN} \prec \boldsymbol{\infty}$.
Assumption 2. $\boldsymbol{x}_{OOD} \preceq \boldsymbol{M}_{OOD} \prec \boldsymbol{\infty}$.
Assumption 3. $\boldsymbol{x}_{OOD} = \boldsymbol{\psi}(\boldsymbol{x}_{IN})$.

The first two assumptions above are entailing that, any in-distribution sample and any out-of-distribution sample is contained in balls of radius $||\boldsymbol{M}_{IN}||$ and $||\boldsymbol{M}_{OOD}||$ respectively. The third is imposing that any out-of-distribution point is a generic map of an in-distribution as long as it respects the imposed bounds.

Definition 4.3. Let us define logistic function as

$$\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}) := \frac{1}{1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}} \tag{11}$$

Hereby, $\boldsymbol{w} \in \mathbb{R}^p$, and, is, thus, a vector.

Definition 4.4. Let us define logistic loss as

$$\mathcal{L}(x) = y \log \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}) + (1 - y) \log(1 - \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})) \tag{12}$$

Lemma 4.1. $\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}) = (y - \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))\boldsymbol{x}$

Proof. From the definition, we have

$$\begin{aligned}\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}) &= \left(\frac{y}{\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})} - \frac{(1-y)}{1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})}\right)\boldsymbol{\sigma}^{'}(\boldsymbol{w}^T\boldsymbol{x})\boldsymbol{x} = \left(\frac{y}{\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})} - \frac{(1-y)}{1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})}\right)\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))\boldsymbol{x} \\ &= \frac{[y - y\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}) + y\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})]\boldsymbol{\sigma}^{'}(\boldsymbol{w}^T\boldsymbol{x})\boldsymbol{x}}{\boldsymbol{\sigma}^{'}(\boldsymbol{w}^T\boldsymbol{x})} - \frac{\boldsymbol{\sigma}^{'}(\boldsymbol{w}^T\boldsymbol{x})\boldsymbol{x}}{1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})} = \frac{(y-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))}{\boldsymbol{\sigma}^{'}(\boldsymbol{w}^T\boldsymbol{x})}\boldsymbol{\sigma}^{'}(\boldsymbol{w}^T\boldsymbol{x})\boldsymbol{x} \\ &= (y - \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))\boldsymbol{x}\end{aligned}$$

Insofar as, $\boldsymbol{\sigma}^{'}(\boldsymbol{w}^T\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))$.

Corollary. $||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| = (y - \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))^2||\boldsymbol{x}||$

Proof. From the definition, the above statement follows.

Lemma 4.2. $\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))\boldsymbol{x}\boldsymbol{x}^T$

Proof. From the definition, we have

$$\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x}) := \frac{\partial}{\partial \boldsymbol{w}\boldsymbol{w}^T}\mathcal{L}(\boldsymbol{x}) = \nabla_{\boldsymbol{w}}(\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})) = \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))\boldsymbol{x}\boldsymbol{x}^T$$

Corollary. $||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| = \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))||\boldsymbol{x}||$

Proof. From the definition, we have

$$\begin{aligned}||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| &= \sigma_{max}\left(\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})\right) = \sigma_{max}(\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))\boldsymbol{x}\boldsymbol{x}^T) = \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))\sigma_{max}(\boldsymbol{x}\boldsymbol{x}^T) \\ &= \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))||\boldsymbol{x}||\end{aligned}$$

This is given to the fact that a rank-one matrix $\boldsymbol{A} = \boldsymbol{u}\boldsymbol{v}^T$ has SVD decomposition $\boldsymbol{U} = \frac{\boldsymbol{u}}{||\boldsymbol{u}||}$, $\boldsymbol{\Sigma} = ||\boldsymbol{u}||||\boldsymbol{v}||$ and $\boldsymbol{V} = \frac{\boldsymbol{v}}{||\boldsymbol{v}||}$ as in

$$\boldsymbol{A} = \frac{\boldsymbol{u}}{||\boldsymbol{u}||} \cdot ||\boldsymbol{u}||||\boldsymbol{v}|| \cdot \frac{\boldsymbol{v}^T}{||\boldsymbol{v}||}$$

Theorem 4.3. $||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| \in \left[\frac{||\boldsymbol{M}||}{e^{|\boldsymbol{w}^T\boldsymbol{M}|}+3}, ||\boldsymbol{x}||\right]$

Proof. For the upper and lower bound let us make the following considerations:

- Upper Bound:

$$||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| = \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))||\boldsymbol{x}|| \leq ||\boldsymbol{x}||$$

  given that $\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})) \leq 1$.

- Lower Bound:

$$||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| = \frac{||\boldsymbol{x}||}{e^{\boldsymbol{w}^T\boldsymbol{x}}+2+e^{-\boldsymbol{w}^T\boldsymbol{x}}} = \frac{e^{\boldsymbol{w}^T\boldsymbol{x}}||\boldsymbol{x}||}{e^{2\boldsymbol{w}^T\boldsymbol{x}}+2e^{\boldsymbol{w}^T\boldsymbol{x}}+1} \geq \frac{||\boldsymbol{x}||}{e^{|\boldsymbol{w}^T\boldsymbol{x}|}+3} \geq \frac{||\boldsymbol{M}||}{e^{|\boldsymbol{w}^T\boldsymbol{M}|}+3}$$

Corollary. *The tightness of the* $\frac{||\boldsymbol{M}||}{e^{|\boldsymbol{w}^T\boldsymbol{M}|}+3}$ *lower bound is at most* $\frac{1}{8}(5\sqrt{5}-11)$.

Proof. Let us nullify the derivative of the difference $t(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})(1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x})) - \frac{||\boldsymbol{x}||}{e^{|\boldsymbol{w}^T\boldsymbol{x}|}+3}$ as in $\frac{\partial}{\partial \boldsymbol{x}}t(\boldsymbol{x}) = 0$. This yields the maximum $t_{max} = \frac{1}{8}(5\sqrt{5}-11)$.

Theorem 4.4. *The following chain of inequalities holds*

$$||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| \leq \frac{||\boldsymbol{x}||}{e^{|\boldsymbol{w}^T\boldsymbol{x}|}+3} \leq ||\mathcal{H}_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| \tag{13}$$

Proof. As the second inequality has been shown in a previous theorem, we are left to show the first one. Let us first make the following consideration on the gradient norm and the sign of $\boldsymbol{w}^T\boldsymbol{x}$.

The learnt hyperplane will classify samples such that $\boldsymbol{w}^T\boldsymbol{x} \leq 0$ as $y = 0$ and $y = 1$ otherwise. Let us recall that $||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| = (y-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))^2||\boldsymbol{x}||$, thus, when $\boldsymbol{w}^T\boldsymbol{x} > 0$, we are only interested in $y = 1$ and, hence, $||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| = (1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))^2||\boldsymbol{x}||$, and, otherwise, $||\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{x})|| = \boldsymbol{\sigma}^2(\boldsymbol{w}^T\boldsymbol{x})||\boldsymbol{x}||$.

Therefore, what we are left to show is that, given that $||\boldsymbol{x}||$ is a common term,

$$\begin{cases} \boldsymbol{\sigma}^2(\boldsymbol{w}^T\boldsymbol{x}) \leq \frac{1}{e^{|\boldsymbol{w}^T\boldsymbol{x}|}+3} & \text{for } \boldsymbol{w}^T\boldsymbol{x} \leq 0 \\ (1-\boldsymbol{\sigma}(\boldsymbol{w}^T\boldsymbol{x}))^2 \leq \frac{1}{e^{|\boldsymbol{w}^T\boldsymbol{x}|}+3} & \text{for } \boldsymbol{w}^T\boldsymbol{x} > 0 \end{cases}$$

This is equivalent to

$$\begin{cases} e^{|\boldsymbol{w}^T\boldsymbol{x}|}+3 \leq 1+2e^{-\boldsymbol{w}^T\boldsymbol{x}}+e^{-2\boldsymbol{w}^T\boldsymbol{x}} & \text{for } \boldsymbol{w}^T\boldsymbol{x} \leq 0 \\ e^{-2\boldsymbol{w}^T\boldsymbol{x}}(e^{|\boldsymbol{w}^T\boldsymbol{x}|}+3) \leq 1+2e^{-\boldsymbol{w}^T\boldsymbol{x}}+e^{-2\boldsymbol{w}^T\boldsymbol{x}} & \text{for } \boldsymbol{w}^T\boldsymbol{x} > 0 \end{cases}$$

Which, in turn, is equivalent to

$$\begin{cases} e^{\boldsymbol{w}^T\boldsymbol{x}}(2e^{\boldsymbol{w}^T\boldsymbol{x}}-1) \leq 1 & \text{for } \boldsymbol{w}^T\boldsymbol{x} \leq 0 \\ 1 \leq e^{\boldsymbol{w}^T\boldsymbol{x}}(e^{2\boldsymbol{w}^T\boldsymbol{x}}+2e^{\boldsymbol{w}^T\boldsymbol{x}}-2) & \text{for } \boldsymbol{w}^T\boldsymbol{x} > 0 \end{cases}$$

The first inequality holds given that both factors are smaller than or equal to 1 for $\boldsymbol{w}^T\boldsymbol{x} \leq 0$. The second holds given that both factors are greater than or equal to 1 for $\boldsymbol{w}^T\boldsymbol{x} > 0$. Thus, the theorem directly follows.

The proved chain of inequalities can be visualized in the following one-dimensional plot:

Figure 2: The green curve represents the *Hessian Spectral Norm*, the black one the its lower bound and the red and blue curves the *Gradient Norm* when $\boldsymbol{w}^T\boldsymbol{x} > 0$ and $\boldsymbol{w}^T\boldsymbol{x} \leq 0$ respectively, where $\boldsymbol{w} = 1$.

The meaning of the Theorem 3.4 is that, unlike the gradient magnitude, the Hessian Spectral Norm is strictly greater (unless $\boldsymbol{w}^T\boldsymbol{x} = 0$, in which case equality holds), and thus, we expect the distribution of the Hessian Spectral Norm of the IN samples versus the OOD samples to be more distinguishing. In other words, if we denote by ρ^1, ρ^2 what follows, we would expect

$\rho^2 > \rho^1$:

$$\rho^1 = \frac{||\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{x_{IN}})||}{||\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{x_{OOD}})||} \tag{14}$$

$$\rho^2 = \frac{||\mathcal{H}_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{x_{IN}})||}{||\mathcal{H}_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{x_{OOD}})||} \tag{15}$$

This general framework provides good intuition and some formal arguments about why HNA might be a better approach to OOD detection than GMA could be. The following section gives strong empirical support to HNA over GMA. By no means, however, we imply that the two are necessarily alternative to each other, insofar as, they are assessing two different aspects of the loss function surface for IN and OOD samples: GMA investigates the topology of the flat valleys in the loss function surface, whilst, HNA analyzes the concavity (thus, curvature) of the loss function surface. As a matter of fact, as we will point out in the conclusion, a possible expansion of this work could be that of combining preexisting methods, largely based on Confidence Thresholding or similar techniques to GMA, and, our HNA.

5. Evaluation

5.1. Experiment design

The experimental design consists of the following steps which are implemented through the renowned Python Library named TensorFlow [1] and run on a NVIDIA GPU CUDA Environment [14], as explained visually by Figure 3:

1. Split in-distribution dataset into train and test set.
2. Train Wide-ResNet (which can be found at https://github.com/MadryLab/cifar10_challenge) on training set.
3. Reshape OOD datasets to the CIFAR-10 image shape (32×32 pixels).
4. Perform GMA and HNA on IN (CIFAR-10 test set) and OOD test samples in each of the following four settings:
   - Naively trained network on benign data samples.
   - Naively trained network on adversarially perturbed data samples.
   - Adversarially robust network on benign data samples.
   - Adversarially robust network on adversarially perturbed data samples.

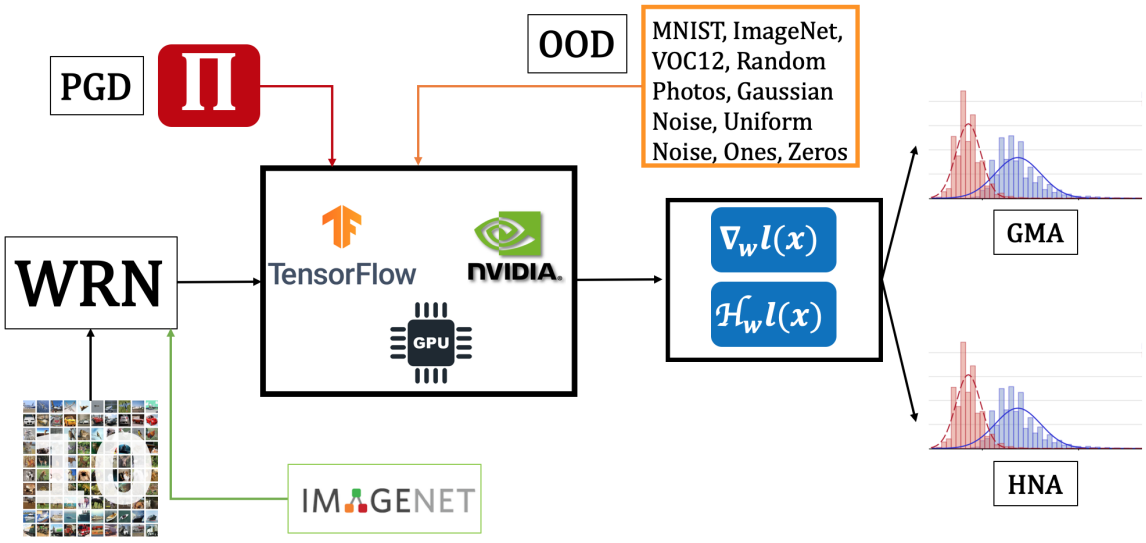


Figure 3: Experimental Design for Robust OOD Detection in OWL Settings. The full implementation can be found at the following GitHub link: https://github.com/matteorusso/robust_hessian_ood

5.2. Data

The data consists of the in-distribution and out-of-distribution datasets:

- IN: CIFAR-10 data [15].
- OOD: MNIST [17], ImageNet [6], VOC-12 [9], Google Images, Gaussian Noise Images, Uniform Noise Images, Images of Ones and Images of Zeros.

5.3. Metrics

The metrics we have decided to introduce are the following ones:

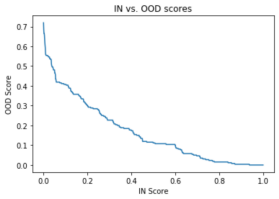
- **Graph of OOD score vs. IN score**: this portrays the percentage of OOD points that are detected as such when a varying percentage of IN points are detected as such.
- **Graph of F1 score vs. IN score**: this portrays the product between OOD score and IN score when a varying percentage of IN points are detected as such.
- **Histogram of IN vs. OOD norms** (whether Gradient related or Hessian related).
- **90% IN**: the OOD score when the IN score is 90%.
- **Best IN-Score**: value in correspondence of maximum in the **Graph of F1 score vs. IN score**.
- **Best F1-Score**: maximum in the **Graph of F1 score vs. IN score**.
- **Best OOD-Score**: ratio between **Best F1-Score** and **Best IN-Score**.

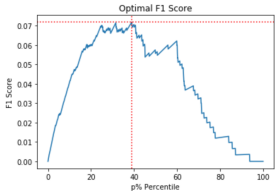
5.4. Qualitative results

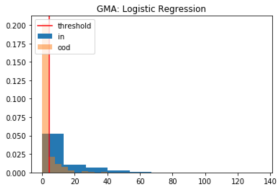
The qualitative results refer to the Logistic Case example, theoretically described in Section 4.2. We have used the SpamBase [12] dataset, executing the same exact pipeline of Section 5.1, with the exception of train a Linear Logistic Classifier rather than a Wide-ResNet and on benign OOD samples only. As we can observe from the graphs below, and especially in the histogram, our theoretical intuition about how the Hessian based detector performs when compared to the Gradient based one, are confirmed. As a matter of fact, we can see how the optimal threshold (in terms of F1 score) is capable of detecting many more OOD points for

the same amount of IN points detected as such.

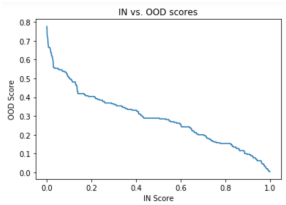
GMA on Logistic Classifier

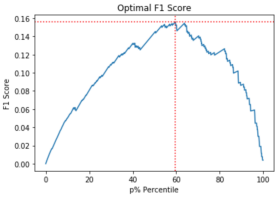


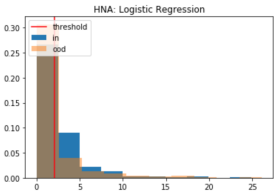




HNA on Logistic Classifier







5.5. Quantitative results and comparisons

Following Section 2 metric evaluation and table display, we have the following tables, where, all the statistics reported are in the format (GMA, HNA), meaning what the two different methods have yielded as results. Although from the tables and graphs we are not able to claim that our method is strictly better than the best of the previous work done on the OOD detection spectrum (namely Mahalanobis Detector), we can make the following strong claims:

- Regarding table 2, we can safely assert that gradient based detection performs strictly worse on average on OOD benign when compared to Hessian based detection. Since several approaches in the related work rely on gradient based approaches, HNA performs better than several previous approaches.

- Regarding table 3, we can safely assert that GMA and HNA perform equally poorly insofar the network has not been trained on adversarial points and thus, it is weak to these types of adversarial attacks even when the perturbation is applied to IN points.
- Regarding table 4, we can safely assert that gradient based approaches when the network is trained on adversarial examples, perform very poorly on benign points, whereas HNA performs consistently across different types of trainings.
- Regarding table 5, we can safely assert that gradient based approaches when the network is trained on adversarial examples, perform much better than in the previous case, even if there is a dramatic failure rate if compared to the first case. When the network is trained on adversarial examples, this training is gradient based as per [], thus, even though HNA performs in an aligned manner to GMA, we expect that if the robust training where to be Hessian based, the performance rates would be much higher.
- Regarding table 6, for benign samples on naively trained networks, we observe a stability property for the Hessian based detection. In particular, for non-synthetic datasets, HNA consistently finds the optimal threshold of separation between norms distributions of IN vs. OOD points to be around 10^{-1}.

OOD Dataset	90% IN	Best IN-Score	Best OOD-Score	Best F1-Score
MNIST	(20%, 34%)	(73%, 73%)	(90%, 88%)	(65%, 64%)
Imagenet	(20%, 47%)	(76%, 75%)	(89%, 89%)	(68%, 67%)
VOC-12	(20%, 45%)	(76%, 75%)	(93%, 94%)	(70%, 70%)
Google Images	(20%, 49%)	(75%, 74%)	(87%, 88%)	(65%, 65%)
Gaussian Noise	(20%, 80%)	(80%, 80%)	(95%, 96%)	(76%, 77%)
Uniform Noise	(20%, 80%)	(79%, 81%)	(95%, 97%)	(75%, 78%)
Zeros Images	(100%, 100%)	(98%, 93%)	(100%, 100%)	(98%, 93%)
Ones Images	(100%, 100%)	(98%, 93%)	(100%, 100%)	(98%, 93%)

Table 2: CIFAR-10 naively trained Wide-ResNet on benign IN and OOD test samples

OOD Dataset	90% IN	Best IN-Score	Best OOD-Score	Best F1-Score
MNIST	(0%, 0%)	(33%, 28%)	(35%, 40%)	(12%, 11%)
Imagenet	(0%, 0%)	(15%, 16%)	(40%, 41%)	(6%, 6%)
VOC-12	(0%, 0%)	(41%, 14%)	(50%, 38%)	(21%, 5%)
Google Images	(0%, 0%)	(16%, 19%)	(30%, 28%)	(5%, 5%)
Gaussian Noise	(0%, 0%)	(34%, 30%)	(73%, 79%)	(25%, 24%)
Uniform Noise	(5%, 5%)	(59%, 61%)	(88%, 84%)	(52%, 51%)
Zeros Images	(9%, 10%)	(41%, 53%)	(50%, 41%)	(21%, 22%)
Ones Images	(9%, 10%)	(53%, 52%)	(41%, 40%)	(22%, 21%)

Table 3: CIFAR-10 naively trained Wide-ResNet on adversarially perturbed IN and OOD test samples

OOD Dataset	90% IN	Best IN-Score	Best OOD-Score	Best F1-Score
MNIST	(37%, 35%)	(33%, 78%)	(35%, 81%)	(12%, 63%)
Imagenet	(11%, 46%)	(15%, 37%)	(40%, 43%)	(6%, 16%)
VOC-12	(11%, 51%)	(41%, 32%)	(50%, 38%)	(21%, 12%)
Google Images	(15%, 48%)	(16%, 50%)	(30%, 51%)	(5%, 25%)
Gaussian Noise	(0%, 66%)	(34%, 29%)	(73%, 94%)	(25%, 27%)
Uniform Noise	(0%, 67%)	(59%, 31%)	(88%, 93%)	(52%, 29%)
Zeros Images	(100%, 100%)	(41%, 41%)	(50%, 100%)	(21%, 41%)
Ones Images	(100%, 100%)	(53%, 41%)	(41%, 100%)	(22%, 41%)

Table 4: CIFAR-10 adversarially trained Wide-ResNet on benign IN and OOD test samples

In the Appendix, one could find, for all the OOD datasets, in the four cases of naively trained model on benign OOD samples, naively trained model on adversarially perturbed OOD samples, adversarially trained model on benign OOD samples, and adversarially trained model on adversarial OOD samples, the **Graph of OOD score vs. IN score**, the **Graph**

OOD Dataset	90% IN	Best IN-Score	Best OOD-Score	Best F1-Score
MNIST	(52%, 53%)	(78%, 76%)	(72%, 84%)	(56%, 64%)
Imagenet	(12%, 12%)	(44%, 37%)	(93%, 44%)	(41%, 16%)
VOC-12	(12%, 14%)	(45%, 28%)	(86%, 44%)	(39%, 12%)
Google Images	(18%, 10%)	(49%, 43%)	(89%, 54%)	(44%, 23%)
Gaussian Noise	(0%, 0%)	(51%, 28%)	(99%, 98%)	(51%, 27%)
Uniform Noise	(0%, 0%)	(50%, 29%)	(100%, 96%)	(50%, 28%)
Zeros Images	(0%, 0%)	(12%, 31%)	(72%, 65%)	(8%, 20%)
Ones Images	(0%, 0%)	(10%, 28%)	(80%, 68%)	(8%, 19%)

Table 5: CIFAR-10 adversarially trained Wide-ResNet on adversarially perturbed IN and OOD test samples

OOD Dataset	τ_{BB}	τ_{BA}	τ_{AB}	τ_{AA}
MNIST	(0.00, 0.03)	(0.00, 0.00)	(2.00, 4.42)	(2.01, 4.21)
Imagenet	(0.01, 0.07)	(0.00, 0.00)	(1.14, 1.73)	(1.10, 1.70)
VOC-12	(0.01, 0.06)	(0.00, 0.00)	(1.01, 1.51)	(1.14, 1.31)
Google Images	(0.01, 0.05)	(0.00, 0.00)	(1.23, 2.29)	(1.24, 1.98)
Gaussian Noise	(0.03, 0.29)	(0.00, 0.00)	(1.25, 1.37)	(1.29, 1.31)
Uniform Noise	(0.02, 0.36)	(0.00, 0.00)	(1.25, 1.44)	(1.26, 1.37)
Zeros Images	(3.75, 10.03)	(0.00, 0.00)	(2.48, 1.89)	(0.14, 1.46)
Ones Images	(3.75, 10.06)	(0.00, 0.00)	(2.48, 1.89)	(0.11, 1.31)

Table 6: GMA and HNA established thresholds to distinguish IN and OOD test samples, where the subscripts BB, BA, AB and AA mean respectively naively trained network on benign samples, naively trained network on adversarial samples, adversarially trained network on benign samples, adversarially trained network on adversarial samples

of F1 score vs. IN score and the **Histogram of IN vs. OOD norms** for both the GMA and the HNA detection frameworks.

6. Open-World Learning Issues and Public Policy Implications

The aim of this section is the one of showing how enormously vulnerable to OOD adversarial attacks real-world data driven devices and apparatuses truly are. We, hence, hereby suggest some policy remedies that could be adopted to prevent or cure .

6.1. Real-World Context

In order to contextualise the real-world applications this theoretical framework may have, let us consider the following case scenario in diagnostic biomedicine which is inspired by Finlayson et al.'s work on adversarial perturbations on melanoma data [10]. A technologically advanced hospital uses an artificially intelligent diagnosis test powered by a deep learning algorithm. This test is designed to determine whether, based on the patient's somatic traits (unlabelled data sample features) and other patients' history (labelled data sample features), the former suffers from one of the listed diseases (data samples label). Thus, the Deep Learning model is trained on that portion of the dataset that contains previous patients' history as in labels corresponding to specific somatic traits. After the model has been trained, the test has to be able to predict which of the labelled diseases the current patient suffers from. Nevertheless, it so happens that the patient's somatic traits do not correspond to any particular class of known diseases, but since the number of the latter is fixed, the model is constrained to predict one of those diseases, either giving false alarms to the patient or seriously underestimating the gravity of the situation, as illustrated in Figure 4.

Furthermore, let us think about a self-driving car which is mislead, by OOD adversarial attacks exploited by malicious agents, to interpret a STOP sign as a Priority Right sign. One may claim that the self-driving car industry is not yet fully at scale for a government to be worried to pass law bills that would regulate this sorts of issues. However, there are technology market sectors, such as the aircraft industry that are already indissolubly linked to Data Science, Machine Learning and Pattern Recognition. In this very case, aircrafts installed auto-piloting devices totally rely upon adaptive and learning controls that could

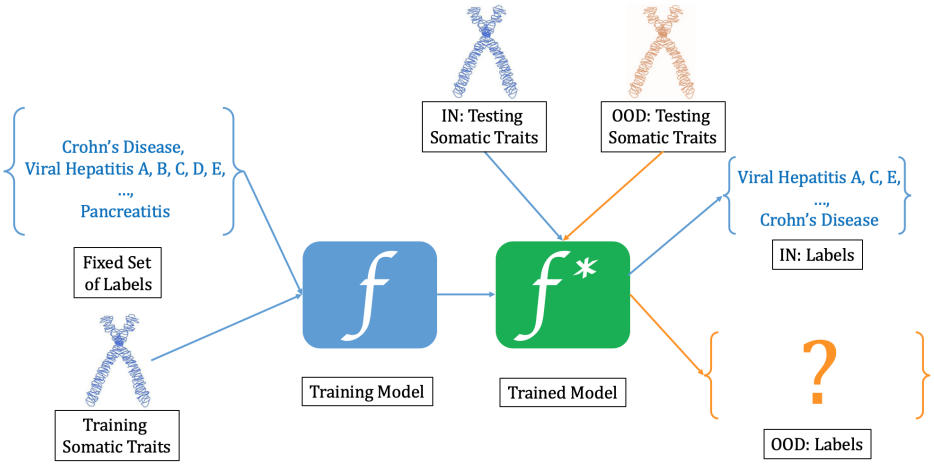


Figure 4: Diagnostic biomedicine example applied to gastroenterological diseases.

be easily mislead by such adversarial attacks and possibly cause tremendous amounts of fatalities. Another glaring example comes from network traffic attacks such as malware or ransomware where OOD adversarial learning could pose a serious threat to any modern malware or ransomware detection systems.

Putting forward this potential threat is the first step to raise awareness about this real danger. It is thus extremely important for central governments to pursue two parallel policy paths, i.e. a a technological and a normative one, because as the proverb says: "better safe than sorry". On one hand, new stricter regulations need to be approved so that (OOD) adversarial attacks perpetrators receive strong punishments. On the other, central governments need to invest in technologically advanced defences against these types of attacks.

6.2. Technological and Normative Policy Paths

In [3], a foundational paper in proposed normative advances against Adversarial Machine Learning, Calo et al. thoroughly analyze the current US legal framework with regards to adversarial attacks on Machine Learning powered systems, always drawing cases from CFAA(Computer Fraud and Abuse Act). The main point of Calo et al.'s report is one of claiming that there exists a substantial misalignment between the current hacking and computer fraud legislation and the machine learning posed threats to those very systems. In particular, the current CFAA or equivalent European cybersecurity and anti-hacking statutes

presuppose a virtual intrusion and subsequent physical or computing incapacitation of a specific device. Nevertheless, Adversarial Machine Learning in general and OOD Adversarial attacks in particular do not technically cause or subsume an intrusion in any kind of cyberspace, but are attacks which are able to statistically exploit the ML system vulnerabilities from the exterior at inference time. Adversarial Machine Learning, thus, would generate serious doubts on where and whether the CFAA normatives actually apply. In turn, this could be the cause of polarized countermeasures taken in court against this kind of cyber-offences. Moreover, this leads to lower consideration to the building of a comprehensive statute that would solidify and toughen the regulations on ML powered products and respective firms. On the same wavelength of thought, Shankar et al. wrote one of the most important papers that delve into the intersection of Adversarial Machine Learning and Law [16], whilst raising also some crucial ethical concerns.

6.3. Call to Action and Current Unsolved Issues

From what we have developed in this research work, the literature and the existing concerns which both Calo et al.'s and Shankar et al.'s have raised, we are able to present policy-driven proposals, which would bridge the gap for the growing need of regulators to understand how different Adversarial and OOD Adversarial Machine Learning Attacks are from traditional network or software intrusion attacks. We hereby posit them:

- **Technological**:
  1. Initiate the **NCAL** (National Commission of Adversarial Learning), a commission of Machine Learning experts aimed at crafting attacks in order to test the claimed Adversarial Machine Learning safety guarantees.
  2. In [11], Gilmer et al. propose a taxonomy of realistic adversarial attacks. We suggest adding benchmarks and rankings not only for adversarial attacks but also for out-of-distribution random samples injections, with, at the very least, the degree of testing we have carried out during this research paper (let us name it **AdvML-Tax**). This

would be really helpful to understand, both from a technical and legal perspective, the realm of such a branch of machine learning, henceforth, allowing for more rigorous and extensive regulatory directives.

- **Normative**:
  1. A country should make the building of Machine Learning powered systems comply with investigation standards. This means that the developed learning systems should take forensics into account: this could include "mechanisms to alert when the system is under adversarial attack, recommend appropriate logging, construct playbooks for incident response" [16]. during an attack and formulate remediation plan to recover to from the adversarial attack.
  2. Thanks to the above mentioned **AdvML-Tax**, it will be much easier to legal practitioners to develop a more granular set of norms to punish the perpetrators of such attacks and recognize the potential benefits of adversarial attacks.

The space of Adversarial Machine Learning has been timidly covered by current statutes, but there are evolving normative procedures that come from the CFAA text which could justly punish perpetrators of such attacks. In the case of benign OOD sample (and not adversarial), it is clear how, the potential detrimental effect relates to no perpetrators nor to any type of fault. Here, the call to action is purely technological and would be of great importance to regulators in order for them to enforce private companies to guarantee high safety standards in their products even with respect to out-of-distribution detection systems. For what regards the EU, the EU Commission has announced ([5]) that by the middle of 2019 a new and updated security framework, which is specific to Machine Learning vulnerabilities and potential attacks will be redacted and voted in the Strasbourg Parliament in order for it to be added to the EU GDPR (General Data Protection Regulation) directive [26]. In general, it is crucial to start developing or improving normative procedures that could protect a private citizen, enterprise or public entity from adversarial attacks.

7. Summary

7.1. Conclusions

In this research paper, we have narrated the general and powerful framework of Open-World Learning, acknowledging and assessing the potentialities and the deep flaws of nowadays learning systems. We have proposed two OOD detection systems of increasing performances that give provable bounds on their performance as proved. Both GMA and HNA give strong interpretability of results on detection and provide a self-contained OOD detection framework of reference.

7.2. Limitations

As we have pointed out throughout the paper, we have found the following three strong limitations that have to be addressed in future developments of the same work:

1. GMA and HNA are not at all robust to adversarial perturbations.
2. During our evaluation process, we have clipped all OOD sample images to CIFAR-10 size, hence, leading to an unavoidable noisy downsampling that could result in an unexpected underperformance.
3. Unfortunately, we cannot claim that either of GMA and HNA perform better than the best of previous approaches, despite give unified theoretical understanding of the OOD detection problem.

7.3. Future Work

Following the concerns raised in the previous section, we have found the following four steps to be the most important ones for a deeper understanding of the problem and a even more thorough testing of the OOD detection techniques:

1. We would like to expand the testing of GMA and HNA to Imagenet as IN dataset in order to prevent the aforementioned clipping problem of OOD image samples.

2. We would like to expand the testing of GMA and HNA to expand testing to 30 OOD datasets rather than just sticking to 8, 4 of which are synthetic.
3. We would like to combine HNA with previous existing approaches of OOD detection to increase their robustness and performance.
4. We would like to adversarially robustly train the Wide-ResNet with Hessian aware adversarial training techniques.

8. Acknowledgments

I would like to thank Professor Prateek Mittal for his incredibly helpful support and feedback during the development of our research process. I would also like to thank deeply Arjun Bhagoji and Vikash Sehwag for their wonderful help and great insights that gave me the opportunity to improve the quality of my Independent Work. Finally, I would like to thank the Madry Lab for the adversarial code that they have made available on GitHub for public use.

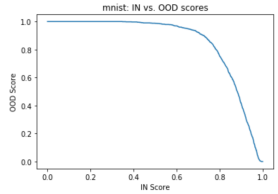
9. Ethics

This report represents my work in accordance to University regulations. I pledge my honor that I have not violated the Honor Code during the composition of my Independent Work.

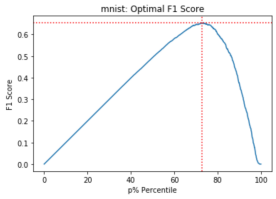
10. Appendices

10.1. GMA Results

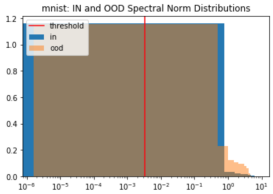
Naively Trained Model on Benign Samples



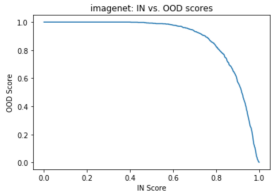
MNIST Scores



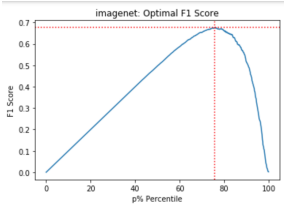
MNIST F1 Curve



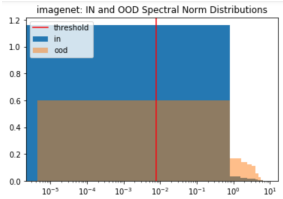
MNIST Distribution



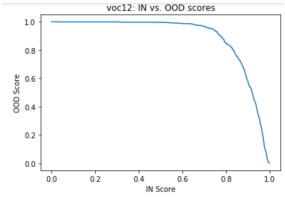
Imagenet Scores



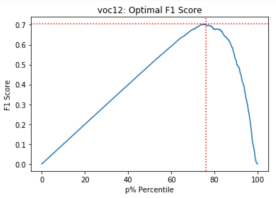
Imagenet F1 Curve



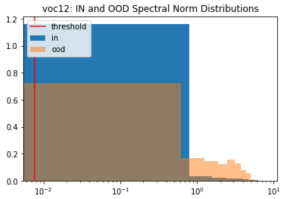
Imagenet Distribution



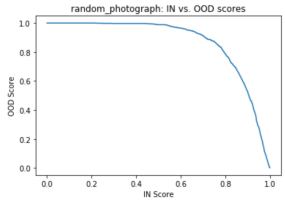
VOC-12 Scores



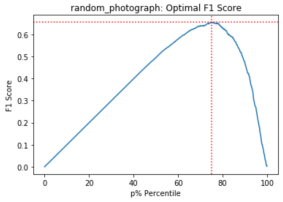
VOC-12 F1 Curve



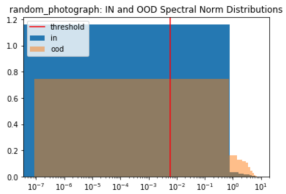
VOC-12 Distribution



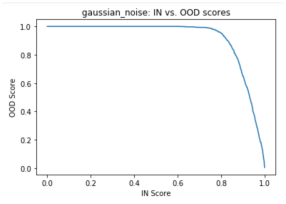
Google Images Scores



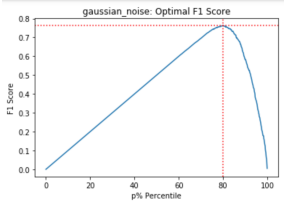
Google Images F1 Curve



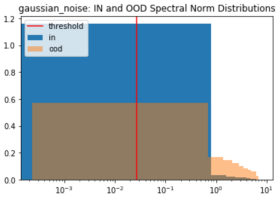
Google Images Distribution



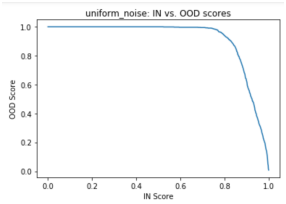
Gaussian Scores



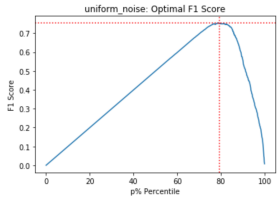
Gaussian F1 Curve



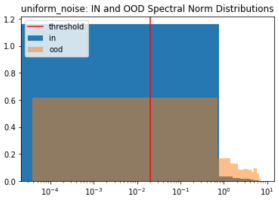
Gaussian Distribution



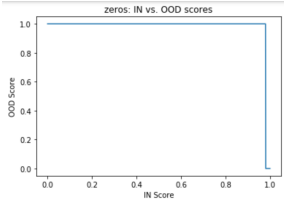
Uniform Scores



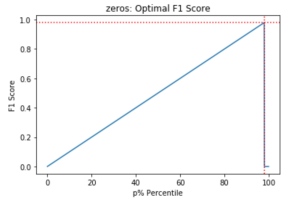
Uniform F1 Curve



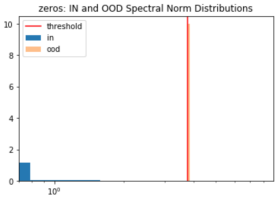
Uniform Distribution



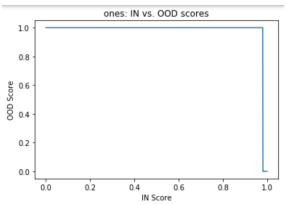
Zeros Scores



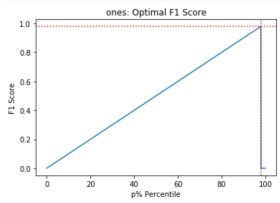
Zeros F1 Curve



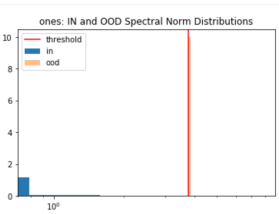
Zeros Distribution



Ones Scores

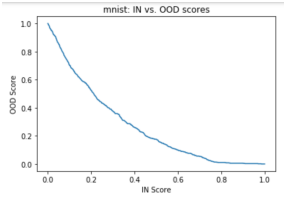


Ones F1 Curve

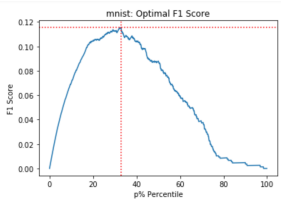


Ones Distribution

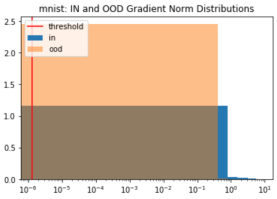
Naively Trained Model on Adversarial Samples



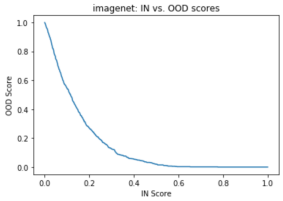
MNIST Scores



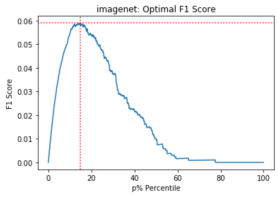
MNIST F1 Curve



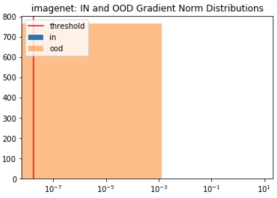
MNIST Distribution



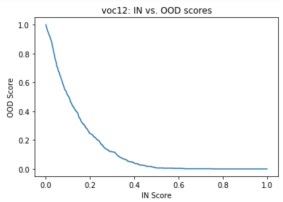
Imagenet Scores



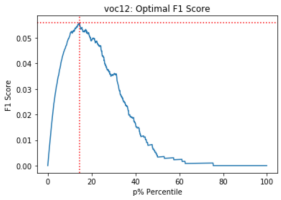
Imagenet F1 Curve



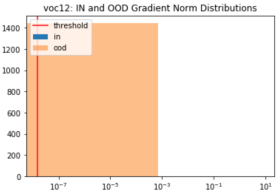
Imagenet Distribution



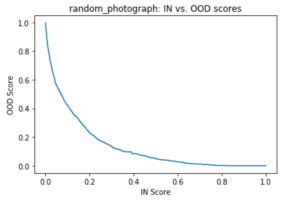
VOC-12 Scores



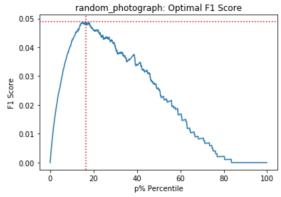
VOC-12 F1 Curve



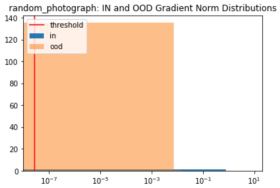
VOC-12 Distribution



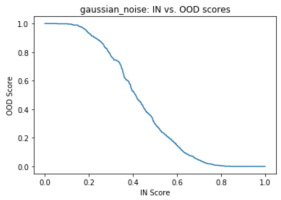
Google Images Scores



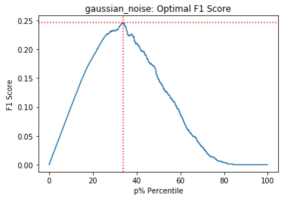
Google Images F1 Curve



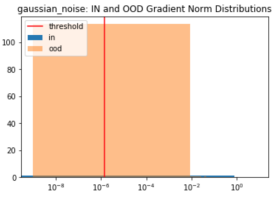
Google Images Distribution



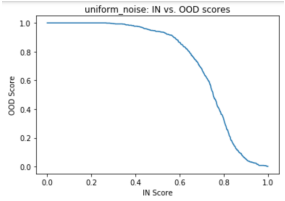
Gaussian Scores



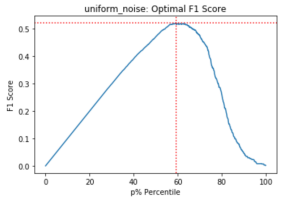
Gaussian F1 Curve



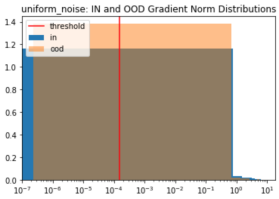
Gaussian Distribution



Uniform Scores

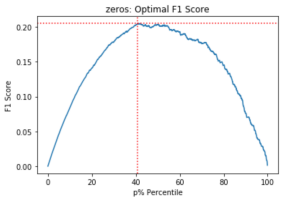


Uniform F1 Curve

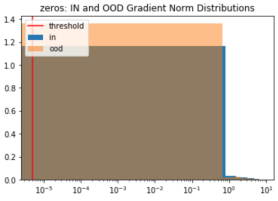


Uniform Distribution

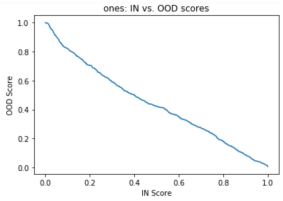
Zeros Scores



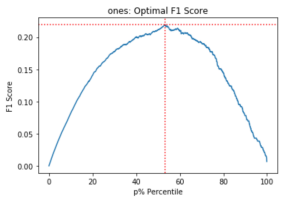
Zeros F1 Curve



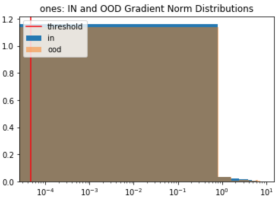
Zeros Distribution



Ones Scores

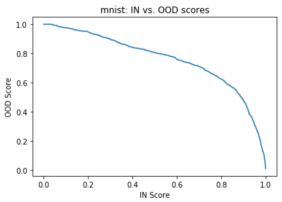


Ones F1 Curve

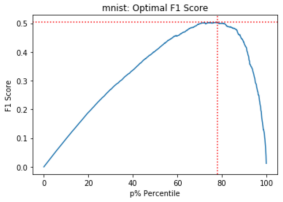


Ones Distribution

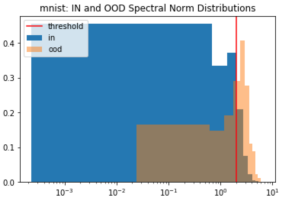
Adversarially Trained Model on Benign Samples



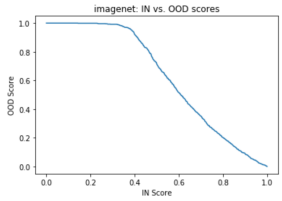
MNIST Scores



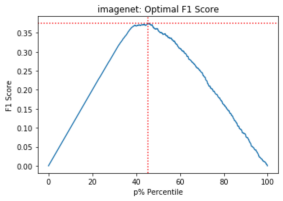
MNIST F1 Curve



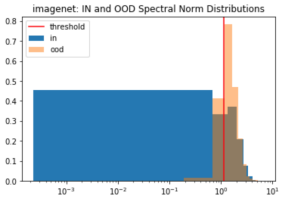
MNIST Distribution



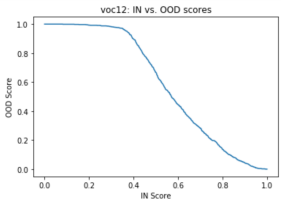
Imagenet Scores



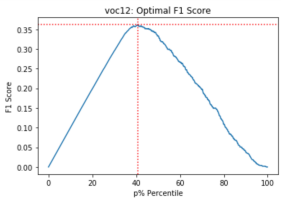
Imagenet F1 Curve



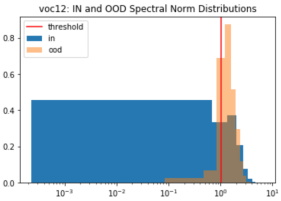
Imagenet Distribution



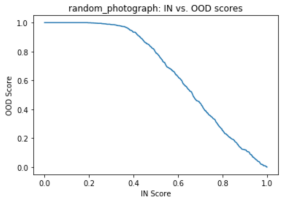
VOC-12 Scores



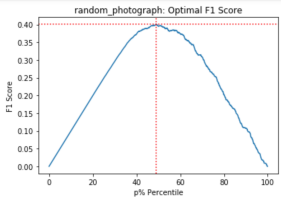
VOC-12 F1 Curve



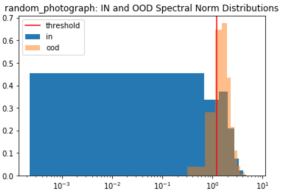
VOC-12 Distribution



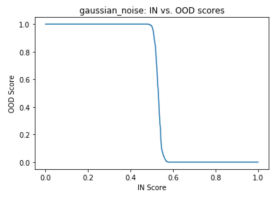
Google Images Scores



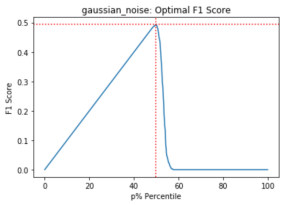
Google Images F1 Curve



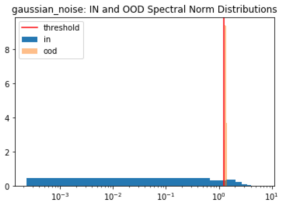
Google Images Distribution



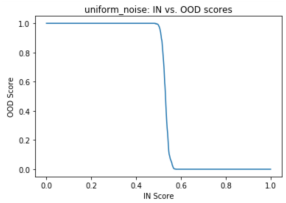
Gaussian Scores



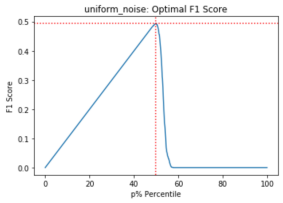
Gaussian F1 Curve



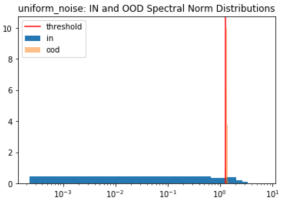
Gaussian Distribution



Uniform Scores

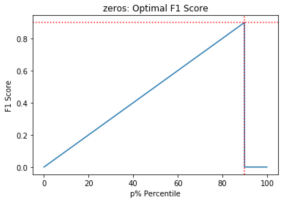


Uniform F1 Curve

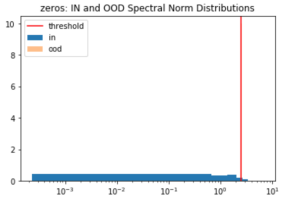


Uniform Distribution

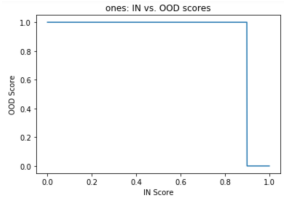
Zeros Scores



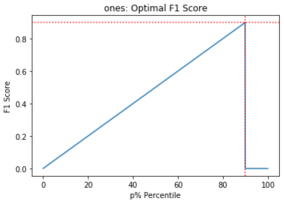
Zeros F1 Curve



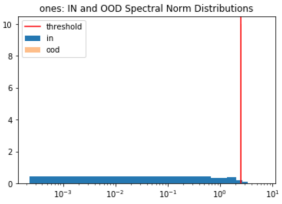
Zeros Distribution



Ones Scores

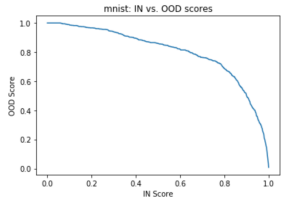


Ones F1 Curve

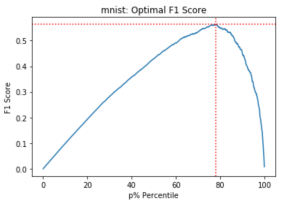


Ones Distribution

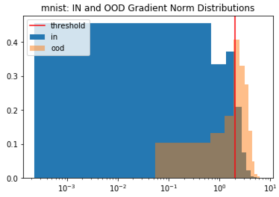
Adversarially Trained Model on Adversarial Samples



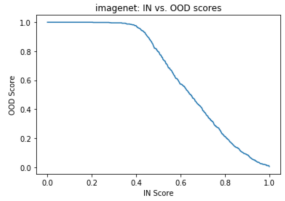
MNIST Scores



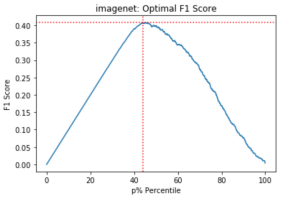
MNIST F1 Curve



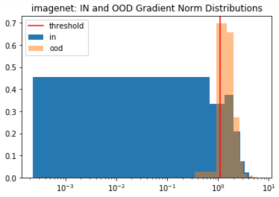
MNIST Distribution



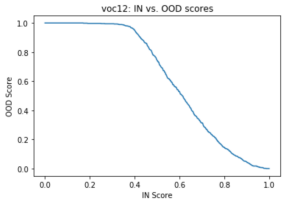
Imagenet Scores



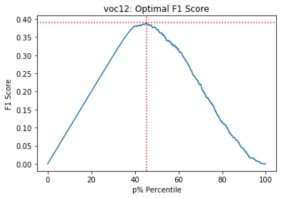
Imagenet F1 Curve



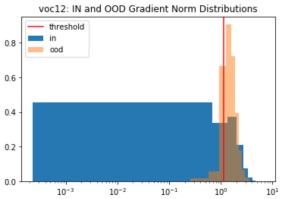
Imagenet Distribution



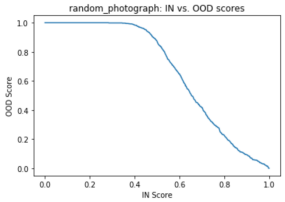
VOC-12 Scores



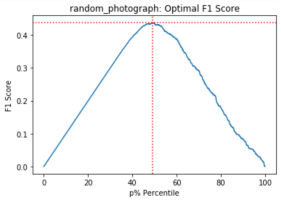
VOC-12 F1 Curve



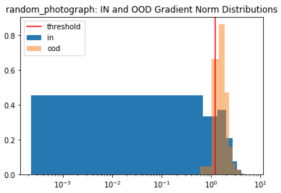
VOC-12 Distribution



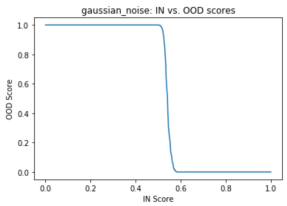
Google Images Scores



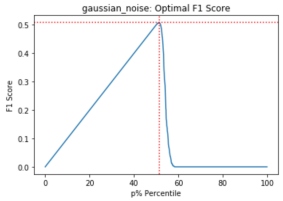
Google Images F1 Curve



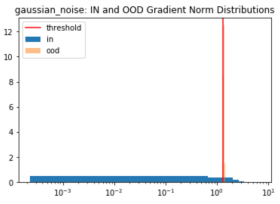
Google Images Distribution



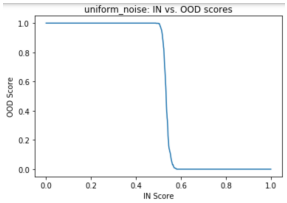
Gaussian Scores



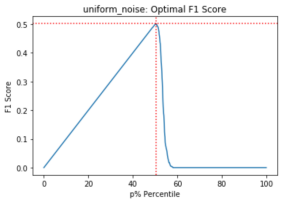
Gaussian F1 Curve



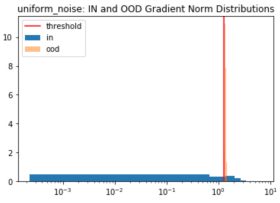
Gaussian Distribution



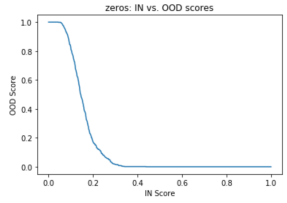
Uniform Scores



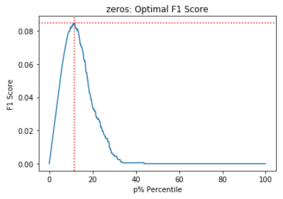
Uniform F1 Curve



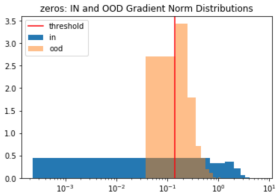
Uniform Distribution



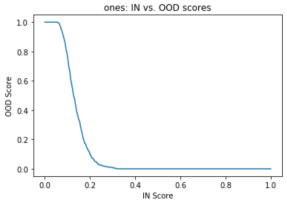
Zeros Scores



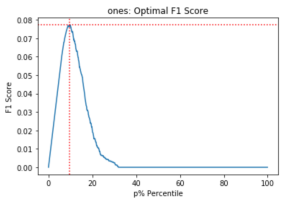
Zeros F1 Curve



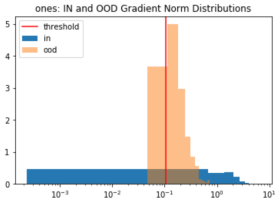
Zeros Distribution



Ones Scores



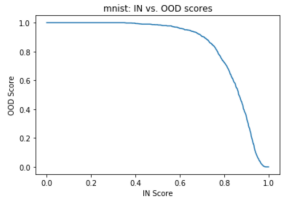
Ones F1 Curve



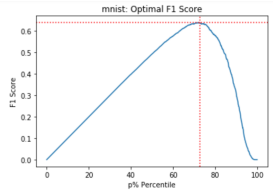
Ones Distribution

10.2. HNA Results

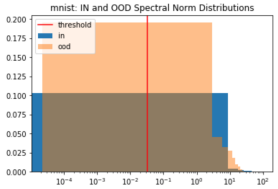
Naively Trained Model on Benign Samples



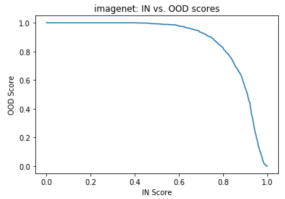
MNIST Scores



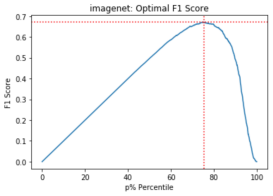
MNIST F1 Curve



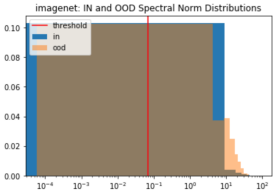
MNIST Distribution



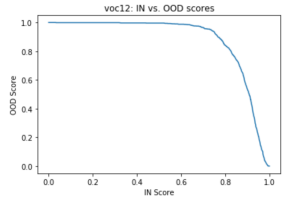
ImageNet Scores



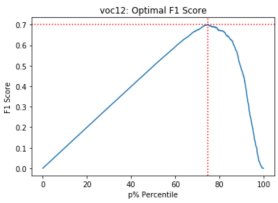
ImageNet F1 Curve



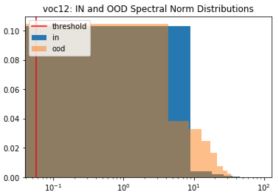
ImageNet Distribution



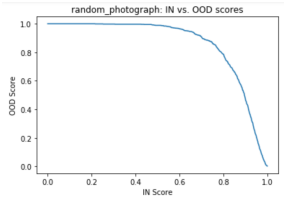
VOC-12 Scores



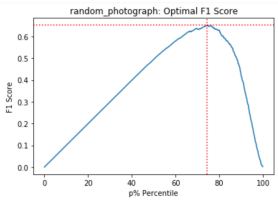
VOC-12 F1 Curve



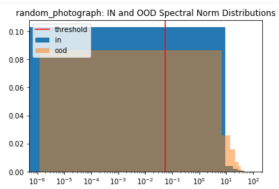
VOC-12 Distribution



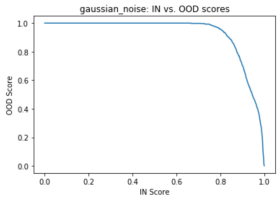
Google Images Scores



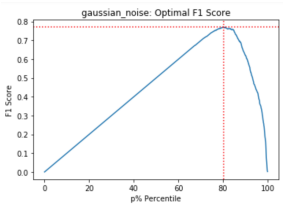
Google Images F1 Curve



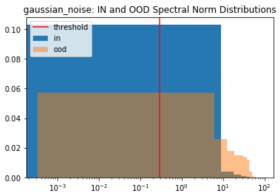
Google Images Distribution



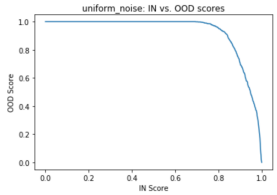
Gaussian Scores



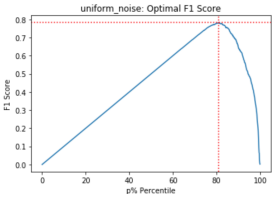
Gaussian F1 Curve



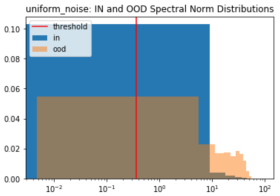
Gaussian Distribution



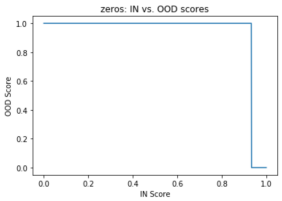
Uniform Scores



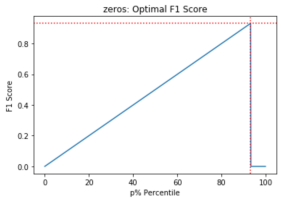
Uniform F1 Curve



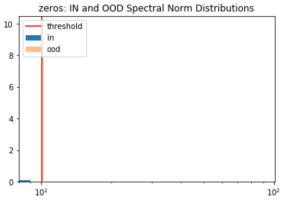
Uniform Distribution



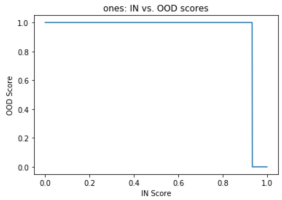
Zeros Scores



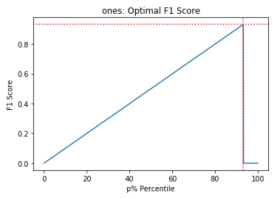
Zeros F1 Curve



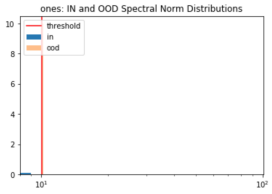
Zeros Distribution



Ones Scores

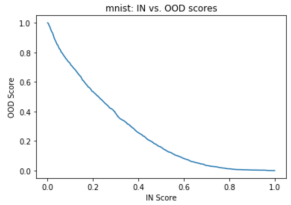


Ones F1 Curve

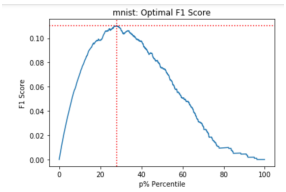


Ones Distribution

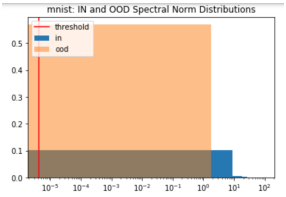
Naively Trained Model on Adversarial Samples



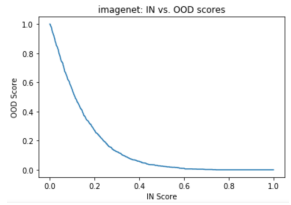
MNIST Scores



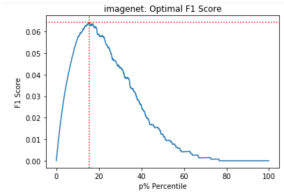
MNIST F1 Curve



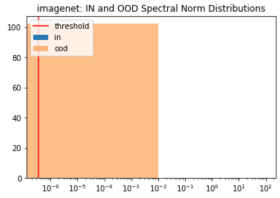
MNIST Distribution



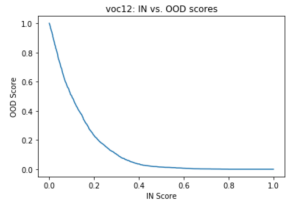
ImageNet Scores



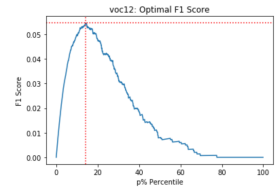
ImageNet F1 Curve



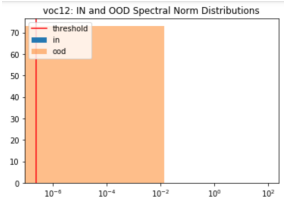
ImageNet Distribution



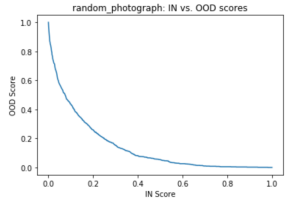
VOC-12 Scores



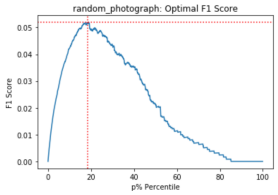
VOC-12 F1 Curve



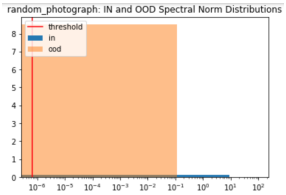
VOC-12 Distribution



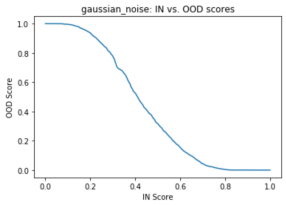
Google Images Scores



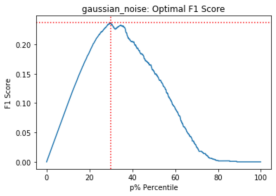
Google Images F1 Curve



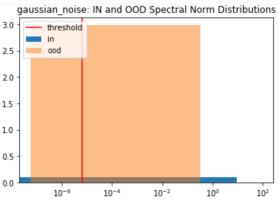
Google Images Distribution



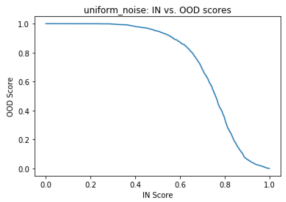
Gaussian Scores



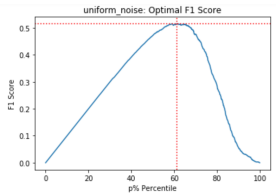
Gaussian F1 Curve



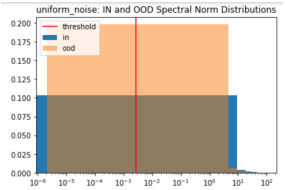
Gaussian Distribution



Uniform Scores

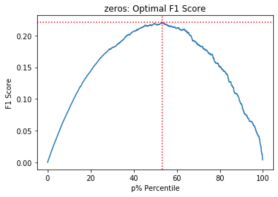


Uniform F1 Curve

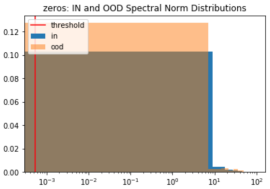


Uniform Distribution

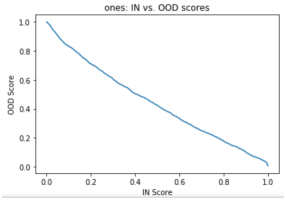
Zeros Scores



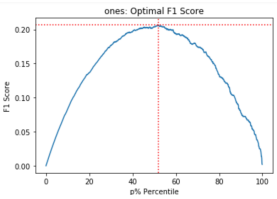
Zeros F1 Curve



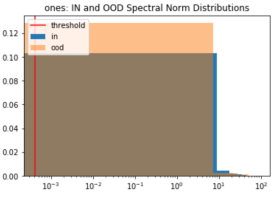
Zeros Distribution



Ones Scores

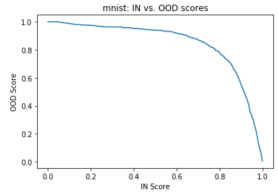


Ones F1 Curve

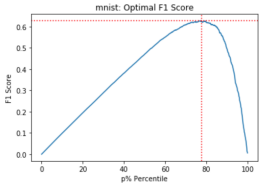


Ones Distribution

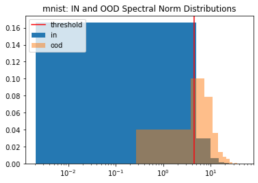
Adversarially Trained Model on Benign Samples



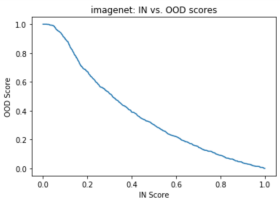
MNIST Scores



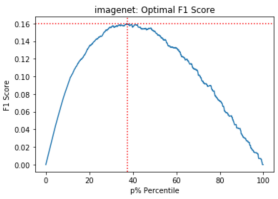
MNIST F1 Curve



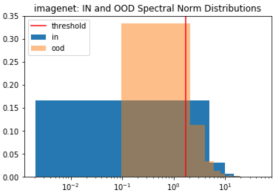
MNIST Distribution



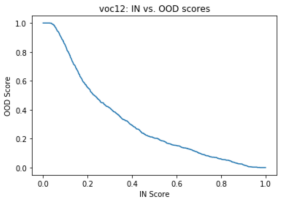
Imagenet Scores



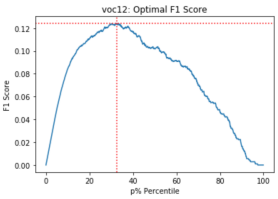
Imagenet F1 Curve



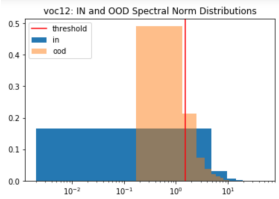
Imagenet Distribution



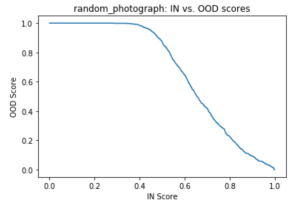
VOC-12 Scores



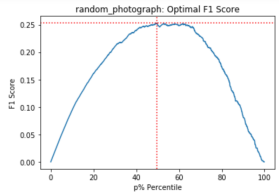
VOC-12 F1 Curve



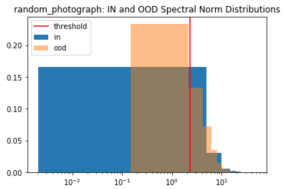
VOC-12 Distribution



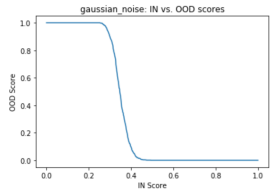
Google Images Scores



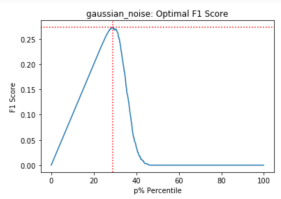
Google Images F1 Curve



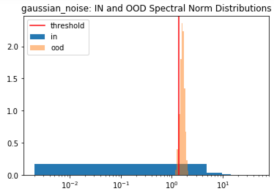
Google Images Distribution



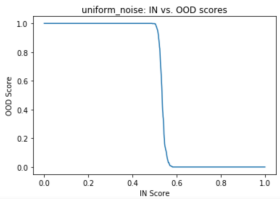
Gaussian Scores



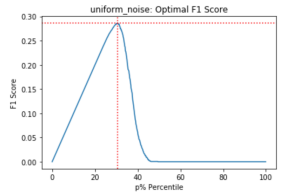
Gaussian F1 Curve



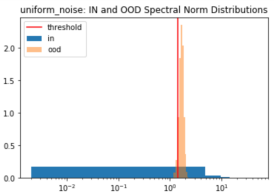
Gaussian Distribution



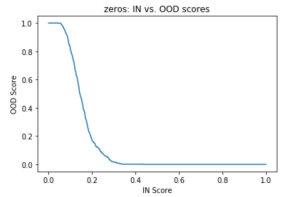
Uniform Scores



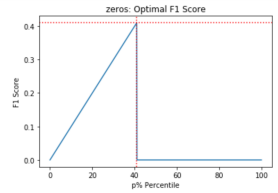
Uniform F1 Curve



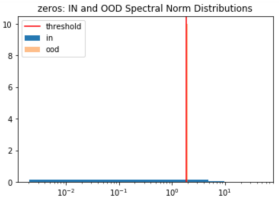
Uniform Distribution



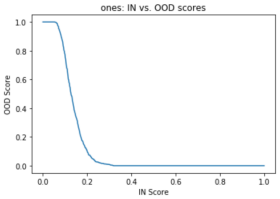
Zeros Scores



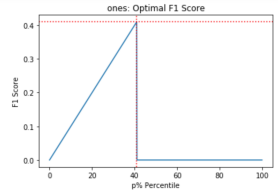
Zeros F1 Curve



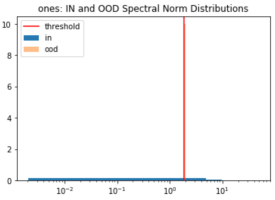
Zeros Distribution



Ones Scores

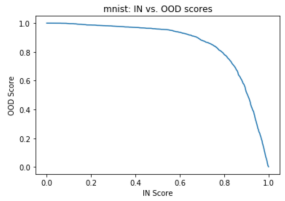


Ones F1 Curve

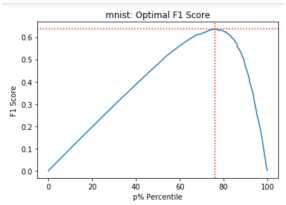


Ones Distribution

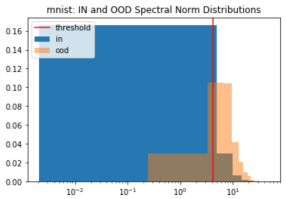
Adversarially Trained Model on Adversarial Samples



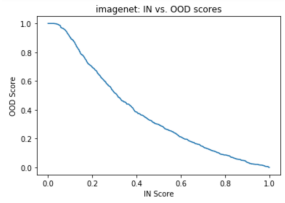
MNIST Scores



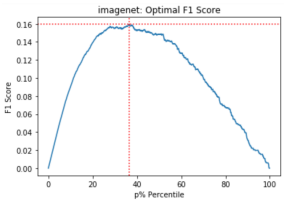
MNIST F1 Curve



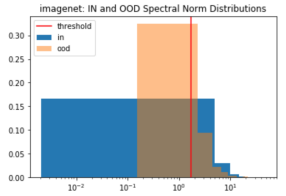
MNIST Distribution



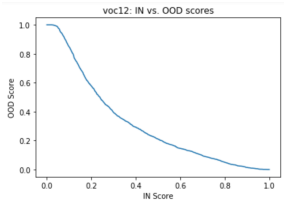
Imagenet Scores



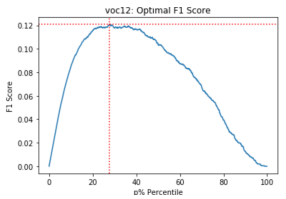
Imagenet F1 Curve



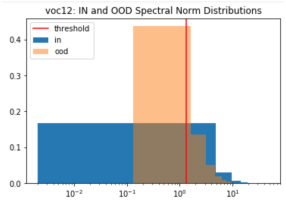
Imagenet Distribution



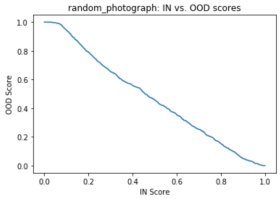
VOC-12 Scores



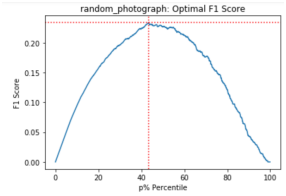
VOC-12 F1 Curve



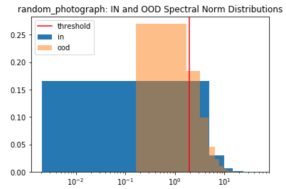
VOC-12 Distribution



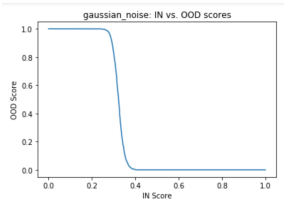
Google Images Scores



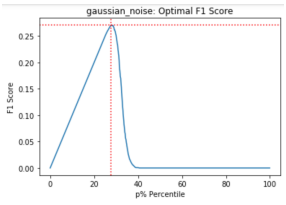
Google Images F1 Curve



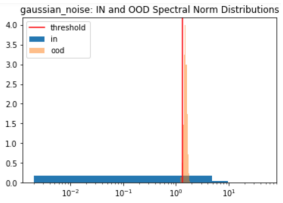
Google Images Distribution



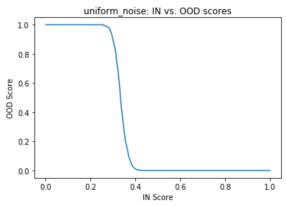
Gaussian Scores



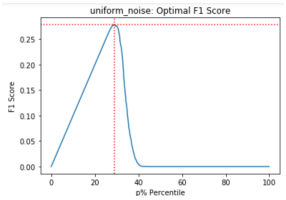
Gaussian F1 Curve



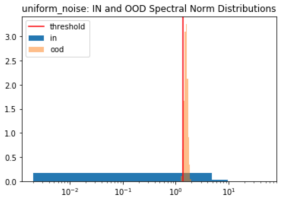
Gaussian Distribution



Uniform Scores

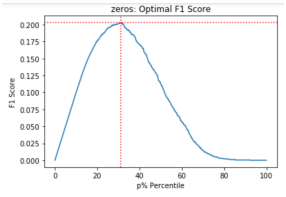


Uniform F1 Curve

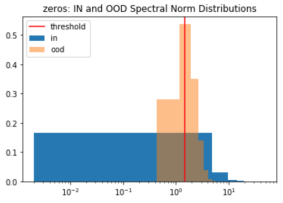


Uniform Distribution

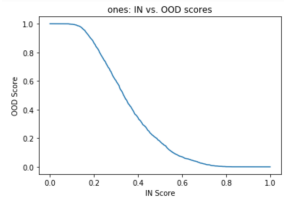
Zeros Scores



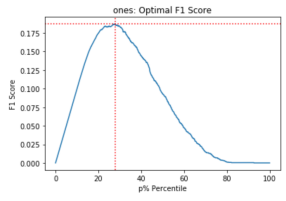
Zeros F1 Curve



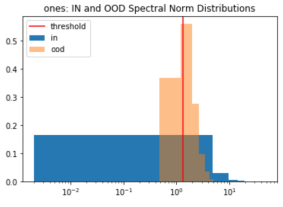
Zeros Distribution



Ones Scores



Ones F1 Curve



Ones Distribution

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[2] A. Authors, "Towards a rigorous evaluation of the robustness of open world deep learning," *International Conference on Computer Vision*, 2019.

[3] R. Calo, I. Evtimov, E. Fernandes, T. Kohno, and D. O'Hair, "Is tricking a robot hacking?" *SSRN Electronic Journal*, 01 2018.

[4] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," *CoRR*, vol. abs/1704.06743, 2017. [Online]. Available: http://arxiv.org/abs/1704.06743

[5] E. Commission, "Communication from the commission to the european parliament, the european council, the council, the european economic and social committee and the committee of the regions: Artificial intelligence for europe," 2018. [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/communication-artificial-intelligence-europe

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," 2009.

[7] A. R. Dhamija, M. Günther, and T. E. Boult, "Reducing network agnostophobia," *CoRR*, vol. abs/1811.04110, 2018. [Online]. Available: http://arxiv.org/abs/1811.04110

[8] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121 – 134, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320316300267

[9] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[10] S. G. Finlayson, I. S. Kohane, and A. L. Beam, "Adversarial attacks against medical deep learning systems," *CoRR*, vol. abs/1804.05296, 2018. [Online]. Available: http://arxiv.org/abs/1804.05296

[11] J. Gilmer, R. P. Adams, I. J. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," *CoRR*, vol. abs/1807.06732, 2018. [Online]. Available: http://arxiv.org/abs/1807.06732

[12] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase dataset," *Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304*.

[13] H. Jiang, B. Kim, M. Guan, and M. Gupta, "To trust or not to trust a classifier," pp. 5541–5552, 2018. [Online]. Available: http://papers.nips.cc/paper/7798-to-trust-or-not-to-trust-a-classifier.pdf

[14] D. Kirk, "Nvidia cuda software and gpu parallel computing architecture," pp. 103–104, 2007. [Online]. Available: http://doi.acm.org/10.1145/1296907.1296909

[15] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[16] R. S. S. Kumar, D. R. O'Brien, K. Albert, and S. Vilojen, "Law and adversarial machine learning," *CoRR*, vol. abs/1810.10731, 2018. [Online]. Available: http://arxiv.org/abs/1810.10731

[17] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[18] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," pp. 7167–7177, 2018. [Online]. Available: http://papers.nips.cc/paper/7947-a-simple-unified-framework-for-detecting-out-of-distribution-samples-and-adversarial-attacks

[19] V. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," 2018. [Online]. Available: https://openreview.net/pdf?id=H1VGkIxRZ

[20] S. Liu, R. Garrepalli, T. Dietterich, A. Fern, and D. Hendrycks, "Open category detection with PAC guarantees," vol. 80, pp. 3169–3178, 10–15 Jul 2018. [Online]. Available: http://proceedings.mlr.press/v80/liu18e.html

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2018. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[22] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," vol. 80, pp. 4393–4402, 10–15 Jul 2018. [Online]. Available: http://proceedings.mlr.press/v80/ruff18a.html

[23] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," *CoRR*, vol. abs/1703.05921, 2017. [Online]. Available: http://arxiv.org/abs/1703.05921

[24] V. Vikash, P. Mittal, D. Cullina, A. Bhagoji, and L. Song, "Better the devil you know: Analyzing open-world evasion attacks on deep learning," 2019.

[25] K. Vodrahalli, K. Li, and J. Malik, "Are all training examples created equal? an empirical study," *CoRR*, vol. abs/1811.12569, 2018. [Online]. Available: http://arxiv.org/abs/1811.12569

[26] P. Voigt and A. v. d. Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*, 1st ed. Springer Publishing Company, Incorporated, 2017.